Open Source Used In DNAC Platform
Antelope

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1204148659
This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please contact us at external-opensource-requests@cisco.com.

In your requests please include the following reference number 78EE117C99-1204148659

Contents

1.1 bindings 1.2.1
   1.1.1 Available under license
1.2 babel-plugin-transform-decorators 6.24.1
   1.2.1 Available under license
1.3 babel-plugin-transform-es2015-modules-commonjs 6.26.2
   1.3.1 Available under license
1.4 babel-plugin-check-es2015-constants 6.22.0 6.22.0
   1.4.1 Available under license
1.5 httpcore-nio 4.4.6
   1.5.1 Available under license
1.6 @babel/helper-split-export-declaration 7.11.0
   1.6.1 Available under license
1.7 iconv-lite 0.4.24
   1.7.1 Available under license
1.8 babel-plugin-transform-es2015-template-literals 6.22.0
   1.8.1 Available under license
1.9 netty 4.1.53.Final
   1.9.1 Available under license
1.10 @babel/plugin-transform-modules-umd 7.13.0
   1.10.1 Available under license
1.11 es6-promise 3.2.1 3.2.1
   1.11.1 Available under license
1.12 get-package-type 0.1.0
   1.12.1 Available under license
1.13 babel-plugin-transform-strict-mode 6.24.1
   1.13.1 Available under license
1.14 pend 1.2.0
  1.14.1 Available under license
1.15 parsejson 0.0.3
  1.15.1 Available under license
1.16 dtrace-provider 0.6.0 0.6.0
  1.16.1 Available under license
1.17 common-prefix 1.1.0 1.1.0
  1.17.1 Available under license
1.18 cross-spawn 6.0.5
  1.18.1 Available under license
1.19 cache-base 1.0.1
  1.19.1 Available under license
1.20 repeating 2.0.1 2.0.1
  1.20.1 Available under license
1.21 javax-el 3.0.0
  1.21.1 Available under license
1.22 assert-plus 1.0.0
  1.22.1 Available under license
1.23 netty-buffer 4.1.59.Final
  1.23.1 Available under license
1.24 babel-preset-stage-3 6.24.1
  1.24.1 Available under license
1.25 cli-cursor 2.1.0
  1.25.1 Available under license
1.26 cookiejar 2.1.2 2.1.2
  1.26.1 Available under license
1.27 jetty-continuation 9.4.17.v20190418
  1.27.1 Available under license
1.28 netty-tcnative-boringssl-static 2.0.26.Final
  1.28.1 Available under license
1.29 inherits 2.0.3 2.0.3
  1.29.1 Available under license
1.30 commons-io 2.7
  1.30.1 Available under license
1.31 ant-junit 1.10.9
  1.31.1 Available under license
1.32 batik-constants 1.13
  1.32.1 Available under license
1.33 prompt-tmp 1.0.0
<table>
<thead>
<tr>
<th>Version</th>
<th>Package Name</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.33.1</td>
<td>Open Source Used In DNAC Platform Antelope</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.34</td>
<td>cliui</td>
<td>7.0.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.35</td>
<td>hamcrest-library</td>
<td>1.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.36</td>
<td>spdx-expression-parse</td>
<td>3.0.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.37</td>
<td>escape-string-regexp</td>
<td>1.0.5</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.38</td>
<td>optionator</td>
<td>0.8.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.39</td>
<td>globals</td>
<td>9.18.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.40</td>
<td>@babel/core</td>
<td>7.13.14</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.41</td>
<td>hive-exec</td>
<td>3.1.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.42</td>
<td>mkdirp</td>
<td>0.5.5</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.43</td>
<td>acorn-jsx</td>
<td>3.0.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.44</td>
<td>eslint-module-utils</td>
<td>2.6.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.45</td>
<td>xmlhttprequest-ssl</td>
<td>1.5.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.46</td>
<td>kerby-xdr</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.47</td>
<td>typedarray</td>
<td>0.0.6</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.48</td>
<td>function-bind</td>
<td>1.1.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.49</td>
<td>uuid</td>
<td>3.4.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.50</td>
<td>camelcase</td>
<td>2.1.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.51</td>
<td>typetools</td>
<td>0.5.0</td>
<td></td>
</tr>
<tr>
<td>1.52</td>
<td>babel-plugin-transform-member-expression-literals</td>
<td>7.12.13</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.53</td>
<td>jnr-netdb</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>
1.53.1 Available under license

1.54 pdfjs-dist 2.0.305 2.0.305
1.54.1 Available under license

1.55 locate-path 2.0.0
1.55.1 Available under license

1.56 har-validator 5.1.5
1.56.1 Available under license

1.57 underscore 1.9.2 1.9.2
1.57.1 Available under license

1.58 ext 1.4.0
1.58.1 Available under license

1.59 whatwg-fetch 2.0.4 2.0.4
1.59.1 Available under license

1.60 resolve-url 0.2.1
1.60.1 Available under license

1.61 jackson-databind 2.10.5.1
1.61.1 Available under license

1.62 netty-transport-native-unix-common 4.1.59.Final
1.62.1 Available under license

1.63 fd-slicer 1.1.0
1.63.1 Available under license

1.64 prepend-http 2.0.0
1.64.1 Available under license

1.65 expiringmap 0.5.8

1.66 diff-match-patch 1.0.5
1.66.1 Available under license

1.67 ansi-escapes 1.4.0
1.67.1 Available under license

1.68 org.jacoco.agent.rt 0.7.9
1.68.1 Available under license

1.69 optionator 0.8.3
1.69.1 Available under license

1.70 checker-qual 3.5.0
1.70.1 Available under license

1.71 libevent 1.4.13
1.71.1 Available under license

1.72 poi-ooxml 5.0.0
1.72.1 Available under license

1.73 protobuf-java 2.5.0
1.74 jul-to-slf4j 1.7.26
1.75 mockito 2.23.4
   1.75.1 Available under license
1.76 babel-plugin-syntax-trailing-function-commas 6.22.0
   1.76.1 Available under license
1.77 file-type 11.1.0
   1.77.1 Available under license
1.78 jetty-security 9.4.34.v20201102
   1.78.1 Available under license
1.79 jffi 1.2.16
   1.79.1 Available under license
1.80 jmockit 1.47
   1.80.1 Available under license
1.81 isemail 1.2.0
   1.81.1 Available under license
1.82 jetty-io 9.3.27.v20190418
   1.82.1 Available under license
1.83 glob 7.1.3
   1.83.1 Available under license
1.84 pluralize 1.2.1
   1.84.1 Available under license
1.85 jackson 2.11.4
   1.85.1 Available under license
1.86 akka 2.5.31
   1.86.1 Available under license
1.87 moment-timezone 0.5.21 0.5.21
   1.87.1 Available under license
1.88 babel-preset-stage-2 6.24.1
   1.88.1 Available under license
1.89 type-is 1.6.16 1.6.16
   1.89.1 Available under license
1.90 jackson-dataformat-cbor 2.6.7
   1.90.1 Available under license
1.91 dayjs 1.9.3
   1.91.1 Available under license
1.92 @babel/parser 7.13.13
   1.92.1 Available under license
1.93 listenercount 1.0.1
   1.93.1 Available under license
1.94 is-arrayish 0.2.1
  1.94.1 Available under license
1.95 socket.io-client 2.3.0
  1.95.1 Available under license
1.96 amcharts3-animate 1.1.2 1.1.2
  1.96.1 Available under license
1.97 hadoop-hdfs-client 3.2.2
  1.97.1 Available under license
1.98 redux-act 1.7.4 1.7.4
  1.98.1 Available under license
1.99 tweetnacl 0.14.5
  1.99.1 Available under license
1.100 dtrace-provider 0.6.0
  1.100.1 Available under license
1.101 doctrine 2.1.0
  1.101.1 Available under license
1.102 string_decoder 1.0.3 1.0.3
  1.102.1 Available under license
1.103 httpcomponents-core 4.4.13
  1.103.1 Available under license
1.104 proxy-addr 2.0.6
  1.104.1 Available under license
1.105 @babel/plugin-transform-arrow-functions 7.13.0
  1.105.1 Available under license
1.106 require-uncached 1.0.3
  1.106.1 Available under license
1.107 require-directory 2.1.1 2.1.1
  1.107.1 Available under license
1.108 intl-messageformat-parser 1.4.0 1.4.0
  1.108.1 Available under license
1.109 inquirer 6.2.2
  1.109.1 Available under license
1.110 vary 1.0.1 1.0.1
  1.110.1 Available under license
1.111 jetty-util-ajax 9.4.41.v20210516
  1.111.1 Available under license
1.112 regenerator-runtime 0.13.7 0.13.7
  1.112.1 Available under license
1.113 hbase-metrics-api 2.4.2
1.113.1 Available under license

1.114 lucene 7.4.0
1.114.1 Available under license

1.115 jersey-server 2.25.1
1.115.1 Available under license

1.116 readable-stream 2.3.6 2.3.6
1.116.1 Available under license

1.117 caffeine 2.7.0
1.118 jnr-posix 1.1.4
1.118.1 Available under license

1.119 global-dirs 2.1.0
1.119.1 Available under license

1.120 htrace-core 3.2.0-incubating
1.120.1 Available under license

1.121 psl 1.8.0 1.8.0
1.121.1 Available under license

1.122 node-fetch 2.6.1 2.6.1
1.122.1 Available under license

1.123 commons-email 1.5
1.123.1 Available under license

1.124 wayland 1.12.0-1~ubuntu16.04.3
1.124.1 Available under license

1.125 expand-brackets 0.1.5
1.125.1 Available under license

1.126 odata-parser 1.4.1
1.126.1 Available under license

1.127 string-replace-async 2.0.0
1.127.1 Available under license

1.128 finalhandler 0.4.1 0.4.1
1.128.1 Available under license

1.129 composite-error 0.1.1 0.1.1
1.129.1 Available under license

1.130 hadoop-core 1.2.1
1.130.1 Available under license

1.131 batik-squiggle 1.13
1.131.1 Available under license

1.132 rimraf 3.0.2
1.132.1 Available under license

1.133 drill-shaded-guava 28.2-jre
1.134 lodash-difference 4.5.0
   1.134.1 Available under license
1.135 strip-ansi 3.0.1
   1.135.1 Available under license
1.136 @babel/helper-builder-binary-assignment-operator-visitor 7.12.13
   1.136.1 Available under license
1.137 batik-awt-util 1.13
   1.137.1 Available under license
1.138 jsreport-browser-client-dist 1.3.0
   1.138.1 Available under license
1.139 require_optional 1.0.1
   1.139.1 Available under license
1.140 stax 4.2
   1.140.1 Available under license
1.141 micromatch 2.3.11
   1.141.1 Available under license
1.142 flux 3.1.3 3.1.3
   1.142.1 Available under license
1.143 amqplib-tutorials 0.0.1
   1.143.1 Available under license
1.144 engine.io-client 1.8.5 1.8.5
   1.144.1 Available under license
1.145 regenerator-transform 0.10.1 0.10.1
   1.145.1 Available under license
1.146 asm-util 7.3.1
   1.146.1 Available under license
1.147 @babel/highlight 7.13.10
   1.147.1 Available under license
1.148 core-js 2.6.11
   1.148.1 Available under license
1.149 statuses 1.2.1
   1.149.1 Available under license
1.150 j2objc-annotations 1.1
   1.150.1 Available under license
1.151 bytes 0.2.1
   1.151.1 Available under license
1.152 lodash.escaperegexp 4.1.2
   1.152.1 Available under license
1.153 curator-x-discovery 5.1.0
1.153.1 Available under license
1.154 jersey-media-multipart 2.5.1
   1.154.1 Available under license
1.155 source-map-support 0.5.19
   1.155.1 Available under license
1.156 strip-bom-buf 2.0.0
   1.156.1 Available under license
1.157 object-keys 0.4.0
   1.157.1 Available under license
1.158 utils-merge 1.0.1
   1.158.1 Available under license
1.159 babel-generator 6.26.1 6.26.1
   1.159.1 Available under license
1.160 babel-plugin-transform-strict-mode 6.24.1 6.24.1
   1.160.1 Available under license
1.161 okhttp 4.8.0
   1.161.1 Available under license
1.162 jsesc 2.5.2
   1.162.1 Available under license
1.163 dashify 0.2.2 0.2.2
   1.163.1 Available under license
1.164 apache-log4j 2.14.0
   1.164.1 Available under license
1.165 xmlbuilder 10.1.1
   1.165.1 Available under license
1.166 @babel/plugin-syntax-dynamic-import 7.8.3
   1.166.1 Available under license
1.167 okhttp 4.4.0
   1.167.1 Available under license
1.168 parquet-jackson 1.10.0
   1.168.1 Available under license
1.169 node-signpdf 1.3.2
   1.169.1 Available under license
1.170 buffer-alloc-unsafe 1.1.0
   1.170.1 Available under license
1.171 ansi-styles 1.0.0 1.0.0
   1.171.1 Available under license
1.172 verror 1.10.0 1.10.0
   1.172.1 Available under license
1.173 groovy-groovydoc 3.0.8  
   1.173.1 Available under license
1.174 kerby-util 1.0.0  
   1.174.1 Available under license
1.175 mongodb 3.6.6 3.6.6  
   1.175.1 Available under license
1.176 assertion-error 1.1.0  
   1.176.1 Available under license
1.177 jetty-security 9.3.27.v20190418
1.178 cglib 3.2.4  
   1.178.1 Available under license
1.179 drill-yarn 1.19.0  
   1.179.1 Available under license
1.180 seamless-immutable 7.1.3 7.1.3  
   1.180.1 Available under license
   1.181.1 Available under license
1.182 hadoop-yarn-client 3.2.2  
   1.182.1 Available under license
1.183 bufferutil 1.2.1  
   1.183.1 Available under license
1.184 jline 2.14.3
1.185 bops 0.1.1  
   1.185.1 Available under license
1.186 yauaa-logparser 5.20  
   1.186.1 Available under license
1.187 retrofit 2.8.1  
   1.187.1 Available under license
1.188 he 1.2.0  
   1.188.1 Available under license
1.189 graphlib 2.1.8 2.1.8  
   1.189.1 Available under license
1.190 ms 0.7.2 0.7.2  
   1.190.1 Available under license
1.191 etag 1.8.1  
   1.191.1 Available under license
1.192 merge-class-names 1.3.0 1.3.0  
   1.192.1 Available under license
1.193 load-json-file 2.0.0
1.193.1 Available under license
1.194 immediate 3.0.6
  1.194.1 Available under license
1.195 jetty-io 9.4.17.v20190418
  1.195.1 Available under license
1.196 @babel/helper-validator-option 7.12.17
  1.196.1 Available under license
1.197 object-assign 4.1.0
  1.197.1 Available under license
1.198 kafka-avro-serializer 6.1.1
  1.198.1 Available under license
1.199 batch 0.4.0
  1.199.1 Available under license
1.200 buffer-equal-constant-time 1.0.1 1.0.1
  1.200.1 Available under license
1.201 xmlbuilder 8.2.2
  1.201.1 Available under license
1.202 hive-upgrade-acid 3.1.2
  1.202.1 Available under license
1.203 kerb-identity 1.0.0
  1.203.1 Available under license
1.204 libedit 3.1-20150325-1ubuntu2
  1.204.1 Available under license
1.205 jsreport-authorization 2.4.0
  1.205.1 Available under license
1.206 geo-ip 2.14.0
  1.206.1 Available under license
1.207 js-base64 2.6.4 2.6.4
  1.207.1 Available under license
1.208 jersey-container-servlet-core 2.5.1
  1.208.1 Available under license
1.209 babel-plugin-transform-es2015-modules-umd 6.24.1
  1.209.1 Available under license
1.210 drill-format-image 1.19.0
  1.210.1 Available under license
1.211 drill-storage-splunk 1.19.0
  1.211.1 Available under license
1.212 unpipe 1.0.0
  1.212.1 Available under license
<p>| 1.213 | http-errors 1.6.3 1.6.3 |
|       | 1.213.1 Available under license |
| 1.214 | batik-swing 1.13 |
|       | 1.214.1 Available under license |
| 1.215 | arraybuffer.slice 0.0.7 |
|       | 1.215.1 Available under license |
| 1.216 | wtf-8 1.0.0 |
|       | 1.216.1 Available under license |
| 1.217 | jersey-server 2.32 |
|       | 1.217.1 Available under license |
| 1.218 | libpng 1.2.54 1ubuntu1.1 |
|       | 1.218.1 Available under license |
| 1.219 | commander 2.20.3 2.20.3 |
|       | 1.219.1 Available under license |
| 1.220 | datanucleus-api-jdo 4.2.4 |
|       | 1.220.1 Available under license |
| 1.221 | orc-tools 1.5.6 |
|       | 1.221.1 Available under license |
| 1.222 | parquet-format 2.3.0-incubating |
|       | 1.222.1 Available under license |
| 1.223 | body-parser 1.19.0 |
|       | 1.223.1 Available under license |
| 1.224 | get-func-name 2.0.0 |
|       | 1.224.1 Available under license |
| 1.225 | setprototypeof 1.1.0 1.1.0 |
|       | 1.225.1 Available under license |
| 1.226 | emojis-list 3.0.0 |
|       | 1.226.1 Available under license |
| 1.227 | table 3.8.3 |
|       | 1.227.1 Available under license |
| 1.228 | validatejs 0.12.0 0.12.0 |
|       | 1.228.1 Available under license |
| 1.229 | is-ci 2.0.0 |
|       | 1.229.1 Available under license |
| 1.230 | moment 2.24.0 |
|       | 1.230.1 Available under license |
| 1.231 | fast-csv 3.7.0 |
|       | 1.231.1 Available under license |
| 1.232 | wrap-ansi 6.2.0 |</p>
<table>
<thead>
<tr>
<th>Version</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.232.1</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>1.233</td>
<td>@babel/plugin-syntax-object-rest-spread</td>
<td>7.8.3</td>
</tr>
<tr>
<td>1.234.1</td>
<td>jackson-dataformat-xml</td>
<td>2.9.10</td>
</tr>
<tr>
<td>1.235.1</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>1.236.1</td>
<td>is-primitive</td>
<td>2.0.0</td>
</tr>
<tr>
<td>1.237.1</td>
<td>jul-to-slf4j</td>
<td>1.7.5</td>
</tr>
<tr>
<td>1.238.1</td>
<td>batik-ext</td>
<td>1.13</td>
</tr>
<tr>
<td>1.239.1</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>1.240</td>
<td>jtoml</td>
<td>1.0.0</td>
</tr>
<tr>
<td>1.241.1</td>
<td>color-convert</td>
<td>2.0.1</td>
</tr>
<tr>
<td>1.242.1</td>
<td>concat-map</td>
<td>0.0.1</td>
</tr>
<tr>
<td>1.243.1</td>
<td>core-js</td>
<td>1.2.7</td>
</tr>
<tr>
<td>1.244.1</td>
<td>caniuse-lite</td>
<td>1.0.30000874</td>
</tr>
<tr>
<td>1.245.1</td>
<td>jackson-jaxrs-base</td>
<td>2.10.4</td>
</tr>
<tr>
<td>1.246.1</td>
<td>lru-cache</td>
<td>4.1.1</td>
</tr>
<tr>
<td>1.247.1</td>
<td>unicode-property-aliases-ecmascript</td>
<td>1.1.0</td>
</tr>
<tr>
<td>1.248.1</td>
<td>is-property</td>
<td>1.0.2</td>
</tr>
<tr>
<td>1.249.1</td>
<td>buffer-more-ints</td>
<td>1.0.0</td>
</tr>
<tr>
<td>1.250.1</td>
<td>supports-color</td>
<td>5.5.0</td>
</tr>
<tr>
<td>1.251.1</td>
<td>which-boxed-primitive</td>
<td>1.0.2</td>
</tr>
<tr>
<td>1.252.1</td>
<td>commons-lang3</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.253 core-js-compat 3.9.1
  1.253.1 Available under license
1.254 commons-validator 1.7
  1.254.1 Available under license
1.255 curator-recipes 5.1.0
  1.255.1 Available under license
1.256 babel-types 6.26.0 6.26.0
  1.256.1 Available under license
1.257 hive-storage-api 2.7.0
  1.257.1 Available under license
1.258 jakarta-annotation-api 1.3.5
  1.258.1 Available under license
1.259 jackson 2.10.5
  1.259.1 Available under license
1.260 protobuf-java 3.11.0
  1.260.1 Available under license
1.261 utils-merge 1.0.0 1.0.0
  1.261.1 Available under license
1.262 jdo-api 3.0.1
1.263 jython 2.5.2
  1.263.1 Available under license
1.264 mchange-commons-java 0.2.15
  1.264.1 Available under license
1.265 jetty-io 9.4.34.v20201102
  1.265.1 Available under license
1.266 is-desciptor 0.1.6
  1.266.1 Available under license
1.267 drill-jdbc-all 1.18.0
  1.267.1 Available under license
1.268 esrecurse 4.3.0
  1.268.1 Available under license
1.269 component-emitter 1.2.1 1.2.1
  1.269.1 Available under license
1.270 jedis 3.1.0
1.271 split 0.3.3 0.3.3
  1.271.1 Available under license
1.272 jetty-webapp 9.4.41.v20210516
  1.272.1 Available under license
1.273 @babel/plugin-transform-template-literals 7.13.0
1.273.1 Available under license
1.274 spss-reader 1.3
  1.274.1 Available under license
1.275 istanbul-lib-processinfo 2.0.2
  1.275.1 Available under license
1.276 domelementtype 1.3.0 1.3.0
  1.276.1 Available under license
1.277 backo2 1.0.2
  1.277.1 Available under license
1.278 home-or-tmp 2.0.0 2.0.0
  1.278.1 Available under license
1.279 qs 6.7.0
  1.279.1 Available under license
1.280 oauth-sign 0.9.0 0.9.0
  1.280.1 Available under license
1.281 color-name 1.1.1
  1.281.1 Available under license
1.282 commons-lang3 2.6
  1.282.1 Available under license
1.283 progress 2.0.3
  1.283.1 Available under license
1.284 has-symbols 1.0.2
  1.284.1 Available under license
1.285 emojis-list 2.1.0 2.1.0
  1.285.1 Available under license
1.286 xmlhttprequest-ssl 1.5.5
  1.286.1 Available under license
1.287 core-util-is 1.0.2 1.0.2
  1.287.1 Available under license
1.288 fast-infoset 1.2.15
1.289 topo 1.1.0 1.1.0
  1.289.1 Available under license
1.290 @babel/plugin-syntax-class-properties 7.12.13
  1.290.1 Available under license
1.291 dicer 0.2.5 0.2.5
  1.291.1 Available under license
1.292 calcite-avatica 1.17.0
  1.292.1 Available under license
1.293 aws-java-sdk-kms 1.11.375
1.293.1 Available under license
1.294 jetty-jmx 9.4.34.v20201102
  1.294.1 Available under license
1.295 uritemplate 0.3.4 0.3.4
  1.295.1 Available under license
1.296 nan 1.3.0
  1.296.1 Available under license
1.297 @babel/types 7.13.14
  1.297.1 Available under license
1.298 hbase-protocol 2.4.2
  1.298.1 Available under license
1.299 streamsearch 0.1.2
  1.299.1 Available under license
1.300 jsreport-pptx 0.7.0
  1.300.1 Available under license
1.301 javax-el 2.2.4
  1.301.1 Available under license
1.302 debug 4.1.1
  1.302.1 Available under license
1.303 ratelimitj-inmemory 0.4.0
  1.303.1 Available under license
1.304 pkg-dir 4.2.0
  1.304.1 Available under license
1.305 jackson-databind 2.12.3
  1.305.1 Available under license
1.306 ansi-styles 3.2.1
  1.306.1 Available under license
1.307 array-includes 3.1.1
  1.307.1 Available under license
1.308 lodash.reduce 4.5.0 4.5.0
  1.308.1 Available under license
1.309 babel-cli 6.26.0
  1.309.1 Available under license
1.310 regjsgen 0.2.0
  1.310.1 Available under license
1.311 http-errors 1.6.2 1.6.2
  1.311.1 Available under license
1.312 lodash.clonedep 4.5.0 4.5.0
  1.312.1 Available under license
<table>
<thead>
<tr>
<th>1.313</th>
<th>randexp 0.5.3 0.5.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.313.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.314</th>
<th>@babel/helper-simple-access 7.13.12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.314.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.315</th>
<th>aws-java-sdk-s3 1.11.375</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.315.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.316</th>
<th>debug 3.2.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.316.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.317</th>
<th>static-extend 0.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.317.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.318</th>
<th>jline-terminal-jansi 3.12.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.318.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.319</th>
<th>swagger-methods 1.0.8 1.0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.319.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.320</th>
<th>invariant 2.2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.320.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.321</th>
<th>signal-exit 3.0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.321.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.322</th>
<th>passport-http 0.3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.322.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.323</th>
<th>error_prone_annotations 2.1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.323.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.324</th>
<th>netty-codec 4.0.48.Final</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.324.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.325</th>
<th>batik-transcoder 1.13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.325.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.326</th>
<th>h2 1.4.197</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>1.327</th>
<th>babel-plugin-polyfill-corejs2 0.1.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.327.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.328</th>
<th>to-readable-stream 1.0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.328.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.329</th>
<th>async 2.1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.329.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.330</th>
<th>datanucleus-rdbms 4.1.19</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.330.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.331</th>
<th>drill-metastore-api 1.19.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.331.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.332</th>
<th>commondir 1.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.332.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.333 jsreport-browser-client 2.2.2
   1.333.1 Available under license
1.334 drill-hive-exec-shaded 1.19.0
   1.334.1 Available under license
1.335 ansi-styles 4.2.1
   1.335.1 Available under license
1.336 raml-to-swagger 1.1.0 1.1.0
   1.336.1 Available under license
1.337 pinkie-promise 2.0.1 2.0.1
   1.337.1 Available under license
1.338 ecdsa-sig-formatter 1.0.10 1.0.10
   1.338.1 Available under license
1.339 normalize-path 3.0.0
   1.339.1 Available under license
1.340 lucene-queries 7.4.0
   1.340.1 Available under license
1.341 debug 2.3.3
   1.341.1 Available under license
   1.342.1 Available under license
1.343 error_prone_annotations 2.3.4
   1.343.1 Available under license
1.344 nssocket 0.6.0
   1.344.1 Available under license
1.345 mongodb-driver-sync 4.2.3
   1.345.1 Available under license
1.346 aggregate-error 3.1.0
   1.346.1 Available under license
1.347 asap 2.0.6 2.0.6
   1.347.1 Available under license
1.348 stax 3.1.4
   1.348.1 Available under license
1.349 media-typer 0.3.0
   1.349.1 Available under license
1.350 range-parser 1.2.1
   1.350.1 Available under license
1.351 hadoop-yarn-api 3.2.2
   1.351.1 Available under license
1.352 parquet-format-structures 1.12.0
1.352.1 Available under license

1.353 **tough-cookie 2.5.0**
   1.353.1 Available under license

1.354 **babel-helper-optimise-call-expression 6.24.1 6.24.1**
   1.354.1 Available under license

1.355 **xtend 2.1.2 2.1.2**
   1.355.1 Available under license

1.356 **jersey-container-servlet-core 2.25.1**
   1.356.1 Available under license

1.357 **oas-kit-common 1.0.8 1.0.8**
   1.357.1 Available under license

1.358 **is-buffer 2.0.4 2.0.4**
   1.358.1 Available under license

1.359 **jersey-container-servlet-core 2.32**
   1.359.1 Available under license

1.360 **xtend 2.1.2**
   1.360.1 Available under license

1.361 **type-check 0.3.2**
   1.361.1 Available under license

1.362 **assign-symbols 1.0.0**
   1.362.1 Available under license

1.363 **@babel/preset-modules 0.1.4**
   1.363.1 Available under license

1.364 **component-emitter 1.1.2 1.1.2**
   1.364.1 Available under license

1.365 **configstore 5.0.1**
   1.365.1 Available under license

1.366 **get-caller-file 2.0.5**
   1.366.1 Available under license

1.367 **grpc-core 1.27.1**
   1.367.1 Available under license

1.368 **jetty-http 9.3.27.v20190418**
   1.368.1 Available under license

1.369 **append-field 1.0.0**
   1.369.1 Available under license

1.370 **jsp 2.3.1**
   1.370.1 Available under license

1.371 **component-emitter 1.3.0**
   1.371.1 Available under license
1.372 core-util-is 1.0.2
  1.372.1 Available under license
1.373 commons-collections 4.4
  1.373.1 Available under license
1.374 jetty-http-spi 9.4.17.v20190418
  1.374.1 Available under license
1.375 de.huxhorn.lilith.logback.classic 0.9.44
  1.375.1 Available under license
1.376 de.huxhorn.lilith.logback.appender.multiplex-core 0.9.44
  1.376.1 Available under license
1.377 is-extendable 0.1.1
  1.377.1 Available under license
1.378 debug 3.2.7 3.2.7
  1.378.1 Available under license
1.379 swagger 1.5.0
  1.379.1 Available under license
1.380 xregexp 4.0.0
  1.380.1 Available under license
1.381 schema-utils 0.4.7 0.4.7
  1.381.1 Available under license
1.382 is-accessor-descriptor 0.1.6
  1.382.1 Available under license
1.383 for-own 1.0.0
  1.383.1 Available under license
1.384 delayed-stream 1.0.0
  1.384.1 Available under license
1.385 @babel/plugin-transform-function-name 7.12.13
  1.385.1 Available under license
1.386 hawtjni-runtime 1.17
  1.386.1 Available under license
1.387 commons-lang3 3.11
  1.387.1 Available under license
1.388 event-stream 3.3.4 3.3.4
  1.388.1 Available under license
1.389 yargs-parser 18.1.3 18.1.3
  1.389.1 Available under license
1.390 assert-plus 1.0.0 1.0.0
  1.390.1 Available under license
1.391 color-convert 1.9.2
1.391.1 Available under license
1.392 depd 1.1.1 1.1.1
  1.392.1 Available under license
1.393 binary-extensions 1.13.1
  1.393.1 Available under license
1.394 repeat-string 1.6.1
  1.394.1 Available under license
1.395 pdfbox 2.0.22
  1.395.1 Available under license
1.396 chokidar 2.1.8
  1.396.1 Available under license
1.397 babel-code-frame 6.26.0
  1.397.1 Available under license
1.398 acorn 3.3.0
  1.398.1 Available under license
1.399 jszip 3.5.0
  1.399.1 Available under license
1.400 log4j-slf4j-impl 2.8.2
  1.400.1 Available under license
1.401 javax-servlet-jsp-api 2.1
  1.401.1 Available under license
1.402 jline 2.14.6
  1.402.1 Available under license
1.403 annotations 4.1.1.4
  1.403.1 Available under license
1.404 map-age-cleaner 0.1.3
  1.404.1 Available under license
1.405 convert-source-map 1.7.0
  1.405.1 Available under license
1.406 hadoop-client 3.2.2
  1.406.1 Available under license
1.407 ienoopen 1.0.0
  1.407.1 Available under license
1.408 libtiff 4.0.6 1ubuntu0.7
  1.408.1 Available under license
1.409 kafka-raft 2.8.0
  1.409.1 Available under license
1.410 d-bus 1.10.6-1ubuntu3.5
  1.410.1 Available under license
Open Source Used In DNAC Platform Antelope

1.430.1 Available under license
1.431 snapdragon-node 2.1.1
    1.431.1 Available under license
1.432 jsreport-templates 2.4.0
    1.432.1 Available under license
1.433 kerb-core 1.0.0
    1.433.1 Available under license
1.434 hk2-utils 2.6.1
    1.434.1 Available under license
1.435 glib-networking 2.48.2 1~ubuntu16.04.1
    1.435.1 Available under license
1.436 nspr 4.13.1 Oubuntu0.16.04.1
    1.436.1 Available under license
1.437 encoding 0.1.12 0.1.12
    1.437.1 Available under license
1.438 snake-yaml 1.27
    1.438.1 Available under license
1.439 avro-ipc 1.9.1
    1.439.1 Available under license
1.440 json-schema-traverse 0.4.1 0.4.1
    1.440.1 Available under license
1.441 lodash-groupby 4.6.0
    1.441.1 Available under license
1.442 @babel/plugin-proposal-unicode-property-regex 7.12.13
    1.442.1 Available under license
1.443 jackson-xc 2.10.4
    1.443.1 Available under license
1.444 ansicolors 0.2.1
    1.444.1 Available under license
1.445 commons-cli 1.2
    1.445.1 Available under license
1.446 decode-uri-component 0.2.0
    1.446.1 Available under license
1.447 de.huxhorn.sulky.formatting 0.9.17
    1.447.1 Available under license
1.448 log-symbols 3.0.0
    1.448.1 Available under license
1.449 object.getownpropertydescriptors 2.1.2
    1.449.1 Available under license
1.450 twill-zookeeper 0.6.0-incubating
   1.450.1 Available under license
1.451 engine.io-parser 2.2.1
   1.451.1 Available under license
1.452 parquet-column 1.8.1
   1.452.1 Available under license
1.453 istanbuljs-schema 0.1.3
   1.453.1 Available under license
1.454 prop-types 15.6.2 15.6.2
   1.454.1 Available under license
1.455 jcl-over-slf4j 1.7.26
1.456 utf-8-validate 1.2.1
   1.456.1 Available under license
1.457 intl-messageformat 2.2.0 2.2.0
   1.457.1 Available under license
1.458 amcharts3-export 1.4.73 1.4.73
   1.458.1 Available under license
1.459 write 0.2.1
   1.459.1 Available under license
1.460 source-map 0.6.1 0.6.1
   1.460.1 Available under license
1.461 swagger-annotations 1.5.16
   1.461.1 Available under license
1.462 is-utf8 0.2.1
   1.462.1 Available under license
1.463 drill-format-ltsv 1.19.0
   1.463.1 Available under license
1.464 has-ansi 2.0.0
   1.464.1 Available under license
1.465 @babel/code-frame 7.10.4
   1.465.1 Available under license
1.466 vary 1.1.2 1.1.2
   1.466.1 Available under license
1.467 mime-db 1.44.0
   1.467.1 Available under license
1.468 drill-storage-cassandra 1.19.0
   1.468.1 Available under license
1.469 jffi 1.2.15
   1.469.1 Available under license
1.470 restore-cursor 2.0.0
   1.470.1 Available under license
1.471 jsreport-studio-theme-dark 0.3.0
   1.471.1 Available under license
   1.472.1 Available under license
1.473 arr-flatten 1.1.0
   1.473.1 Available under license
1.474 mail 1.5.6
   1.474.1 Available under license
1.475 jnr-posix 3.0.41
   1.475.1 Available under license
1.476 source-map 0.6.0 0.6.0
   1.476.1 Available under license
1.477 nyc 15.1.0
   1.477.1 Available under license
1.478 estraverse 4.2.0
   1.478.1 Available under license
1.479 swagger 2.0.7
   1.479.1 Available under license
1.480 parse-glob 3.0.4
   1.480.1 Available under license
1.481 array-flatten 1.1.1 1.1.1
   1.481.1 Available under license
1.482 install 0.12.2 0.12.2
   1.482.1 Available under license
1.483 is-stream 1.1.0 1.1.0
   1.483.1 Available under license
1.484 asm 3.3.1
   1.484.1 Available under license
1.485 curator-framework 5.1.0
   1.485.1 Available under license
1.486 inherits 2.0.3
   1.486.1 Available under license
1.487 mime 2.6.0
   1.487.1 Available under license
1.488 lodash-reject 4.6.0
   1.488.1 Available under license
1.489 mimic-response 1.0.1
1.489.1 Available under license

1.490 object-assign 4.1.1 4.1.1
   1.490.1 Available under license

1.491 dashdash 1.14.1
   1.491.1 Available under license

1.492 std 4.1
   1.492.1 Available under license

1.493 okhttp 2.7.5
   1.493.1 Available under license

1.494 common-utils 6.1.1
   1.494.1 Available under license

1.495 format-util 1.0.5 1.0.5
   1.495.1 Available under license

1.496 string-width 3.1.0
   1.496.1 Available under license

1.497 ws 1.1.5 1.1.5
   1.497.1 Available under license

1.498 x-xss-protection 1.0.0
   1.498.1 Available under license

1.499 object-assign 2.1.1 2.1.1
   1.499.1 Available under license

1.500 diff 3.5.0
   1.500.1 Available under license

1.501 jmespath-java 1.11.375
   1.501.1 Available under license

1.502 archiver 2.1.1
   1.502.1 Available under license

1.503 commons-beanutils-core 1.8.0
   1.503.1 Available under license

1.504 @babel/plugin-syntax-optional-catch-binding 7.8.3
   1.504.1 Available under license

1.505 hsqldb-jdbc 1.8.0.10
   1.505.1 Available under license

1.506 globals 11.12.0
   1.506.1 Available under license

1.507 extend-shallow 3.0.2
   1.507.1 Available under license

1.508 lodash 4.17.19
   1.508.1 Available under license
1.509 mime-db 1.35.0 1.35.0
   1.509.1 Available under license
1.510 chrome-page-eval 1.3.0
   1.510.1 Available under license
1.511 jasper-compiler 5.5.12
   1.511.1 Available under license
1.512 commons-compress 2.1.1
   1.512.1 Available under license
1.513 caniuse-lite 1.0.30001202
   1.513.1 Available under license
1.514 invert-kv 2.0.0
   1.514.1 Available under license
1.515 graphics2d 0.30
   1.515.1 Available under license
1.516 guava 30.1.1-jre
   1.516.1 Available under license
1.517 babel-plugin-syntax-optional-chaining 7.8.3
   1.517.1 Available under license
1.518 lodash.defaults 4.2.0
   1.518.1 Available under license
1.519 apiguardian-api 1.1.0
   1.519.1 Available under license
1.520 jaxb-core 2.3.1
   1.520.1 Available under license
1.521 mime 1.6.0
   1.521.1 Available under license
1.522 colors.js 1.0.3
   1.522.1 Available under license
1.523 domelementtype 1.1.3 1.1.3
   1.523.1 Available under license
1.524 httpcomponents-core 4.4.10
   1.524.1 Available under license
1.525 drill-storage-kafka 1.19.0
   1.525.1 Available under license
1.526 object-keys 1.0.12
   1.526.1 Available under license
1.527 commons-net 3.6
   1.527.1 Available under license
1.528 co 4.6.0 4.6.0
1.528.1 Available under license
1.529 **topo** 1.1.0
   1.529.1 Available under license
1.530 **semver** 5.7.1 5.7.1
   1.530.1 Available under license
1.531 **jets3t** 0.6.1
   1.531.1 Available under license
1.532 **jacoco-core** 0.7.9
   1.532.1 Available under license
1.533 **mime** 1.6.0 1.6.0
   1.533.1 Available under license
1.534 **animal-sniffer-annotation** 1.14
1.535 **protobuf-java** 3.14.0
   1.535.1 Available under license
1.536 **jctools-core** 2.1.1
   1.536.1 Available under license
1.537 **body-parser** 1.18.3 1.18.3
   1.537.1 Available under license
1.538 **libxml** 2.9.3+dfsg1 1ubuntu0.7
   1.538.1 Available under license
1.539 **word-wrap** 1.2.3
   1.539.1 Available under license
1.540 **jersey-media-jaxb** 2.25.1
   1.540.1 Available under license
1.541 **szmarczak-http-timer** 1.1.2
   1.541.1 Available under license
1.542 **websocket-api** 9.4.17.v20190418
   1.542.1 Available under license
1.543 **slf4j** 1.7.26
   1.543.1 Available under license
1.544 **backo2** 1.0.2 1.0.2
   1.544.1 Available under license
1.545 **freemarker** 2.3.28
   1.545.1 Available under license
1.546 **engine.io-client** 1.8.5
   1.546.1 Available under license
1.547 **cli-cursor** 1.0.2
   1.547.1 Available under license
1.548 **ms** 0.7.1 0.7.1

---

Open Source Used In DNAC Platform Antelope 29
1.548.1 Available under license
1.549 nanoid 3.1.12
1.549.1 Available under license
1.550 httpcomponents-client 4.5.13
1.550.1 Available under license
1.551 has-ansi 2.0.0 2.0.0
1.551.1 Available under license
1.552 range-parser 1.0.3 1.0.3
1.552.1 Available under license
1.553 punycode 2.1.1 2.1.1
1.553.1 Available under license
1.554 got 9.6.0
1.554.1 Available under license
1.555 prefixmap 2.0
1.555.1 Available under license
1.556 esprima 2.7.3
1.556.1 Available under license
1.557 commons-net 1.4.1
1.557.1 Available under license
1.558 readable-stream 1.1.14
1.558.1 Available under license
1.559 statuses 1.5.0 1.5.0
1.559.1 Available under license
1.560 jersey-container-servlet 2.25.1
1.560.1 Available under license
1.561 spawn-wrap 2.0.0
1.561.1 Available under license
1.562 send 0.16.2 0.16.2
1.562.1 Available under license
1.563 sqlline 1.9.0
1.563.1 Available under license
1.564 aws-java-sdk-core 1.11.837
1.564.1 Available under license
1.565 xerces-j 2.11.0
1.565.1 Available under license
1.566 opentest4j 1.2.0
1.566.1 Available under license
1.567 bitsyntx 0.0.4 0.0.4
1.567.1 Available under license
1.568 bunyan-format 0.2.1
   1.568.1 Available under license
1.569 babel-plugin-transform-es2015-duplicate-keys 6.24.1
   1.569.1 Available under license
1.570 commons-compress 1.19
   1.570.1 Available under license
1.571 hbase-shaded-protobuf 3.4.1
   1.571.1 Available under license
1.572 pify 4.0.1
   1.572.1 Available under license
1.573 p-locate 3.0.0
   1.573.1 Available under license
1.574 asm-analysis 7.3.1
   1.574.1 Available under license
1.575 commons-compress 1.20
   1.575.1 Available under license
1.576 hsts 1.0.0
   1.576.1 Available under license
1.577 figures 2.0.0
   1.577.1 Available under license
1.578 lodash.once 4.1.1
   1.578.1 Available under license
1.579 grpc-protobuf 1.27.1
   1.579.1 Available under license
1.580 @babel/plugin-transform-block-scoped-functions 7.12.13
   1.580.1 Available under license
1.581 progress 1.1.8
   1.581.1 Available under license
1.582 methods 1.1.2
   1.582.1 Available under license
1.583 jnr-constants 0.9.0
1.584 output-file-sync 1.1.2
   1.584.1 Available under license
1.585 @babel/plugin-transform-sticky-regex 7.12.13
   1.585.1 Available under license
1.586 @babel/plugin-transform-classes 7.13.0
   1.586.1 Available under license
1.587 http-status-codes 1.4.0 1.4.0
   1.587.1 Available under license
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>combined-stream</td>
<td>1.0.6</td>
<td>1.0.6</td>
</tr>
<tr>
<td>classgraph</td>
<td>4.6.32</td>
<td></td>
</tr>
<tr>
<td>lie</td>
<td>3.3.0</td>
<td>1.0.0</td>
</tr>
<tr>
<td>x-xss-protection</td>
<td>1.0.0</td>
<td>1.0.0</td>
</tr>
<tr>
<td>commons-lang3</td>
<td>3.8.1</td>
<td>1.0.0</td>
</tr>
<tr>
<td>babel-traverse</td>
<td>6.26.0</td>
<td>6.26.0</td>
</tr>
<tr>
<td>generate-schema</td>
<td>2.6.0</td>
<td>2.6.0</td>
</tr>
<tr>
<td>groovy-docgenerator</td>
<td>3.0.8</td>
<td>1.0.0</td>
</tr>
<tr>
<td>httpcore5-h</td>
<td>5.0.2</td>
<td>1.0.0</td>
</tr>
<tr>
<td>cglb</td>
<td>2.2.0-b21</td>
<td>1.0.0</td>
</tr>
<tr>
<td>libpciaccess</td>
<td>0.13.4</td>
<td>1.0.0</td>
</tr>
<tr>
<td>babel-helper-hoist-variables</td>
<td>6.24.1</td>
<td>1.0.0</td>
</tr>
<tr>
<td>engine.io-client</td>
<td>3.4.4</td>
<td>1.0.0</td>
</tr>
<tr>
<td>jackson-dataformat-yaml</td>
<td>2.10.0</td>
<td>1.0.0</td>
</tr>
<tr>
<td>open-csv</td>
<td>5.4</td>
<td>1.0.0</td>
</tr>
<tr>
<td>object.assign</td>
<td>4.1.1</td>
<td>1.0.0</td>
</tr>
<tr>
<td>has-values</td>
<td>1.0.0</td>
<td>1.0.0</td>
</tr>
<tr>
<td>websocket-client</td>
<td>9.3.27.v20190418</td>
<td>1.0.0</td>
</tr>
<tr>
<td>libcroco</td>
<td>0.6.11</td>
<td>1.0.0</td>
</tr>
<tr>
<td>domutils</td>
<td>1.7.0</td>
<td>1.0.0</td>
</tr>
<tr>
<td>on-headers</td>
<td>1.0.1</td>
<td>1.0.1</td>
</tr>
</tbody>
</table>
1.608.1 Available under license
1.609 batik-svgpp 1.13
   1.609.1 Available under license
1.610 cheerio 0.22.0
   1.610.1 Available under license
1.611 destroy 1.0.4
   1.611.1 Available under license
1.612 commons-math 2.1
   1.612.1 Available under license
1.613 loose-envify 1.4.0
   1.613.1 Available under license
1.614 object-inspect 1.10.3 1.10.3
   1.614.1 Available under license
1.615 is-dotfile 1.0.3
   1.615.1 Available under license
1.616 ajv-keywords 3.2.0 3.2.0
   1.616.1 Available under license
1.617 path-to-regexp 2.1.0
   1.617.1 Available under license
1.618 hive-hbase-handler 3.1.2
   1.618.1 Available under license
1.619 setprototypeof 1.1.1
   1.619.1 Available under license
1.620 callsite 1.0.0
   1.620.1 Available under license
1.621 @babel/plugin-transform-block-scoping 7.12.13
   1.621.1 Available under license
1.622 icu 65
   1.622.1 Available under license
1.623 parseuri 0.0.5 0.0.5
   1.623.1 Available under license
1.624 lodash.reduce 4.6.0
   1.624.1 Available under license
1.625 javaewah 0.3.2
   1.625.1 Available under license
1.626 jackson-databind 2.4.0
   1.626.1 Available under license
1.627 babel-plugin-check-es2015-constants 6.22.0
   1.627.1 Available under license
1.628 jetty-servlets 9.4.41.v20210516
   1.628.1 Available under license
1.629 groovy-testng 3.0.8
   1.629.1 Available under license
1.630 regenerator-runtime 0.10.5
   1.630.1 Available under license
1.631 camelcase 5.3.1
   1.631.1 Available under license
1.632 escape-html 1.0.3 1.0.3
   1.632.1 Available under license
1.633 parse-filepath 1.0.2
   1.633.1 Available under license
1.634 should-util 1.0.1 1.0.1
   1.634.1 Available under license
1.635 mime-lookup 0.0.2 0.0.2
   1.635.1 Available under license
1.636 regenerator-runtime 0.11.1 0.11.1
   1.636.1 Available under license
1.637 httpcore-nio 4.4.10
   1.637.1 Available under license
1.638 swagger-jaxrs2 2.0.7
   1.638.1 Available under license
1.639 balanced-match 1.0.0 1.0.0
   1.639.1 Available under license
1.640 metadata-extractor 2.13.0
   1.640.1 Available under license
1.641 rimraf 2.4.5 2.4.5
   1.641.1 Available under license
1.642 shebang-command 1.2.0
   1.642.1 Available under license
1.643 jctools-core 2.0.1
1.644 metrics-json 4.0.2
1.645 lz4-java 1.7.1
   1.645.1 Available under license
1.646 jakarta-ws-rs-api 2.1.6
   1.646.1 Available under license
1.647 lombok 1.18.20
   1.647.1 Available under license
1.648 dbf-reader 0.3.0
1.648.1 Available under license
1.649 reselect 3.0.1 3.0.1
  1.649.1 Available under license
  1.650.1 Available under license
1.651 jackson-datatype-joda 2.4.2
  1.651.1 Available under license
1.652 process-nextick-args 1.0.7 1.0.7
  1.652.1 Available under license
1.653 jsreport-xlsx 2.5.0
  1.653.1 Available under license
1.654 busboy 0.2.14
  1.654.1 Available under license
1.655 bytes 2.3.0
  1.655.1 Available under license
1.656 call-me-maybe 1.0.1 1.0.1
  1.656.1 Available under license
1.657 google-http-client 1.20.0
1.658 mime-types 2.1.19 2.1.19
  1.658.1 Available under license
1.659 d-conf 0.24.0-2
  1.659.1 Available under license
1.660 semver 5.6.0
  1.660.1 Available under license
1.661 ultron 1.0.2 1.0.2
  1.661.1 Available under license
1.662 hive-metastore 3.1.2
  1.662.1 Available under license
1.663 get-intrinsic 1.1.1
  1.663.1 Available under license
1.664 winser-with-api 1.0.1
  1.664.1 Available under license
1.665 mime-types 2.1.30 2.1.30
  1.665.1 Available under license
1.666 mime-pull 1.9.3
1.667 objenesis 2.1
  1.667.1 Available under license
1.668 minio 4.0.0
1.669 de.huxhorn.lilith.logback.appender.multiplex-classic 0.9.44
Open Source Used In DNAC Platform: Antelope

- jetty-webapp 9.4.17.v20190418
  - Available under license
- aws-java-sdk-core 1.11.375
  - Available under license
- registry-url 5.1.0
  - Available under license
- ajv-keywords 1.5.1
  - Available under license
- fast-deep-equal 3.1.3
  - Available under license
- babel-helper-builder-binary-assignment-operator-visitor 6.24.1
  - Available under license
- glob 6.0.4
  - Available under license
- hadoop-distcp 3.2.2
  - Available under license
- netty-handler 4.1.42.Final
  - Available under license
- gir1.2-gdkpixbuf-2.0 2.32.2-1ubuntu1.6
  - Available under license
- sqlite 3.30.1
  - Available under license
- babel-plugin-transform-es2015-spread
  - 6.22.0
  - Available under license
- deep-equal 0.2.2
  - Available under license
- bson 1.0.9
  - Available under license
- map-cache 0.2.2
  - Available under license
- httpsnippet 1.16.5
  - Available under license
- javax-el-api 2.2.4
  - Available under license
- json-path 2.4.0
  - Available under license
- regjsgen 0.5.2
  - Available under license
- commons-pool 2.6.0
  - Available under license
<table>
<thead>
<tr>
<th>Version</th>
<th>Package Name</th>
<th>Version</th>
<th>License Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.689.1</td>
<td>Available under license</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.690</td>
<td>eslint 3.6.0</td>
<td>1.690.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.691</td>
<td>native-promise-only 0.8.1 0.8.1</td>
<td>1.691.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.692</td>
<td>@babel/comapt-data 7.13.12</td>
<td>1.692.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.693</td>
<td>drill-storage-elasticsearch 1.19.0</td>
<td>1.693.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.694</td>
<td>ip-address 5.3.1</td>
<td>1.694.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.695</td>
<td>metrics 4.1.0</td>
<td>1.695.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.696</td>
<td>sparsebitset 1.2</td>
<td>1.696.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.697</td>
<td>async-replace 1.0.1</td>
<td>1.697.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.698</td>
<td>javassist 3.22.0-GA</td>
<td>1.698.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.699</td>
<td>script-manager 0.10.2</td>
<td>1.699.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.700</td>
<td>remove-trailing-separator 1.1.0</td>
<td>1.700.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.701</td>
<td>unicode-match-property-ecmascript 1.0.4</td>
<td>1.701.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.702</td>
<td>jsonify 0.0.0</td>
<td>1.702.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.703</td>
<td>querystringify 2.2.0</td>
<td>1.703.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.704</td>
<td>es-abstract 1.18.0</td>
<td>1.704.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.705</td>
<td>eslint-import-resolver-node 0.3.4</td>
<td>1.705.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.706</td>
<td>object-assign 4.1.0 4.1.0</td>
<td>1.706.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.707</td>
<td>junit-platform-engine 1.7.0</td>
<td>1.707.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.708</td>
<td>json-stable-stringify 1.0.1</td>
<td>1.708.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.709 ultron 1.0.2
   1.709.1 Available under license
1.710 objenesis 2.5.1
   1.710.1 Available under license
1.711 @babel/highlight 7.10.4
   1.711.1 Available under license
1.712 json-schema-faker 0.5.0-rcv.34 0.5.0-rcv.34
   1.712.1 Available under license
1.713 is-callable 1.2.2
   1.713.1 Available under license
1.714 fbjs 0.8.17 0.8.17
   1.714.1 Available under license
1.715 number-is-nan 1.0.1
   1.715.1 Available under license
1.716 regenerator-transform 0.14.5
   1.716.1 Available under license
1.717 commander 1.2.0
   1.717.1 Available under license
1.718 indexof 0.0.1 0.0.1
1.719 end-of-stream 1.4.4 1.4.4
   1.719.1 Available under license
1.720 p-locate 2.0.0
   1.720.1 Available under license
1.721 accessors-smart 1.2
   1.721.1 Available under license
1.722 duplexify 3.7.1 3.7.1
   1.722.1 Available under license
1.723 @babel/template 7.10.4
   1.723.1 Available under license
1.724 make-dir 3.1.0
   1.724.1 Available under license
1.725 ansi-styles 2.2.1 2.2.1
   1.725.1 Available under license
1.726 mkdirp 0.5.1 0.5.1
   1.726.1 Available under license
1.727 kerb-server 1.0.0
   1.727.1 Available under license
1.728 jsreport-fs-store 2.9.1
   1.728.1 Available under license
1.729 string_decoder 0.10.31 0.10.31
  1.729.1 Available under license
1.730 lodash.isstring 4.0.1
  1.730.1 Available under license
1.731 calcite-core 1.26.0
  1.731.1 Available under license
1.732 escape-goat 2.1.1
  1.732.1 Available under license
1.733 camelize 1.0.0 1.0.0
  1.733.1 Available under license
1.734 phantom-page-eval 1.2.0
  1.734.1 Available under license
1.735 duplexer3 0.1.4
  1.735.1 Available under license
1.736 raw-body 2.3.3 2.3.3
  1.736.1 Available under license
1.737 hive-service-rpc 3.1.2
  1.737.1 Available under license
1.738 capnproto 0.5.3-2ubuntu1.1
  1.738.1 Available under license
1.739 ms 2.0.0 2.0.0
  1.739.1 Available under license
1.740 netty-transport 4.1.59.Final
  1.740.1 Available under license
1.741 cookie 0.1.5
  1.741.1 Available under license
1.742 jackson-jaxrs-base 2.8.4
  1.742.1 Available under license
1.743 cacheable-request 6.1.0
  1.743.1 Available under license
1.744 slash 1.0.0 1.0.0
  1.744.1 Available under license
1.745 eslint 3.19.0
  1.745.1 Available under license
1.746 color-name 1.1.4
  1.746.1 Available under license
1.747 accepts 1.2.13 1.2.13
  1.747.1 Available under license
1.748 netty-handler-proxy 4.1.42.Final
1.748.1 Available under license
1.749 is-typedarray 1.0.0
1.749.1 Available under license
1.750 commons-pool 2.6.2
1.751 google-discovery-to-swagger 2.1.0 2.1.0
1.751.1 Available under license
1.752 amcharts3-responsive 1.0.5 1.0.5
1.752.1 Available under license
1.753 safe-buffer 5.1.2
1.753.1 Available under license
1.754 blob 0.0.4 0.0.4
1.754.1 Available under license
1.755 es6-promise 3.3.1 3.3.1
1.755.1 Available under license
1.756 babel-preset-env 1.7.0 1.7.0
1.756.1 Available under license
1.757 to-fast-properties 2.0.0
1.757.1 Available under license
1.758 react-textarea-autosize 6.1.0 6.1.0
1.758.1 Available under license
1.759 commons-logging 1.1.1
1.759.1 Available under license
1.760 jnr-ffi 2.1.7
1.760.1 Available under license
1.761 loader-utils 1.4.0
1.761.1 Available under license
1.762 pinkie 1.0.0 1.0.0
1.762.1 Available under license
1.763 jetty-util 6.1.26
1.764 aopalliance-repackaged 1.3.1.7-5.ph2
1.764.1 Available under license
1.765 path-key 2.0.1
1.765.1 Available under license
1.766 convert-source-map 1.5.1 1.5.1
1.766.1 Available under license
1.767 protobuf-java-util 3.13.0
1.767.1 Available under license
1.768 yargs 3.32.0
1.768.1 Available under license
1.769 mime-pull 1.9.6
  1.769.1 Available under license
1.770 char det 0.7.0
  1.770.1 Available under license
1.771 arr-diff 4.0.0
  1.771.1 Available under license
1.772 crypto-js 4.0.0
  1.772.1 Available under license
1.773 commons-text 1.8
  1.773.1 Available under license
1.774 @babel/plugin-syntax-export-namespace-from 7.8.3
  1.774.1 Available under license
1.775 log4j-over-slf4j 1.7.26
1.776 twill-core 0.6.0-incubating
  1.776.1 Available under license
1.777 path-exists 4.0.0
  1.777.1 Available under license
1.778 amcharts3-dataloader 1.0.19 1.0.19
  1.778.1 Available under license
1.779 punycode 1.4.1 1.4.1
  1.779.1 Available under license
1.780 ms 2.1.3
  1.780.1 Available under license
1.781 expand-brackets 2.1.4
  1.781.1 Available under license
1.782 statuses 1.4.0 1.4.0
  1.782.1 Available under license
1.783 pcre 8.38-3.1
  1.783.1 Available under license
1.784 debug 3.2.6
  1.784.1 Available under license
1.785 process-nextick-args 2.0.1
  1.785.1 Available under license
1.786 listener-collection 1.2.0
  1.786.1 Available under license
1.787 @babel/helper-hoist-variables 7.13.0
  1.787.1 Available under license
1.788 netty-transport-native-epoll 4.1.59.Final
  1.788.1 Available under license
1.789 librest 0.7.93-1
   1.789.1 Available under license
1.790 base64-arraybuffer 0.1.5 0.1.5
   1.790.1 Available under license
1.791 twill-discovery-core 0.6.0-incubating
   1.791.1 Available under license
1.792 gson 2.8.5
1.793 commons-beanutils 1.9.4
   1.793.1 Available under license
1.794 netty 4.1.56.Final
   1.794.1 Available under license
1.795 metrics-jetty 4.1.0
1.796 jansi 1.18
   1.796.1 Available under license
1.797 parquet-encoding 1.10.0
   1.797.1 Available under license
1.798 bson 1.0.9 1.0.9
   1.798.1 Available under license
1.799 chai-http 3.0.0
   1.799.1 Available under license
1.800 snapdragon-util 3.0.1
   1.800.1 Available under license
1.801 type-detect 4.0.8
   1.801.1 Available under license
1.802 is-descriptor 1.0.2
   1.802.1 Available under license
1.803 unpipe 1.0.0 1.0.0
   1.803.1 Available under license
1.804 path-root-regex 0.1.2
   1.804.1 Available under license
1.805 semver 5.7.1
   1.805.1 Available under license
1.806 mime-db 1.47.0 1.47.0
   1.806.1 Available under license
1.807 reflectasm 1.10.1
   1.807.1 Available under license
1.808 jackson 2.12.3
   1.808.1 Available under license
1.809 mimic-fn 2.1.0
1.809.1 Available under license
1.810 jetty-webapp 9.3.27.v20190418
1.811 negotiator 0.5.3
  1.811.1 Available under license
1.812 request 2.88.2
  1.812.1 Available under license
1.813 open-csv 2.3
  1.813.1 Available under license
1.814 babel-helper-regex 6.26.0
  1.814.1 Available under license
1.815 safe-buffer 5.1.2 5.1.2
  1.815.1 Available under license
1.816 has-flag 3.0.0
  1.816.1 Available under license
1.817 arraybuffer.slice 0.0.6 0.0.6
  1.817.1 Available under license
1.818 has-binary 0.1.7 0.1.7
  1.818.1 Available under license
1.819 minimist 0.0.8 0.0.8
  1.819.1 Available under license
1.820 jpam 1.1
  1.820.1 Available under license
1.821 jsesc 0.5.0 0.5.0
  1.821.1 Available under license
1.822 babel-plugin-proposal-dynamic-import 7.13.8
  1.822.1 Available under license
1.823 moment-timezone 0.5.31
  1.823.1 Available under license
1.824 protobuf-java 3.11.1
  1.824.1 Available under license
1.825 is-stream 1.1.0
  1.825.1 Available under license
1.826 cli-width 2.2.1
  1.826.1 Available under license
1.827 globals 9.18.0 9.18.0
  1.827.1 Available under license
1.828 async 1.4.1
  1.828.1 Available under license
1.829 xml-apis 1.4.01
1.829.1 Available under license
1.830 perfmark-api 0.19.0
  1.830.1 Available under license
1.831 hbase-shaded-jetty 3.4.1
  1.831.1 Available under license
1.832 javassist 3.27.0-GA
  1.832.1 Available under license
1.833 netty-codec-http 4.1.42.Final
  1.833.1 Available under license
1.834 babel-plugin-syntax-asnc-functions 6.13.0 6.13.0
  1.834.1 Available under license
1.835 request-local 1.0.5 1.0.5
  1.835.1 Available under license
1.836 expat 2.1.0
  1.836.1 Available under license
1.837 fs.realpath 1.0.0
  1.837.1 Available under license
1.838 twill-api 0.6.0-incubating
  1.838.1 Available under license
1.839 i 0.3.6
  1.839.1 Available under license
1.840 is-equal-shallow 0.1.3
  1.840.1 Available under license
1.841 xtend 4.0.1 4.0.1
  1.841.1 Available under license
1.842 node-readfiles 0.2.0 0.2.0
  1.842.1 Available under license
1.843 yargs 13.3.2
  1.843.1 Available under license
1.844 @babel/helper-skip-transparent-expression-wrappers 7.12.1
  1.844.1 Available under license
1.845 @babel/plugin-syntax-asnc-generators 7.8.4
  1.845.1 Available under license
1.846 debug 3.1.0 3.1.0
  1.846.1 Available under license
1.847 jol-core 0.9
  1.847.1 Available under license
1.848 jsonfile 2.4.0
  1.848.1 Available under license
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>eyes</td>
<td>0.1.8</td>
<td>Available under license</td>
</tr>
<tr>
<td>babel-plugin-transform-regenerator</td>
<td>6.26.0, 6.26.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>hadoop-aws</td>
<td>3.2.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>chalk</td>
<td>1.1.3, 1.1.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>nan</td>
<td>2.10.0, 2.10.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>@babel/plugin-transform-modules-commonjs</td>
<td>7.13.8</td>
<td>Available under license</td>
</tr>
<tr>
<td>flat-cache</td>
<td>1.3.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>guava</td>
<td>30.0-jre</td>
<td>Available under license</td>
</tr>
<tr>
<td>sindresorhus-is</td>
<td>0.14.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>nimbus-jose-jwt</td>
<td>7.9</td>
<td>Available under license</td>
</tr>
<tr>
<td>jackson-annotations</td>
<td>2.11.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>object-assign</td>
<td>4.1.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>has-own-deep</td>
<td>1.1.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>commons-pool</td>
<td>1.6</td>
<td>Available under license</td>
</tr>
<tr>
<td>cassandra-driver-core</td>
<td>3.2.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>jmespath-java</td>
<td>1.11.837</td>
<td>Available under license</td>
</tr>
<tr>
<td>silent-spawn</td>
<td>0.4.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>stream-shift</td>
<td>1.0.1, 1.0.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>extend</td>
<td>3.0.2, 3.0.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>component-bind</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.869 performance-now 2.1.0
  1.869.1 Available under license
1.870 glob 7.1.6
  1.870.1 Available under license
1.871 path-parse 1.0.6
  1.871.1 Available under license
1.872 drill-logical 1.19.0
  1.872.1 Available under license
1.873 netty 4.0.48.Final
  1.873.1 Available under license
1.874 babel-core 6.26.3
  1.874.1 Available under license
1.875 jsreport-pdfjs 0.5.1
  1.875.1 Available under license
1.876 array-flatten 1.1.1
  1.876.1 Available under license
1.877 hikaricp 3.4.2
  1.877.1 Available under license
1.878 netty-codec 4.1.42.Final
  1.878.1 Available under license
1.879 to-regex-range 2.1.1
  1.879.1 Available under license
1.880 boost 1.60
  1.880.1 Available under license
1.881 websocket-common 9.3.27.v20190418
1.882 babel-plugin-syntax-object-rest-spread 6.13.0
  1.882.1 Available under license
1.883 aws-java-sdk-sts 1.11.375
  1.883.1 Available under license
1.884 har-schema 2.0.0
  1.884.1 Available under license
1.885 stream-to-array 2.3.0
  1.885.1 Available under license
1.886 jgrapht-core 1.3.0
  1.886.1 Available under license
1.887 for-own 0.1.5
  1.887.1 Available under license
1.888 vary 1.1.2
  1.888.1 Available under license
1.889 twill-discovery-api 0.6.0-incubating
    1.889.1 Available under license
1.890 libdrm 2.4.91 2~16.04.1
1.891 jackson-jaxrs-base 2.9.10
    1.891.1 Available under license
1.892 js-tokens 3.0.2
    1.892.1 Available under license
1.893 p-try 1.0.0
    1.893.1 Available under license
1.894 spdx-correct 3.1.1
    1.894.1 Available under license
1.895 glib 2.48.2-0ubuntu1
    1.895.1 Available under license
1.896 normalize-package-data 2.5.0
    1.896.1 Available under license
1.897 chai 4.3.4
    1.897.1 Available under license
1.898 object-visit 1.0.1
    1.898.1 Available under license
1.899 backport-util-concurrent 3.1
    1.899.1 Available under license
1.900 set-blocking 2.0.0
    1.900.1 Available under license
1.901 ws 1.1.1 1.1.1
    1.901.1 Available under license
1.902 iceberg-common 0.8.0-incubating
    1.902.1 Available under license
1.903 strip-eof 1.0.0
    1.903.1 Available under license
1.904 shared-mime-info 1.5 2ubuntu0.2
    1.904.1 Available under license
1.905 zlib 1.2.5
    1.905.1 Available under license
1.906 jersey-media-jaxb 2.32
    1.906.1 Available under license
1.907 @babel/plugin-syntax-json-strings 7.8.3
    1.907.1 Available under license
1.908 asm 5.0.3
    1.908.1 Available under license
<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>os-homedir</td>
<td>1.0.2</td>
<td>1.0.2 Available under license</td>
</tr>
<tr>
<td>tephra-core</td>
<td>0.6.0</td>
<td>0.6.0 Available under license</td>
</tr>
<tr>
<td>regenerator-runtime</td>
<td>0.13.3</td>
<td>0.13.3 Available under license</td>
</tr>
<tr>
<td>bcel</td>
<td>2.7.2</td>
<td>2.7.2 Available under license</td>
</tr>
<tr>
<td>natural-compare</td>
<td>1.4.0</td>
<td>1.4.0 Available under license</td>
</tr>
<tr>
<td>jsreport-express</td>
<td>2.8.1</td>
<td>2.8.1 Available under license</td>
</tr>
<tr>
<td>os-locale</td>
<td>1.4.0</td>
<td>1.4.0 Available under license</td>
</tr>
<tr>
<td>apib2swagger</td>
<td>1.12.0</td>
<td>1.12.0 Available under license</td>
</tr>
<tr>
<td>hsql-db</td>
<td>1.8.0.10</td>
<td>1.8.0.10 Available under license</td>
</tr>
<tr>
<td>qs</td>
<td>6.10.1</td>
<td>6.10.1 Available under license</td>
</tr>
<tr>
<td>calcite-elasticsearch</td>
<td>1.21.0-drill-r2</td>
<td>1.21.0 Available under license</td>
</tr>
<tr>
<td>hbase-prefix-tree</td>
<td>2.0.0-alpha4</td>
<td>2.0.0 Available under license</td>
</tr>
<tr>
<td>string-width</td>
<td>2.1.1</td>
<td>2.1.1 Available under license</td>
</tr>
<tr>
<td>popperjs</td>
<td>1.15.0</td>
<td>1.15.0 Available under license</td>
</tr>
<tr>
<td>p-is-promise</td>
<td>2.1.0</td>
<td>2.1.0 Available under license</td>
</tr>
<tr>
<td>duplexer2</td>
<td>0.1.4</td>
<td>0.1.4 Available under license</td>
</tr>
<tr>
<td>ua-parser-js</td>
<td>0.7.18</td>
<td>0.7.18 Available under license</td>
</tr>
<tr>
<td>jsr305</td>
<td>3.0.1</td>
<td>3.0.1 Available under license</td>
</tr>
<tr>
<td>caseless</td>
<td>0.12.0</td>
<td>0.12.0 Available under license</td>
</tr>
<tr>
<td>component-inherit</td>
<td>0.0.3</td>
<td>0.0.3 Available under license</td>
</tr>
</tbody>
</table>
1.929 schema-utils 0.4.7  
   1.929.1 Available under license

1.930 esprima 1.2.2 1.2.2  
   1.930.1 Available under license

1.931 ajv 5.5.2  
   1.931.1 Available under license

1.932 yallist 2.1.2  
   1.932.1 Available under license

1.933 base 0.11.2  
   1.933.1 Available under license

1.934 lodash.uniq 4.5.0  
   1.934.1 Available under license

1.935 jnr-constants 0.9.9  
   1.935.1 Available under license

1.936 deep-extend 0.6.0  
   1.936.1 Available under license

1.937 api-spec-converter 2.12.0 2.12.2  
   1.937.1 Available under license

1.938 camelcase 5.0.0  
   1.938.1 Available under license

1.939 hbase-asyncfs 2.4.2  
   1.939.1 Available under license

1.940 strip-json-comments 2.0.1  
   1.940.1 Available under license

1.941 swagger-jersey2-jaxrs 1.5.0

   1.942.1 Available under license

1.943 is-windows 1.0.2  
   1.943.1 Available under license

1.944 @types/node 12.12.68  
   1.944.1 Available under license

1.945 ansistyles 0.1.3  
   1.945.1 Available under license

1.946 jcodings 1.0.55  
   1.946.1 Available under license

1.947 toggle-selection 1.0.6  
   1.947.1 Available under license

1.948 keygrip 1.0.3  
   1.948.1 Available under license
1.949 picocli 4.5.2
   1.949.1 Available under license
1.950 lodash-map 4.6.0
   1.950.1 Available under license
1.951 util-deprecate 1.0.2 1.0.2
   1.951.1 Available under license
1.952 lodash.once 4.1.1 4.1.1
   1.952.1 Available under license
1.953 mongodb-driver-core 4.2.3
   1.953.1 Available under license
1.954 hogan 3.0.2
   1.954.1 Available under license
1.955 react-json-view 1.19.1 1.19.1
   1.955.1 Available under license
1.956 safe-json-stringify 1.2.0 1.2.0
   1.956.1 Available under license
1.957 javax-inject 2.5.0-b32
   1.957.1 Available under license
1.958 is-number 7.0.0
   1.958.1 Available under license
1.959 isarray 0.0.1
   1.959.1 Available under license
1.960 dashdash 1.14.1 1.14.1
   1.960.1 Available under license
1.961 alsa 1.1.0-0ubuntu1
   1.961.1 Available under license
1.962 codemirror 5.59.2 5.59.2
   1.962.1 Available under license
1.963 global-modules 1.0.0
   1.963.1 Available under license
1.964 unique-string 2.0.0
   1.964.1 Available under license
1.965 ncp 2.0.0
   1.965.1 Available under license
1.966 kafka-clients 2.8.0
   1.966.1 Available under license
1.967 sshpk 1.16.1 1.16.1
   1.967.1 Available under license
1.968 form-data 1.0.1 1.0.1
1.968.1 Available under license
1.969 **string-width** 4.2.2 4.2.2
1.969.1 Available under license
1.970 **hbase-replication** 2.4.2
1.970.1 Available under license
1.971 **istanbuljs-load-nyc-config** 1.1.0
1.971.1 Available under license
1.972 **akka** 2.5.29
1.972.1 Available under license
1.973 **commons-crypto** 1.0.0
1.974 **jsreport-exceljs** 3.7.1
1.974.1 Available under license
1.975 **cors** 2.8.5
1.975.1 Available under license
1.976 **referrer-policy** 1.0.0 1.0.0
1.976.1 Available under license
1.977 **lodash-es** 4.17.15 4.17.15
1.977.1 Available under license
1.978 **jline-reader** 3.12.1
1.978.1 Available under license
1.979 **combined-stream** 1.0.8 1.0.8
1.979.1 Available under license
1.980 **debug** 2.2.0
1.980.1 Available under license
1.981 **jsreport-html-to-xlsx** 2.8.3
1.981.1 Available under license
1.982 **jackson-databind** 2.11.4
1.982.1 Available under license
1.983 **snake-yaml** 1.26
1.983.1 Available under license
1.984 **fresh** 0.5.2
1.984.1 Available under license
1.985 **hive-shims** 3.1.2
1.985.1 Available under license
1.986 **grpc-stub** 1.27.1
1.986.1 Available under license
1.987 **isbinaryfile** 3.0.3
1.987.1 Available under license
1.988 **guice-assistedinject** 3.0
1.988.1 Available under license
1.989 babel-helper-explode-assignable-expression 6.24.1
1.989.1 Available under license
1.990 babel-plugin-transform-es2015-object-super 6.24.1
1.990.1 Available under license
1.991 parseqs 0.0.5
1.991.1 Available under license
1.992 is-binary-path 2.1.0
1.992.1 Available under license
1.993 helmet-csp 1.2.2
1.993.1 Available under license
1.994 lodash.debounce 4.0.8
1.994.1 Available under license
1.995 frameguard 2.0.0
1.995.1 Available under license
1.996 has-binary2 1.0.3
1.996.1 Available under license
1.997 jetty-http 9.4.17.v20190418
1.997.1 Available under license
1.998 formidable 1.2.2
1.998.1 Available under license
1.999 snake-yaml 1.12
1.999.1 Available under license
1.1000 redux 3.7.2 3.7.2
1.1000.1 Available under license
1.1001 @babel/helper-split-export-declaration 7.12.13
1.1001.1 Available under license
1.1002 jetty 9.4.41.v20210516
1.1002.1 Available under license
1.1003 uri-js 4.2.2 4.2.2
1.1003.1 Available under license
1.1004 asm-commons 7.3.1
1.1004.1 Available under license
1.1005 drill-format-spss 1.19.0
1.1005.1 Available under license
1.1006 zlib 1.2.11
1.1006.1 Available under license
1.1007 is-glob 2.0.1
1.1007.1 Available under license
1.1008 is-plain-object 2.0.4 2.0.4
  1.1008.1 Available under license
1.1009 drill-mongo-storage 1.19.0
  1.1009.1 Available under license
1.1010 parquet-format 2.4.0
  1.1010.1 Available under license
  1.1011.1 Available under license
1.1012 strip-ansi 6.0.0
  1.1012.1 Available under license
1.1013 websocket-client 9.4.17.v20190418
  1.1013.1 Available under license
1.1014 bson 1.1.6 1.1.6
  1.1014.1 Available under license
1.1015 mem 4.3.0
  1.1015.1 Available under license
1.1016 eclipse-egit-github-core 2.1.5
  1.1016.1 Available under license
1.1017 proj4j 0.1.0
  1.1017.1 Available under license
1.1018 has-cors 1.1.0
  1.1018.1 Available under license
1.1019 jsonpointer 4.0.1 4.0.1
  1.1019.1 Available under license
1.1020 strip-ansi 3.0.1 3.0.1
  1.1020.1 Available under license
1.1021 browserslist 3.2.8 3.2.8
  1.1021.1 Available under license
1.1022 check-error 1.0.2
  1.1022.1 Available under license
1.1023 yeast 0.1.2
  1.1023.1 Available under license
1.1024 optionator 0.8.2
  1.1024.1 Available under license
1.1025 is-relative 1.0.0
  1.1025.1 Available under license
1.1026 mkdirp 0.5.5 0.5.5
  1.1026.1 Available under license
1.1027 babel-plugin-transform-es2015-unicode-regex 6.24.1
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antelope</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hk2-api</td>
<td>2.5.0-b32</td>
<td>Available under license</td>
</tr>
<tr>
<td>babel-messages</td>
<td>6.23.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>base16</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>failureaccess</td>
<td>1.0.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>debug</td>
<td>2.3.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>jetty-client</td>
<td>9.4.17.v20190418</td>
<td>Available under license</td>
</tr>
<tr>
<td>append-field</td>
<td>0.1.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>jersey-server</td>
<td>2.5.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>uuid</td>
<td>3.3.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>resourcelocator</td>
<td>1.0.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>pkg-dir</td>
<td>3.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>stdio</td>
<td>0.2.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>swagger-annotations</td>
<td>2.0.7</td>
<td>Available under license</td>
</tr>
<tr>
<td>zstd</td>
<td>1.4.9</td>
<td>Available under license</td>
</tr>
<tr>
<td>jetty-servlet</td>
<td>9.4.17.v20190418</td>
<td>Available under license</td>
</tr>
<tr>
<td>httpcomponents-client</td>
<td>4.5.6</td>
<td>Available under license</td>
</tr>
<tr>
<td>regex-cache</td>
<td>0.4.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>bluebird</td>
<td>3.7.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>on-finished</td>
<td>2.3.0</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.1047 has-yarn 2.1.0
   1.1047.1 Available under license
1.1048 protobuf 2.6.1-1.3
   1.1048.1 Available under license
1.1049 js-tokens 4.0.0
   1.1049.1 Available under license
1.1050 resolve-from 2.0.0
   1.1050.1 Available under license
1.1051 jackson-annotations 2.12.1
   1.1051.1 Available under license
1.1052 serializator 1.0.2
   1.1052.1 Available under license
1.1053 duplexer 0.1.1 0.1.1
   1.1053.1 Available under license
1.1054 cookie-signature 1.0.6 1.0.6
   1.1054.1 Available under license
1.1055 abbrev 1.0.9
   1.1055.1 Available under license
1.1056 ant-antlr 1.10.9
   1.1056.1 Available under license
1.1057 urix 0.1.0
   1.1057.1 Available under license
1.1058 babel-register 6.26.0
   1.1058.1 Available under license
1.1059 normalize-url 4.5.0
   1.1059.1 Available under license
1.1060 babel-helper-remap-async-to-generator 6.24.1
   1.1060.1 Available under license
1.1061 has-value 1.0.0
   1.1061.1 Available under license
1.1062 @babel/plugin-proposal-export-namespace-from 7.12.13
   1.1062.1 Available under license
1.1063 validate-npm-package-license 3.0.4
   1.1063.1 Available under license
1.1064 browserslist 3.2.8
   1.1064.1 Available under license
1.1065 netty-transport-native-epoll 4.1.45.Final
   1.1065.1 Available under license
1.1066 diff2html 2.5.0
1.1066.1 Available under license

1.1067 yaml 1.10.2 1.10.2
   1.1067.1 Available under license

1.1068 multer 1.3.1 1.3.1
   1.1068.1 Available under license

1.1069 ecc-jsbn 0.1.2
   1.1069.1 Available under license

1.1070 jetty-util 9.4.41.v20210516
   1.1070.1 Available under license

1.1071 slf4j 1.7.28

1.1072 ienoopen 1.0.0 1.0.0
   1.1072.1 Available under license

1.1073 guava 28.2-jre
   1.1073.1 Available under license

1.1074 babel-plugin-transform-async-to-generator 6.24.1
   1.1074.1 Available under license

1.1075 log4j-slf4j-impl 2.14.0
   1.1075.1 Available under license

1.1076 validator 10.11.0 10.11.0
   1.1076.1 Available under license

1.1077 lodash.isnil 4.0.0
   1.1077.1 Available under license

1.1078 imurmurhash 0.1.4
   1.1078.1 Available under license

1.1079 merge2 1.3.0
   1.1079.1 Available under license

1.1080 make-iterator 1.0.1
   1.1080.1 Available under license

1.1081 jnr-posix 3.0.27
   1.1081.1 Available under license

1.1082 bootstrap 3.4.1 3.4.1
   1.1082.1 Available under license

1.1083 property-expr 2.0.4 2.0.4
   1.1083.1 Available under license

1.1084 unicode-trie 0.3.1
   1.1084.1 Available under license

1.1085 levn 0.3.0 0.3.0
   1.1085.1 Available under license

1.1086 bytes 3.0.0
1.1086.1 Available under license

1.1087 **responselike 1.0.2**
   1.1087.1 Available under license

1.1088 **lcid 2.0.0**
   1.1088.1 Available under license

1.1089 **path-exists 4.0.0 4.0.0**
   1.1089.1 Available under license

1.1090 **avatica-metrics 1.17.0**
   1.1090.1 Available under license

1.1091 **socket.io-client 1.7.4**
   1.1091.1 Available under license

1.1092 **jkeylockmanager 2.1.0**
   1.1092.1 Available under license

1.1093 **hdrhistogram 2.1.12**
   1.1093.1 Available under license

1.1094 **babel-plugin-proposal-nullish-coalescing-operator 7.13.8**
   1.1094.1 Available under license

1.1095 **npm-run-path 2.0.2**
   1.1095.1 Available under license

1.1096 **drill-storage-hbase 1.19.0**
   1.1096.1 Available under license

1.1097 **vary 1.0.1**
   1.1097.1 Available under license

1.1098 **javaparser-core 3.18.0**
   1.1098.1 Available under license

1.1099 **v8 8.8.8.8**
   1.1099.1 Available under license

1.1100 **de.huxhorn.lilith.data.eventsource 0.9.44**
   1.1100.1 Available under license

1.1101 **zip-stream 1.2.0**
   1.1101.1 Available under license

1.1102 **nock 11.9.1**
   1.1102.1 Available under license

1.1103 **@babel/helper-get-function-arity 7.10.4**
   1.1103.1 Available under license

1.1104 **source-map 0.5.7**
   1.1104.1 Available under license

1.1105 **htrace-core4 4.2.0-incubating**
   1.1105.1 Available under license
1.1106 source-map-support 0.4.18
   1.1106.1 Available under license

1.1107 fill-range 2.2.4
   1.1107.1 Available under license

1.1108 esutils 2.0.3 2.0.3
   1.1108.1 Available under license

1.1109 fast-levenshtein 2.0.6 2.0.6
   1.1109.1 Available under license

1.1110 csv-parse 4.8.2 4.8.2
   1.1110.1 Available under license

1.1111 bluebird 3.5.4
   1.1111.1 Available under license

1.1112 connect 3.4.1 3.4.1
   1.1112.1 Available under license

1.1113 asm 3.2
   1.1113.1 Available under license

1.1114 jooq 3.13.1
   1.1114.1 Available under license

1.1115 send 0.13.1 0.13.1
   1.1115.1 Available under license

1.1116 socket.io 1.7.4 1.7.4
   1.1116.1 Available under license

1.1117 compression 1.7.4
   1.1117.1 Available under license

1.1118 datanucleus-core 4.1.17
   1.1118.1 Available under license

1.1119 de.huxhorn.lilith.logback.converter-classic 0.9.44
   1.1119.1 Available under license

1.1120 drill-jdbc-all 1.19.0
   1.1120.1 Available under license

1.1121 pascalcase 0.1.1
   1.1121.1 Available under license

1.1122 batik-anim 1.13
   1.1122.1 Available under license

1.1123 request 2.87.0 2.87.0
   1.1123.1 Available under license

1.1124 keypress 0.1.0
   1.1124.1 Available under license

1.1125 redux-saga 0.16.0 0.16.0
1.1125.1 Available under license

1.1126 tslib 1.14.1
1.1126.1 Available under license

1.1127 entities 1.1.1 1.1.1
1.1127.1 Available under license

1.1128 drill-rpc 1.19.0
1.1128.1 Available under license

1.1129 json5 0.5.1 0.5.1
1.1129.1 Available under license

1.1130 ecdsa-sig-formatter 1.0.11
1.1130.1 Available under license

1.1131 query-string 7.0.0 7.0.0
1.1131.1 Available under license

1.1132 react-highlight-words 0.16.0 0.16.0
1.1132.1 Available under license

1.1133 hbase-hadoop-compat 2.0.0-alpha4
1.1133.1 Available under license

1.1134 resolve 1.9.0
1.1134.1 Available under license

1.1135 kerb-common 1.0.0
1.1135.1 Available under license

1.1136 jsbn 1.1.0
1.1136.1 Available under license

1.1137 fs 0.0.2 0.0.2
1.1137.1 Available under license

1.1138 disruptor 3.4.2

1.1139 compress-lzf 1.1
1.1139.1 Available under license

1.1140 wtf-8 1.0.0 1.0.0
1.1140.1 Available under license

1.1141 babel-types 6.26.0
1.1141.1 Available under license

1.1142 jsreport-data 2.4.0
1.1142.1 Available under license

1.1143 xtend 4.0.2
1.1143.1 Available under license

1.1144 @babel/plugin-transform-object-super 7.12.13
1.1144.1 Available under license

1.1145 ajv 6.10.2 6.10.2
1.1145.1 Available under license

1.1146 zlib 1.2.1
   1.1146.1 Available under license

1.1147 kotlin 1.3.61
   1.1147.1 Available under license

1.1148 babel-plugin-syntax-nullish-coalescing-operator 7.8.3
   1.1148.1 Available under license

1.1149 referrer-policy 1.0.0
   1.1149.1 Available under license

1.1150 batik-slideshow 1.13
   1.1150.1 Available under license

1.1151 engine.io-parser 1.3.2
   1.1151.1 Available under license

1.1152 mime 2.4.4
   1.1152.1 Available under license

1.1153 yargs-parser 18.1.3
   1.1153.1 Available under license

1.1154 avatica 1.17.0
   1.1154.1 Available under license

1.1155 gson-fire 1.5.1
   1.1155.1 Available under license

1.1156 cliui 6.0.0
   1.1156.1 Available under license

1.1157 dashify 0.2.2
   1.1157.1 Available under license

1.1158 @babel/generator 7.11.5
   1.1158.1 Available under license

1.1159 isomorphic-fetch 2.2.1 2.2.1
   1.1159.1 Available under license

1.1160 json-schema-traverse 0.3.1 0.3.1
   1.1160.1 Available under license

1.1161 lodash-es 4.17.10 4.17.10
   1.1161.1 Available under license

1.1162 tpch-sample-data 1.19.0
   1.1162.1 Available under license

1.1163 @babel/traverse 7.13.13
   1.1163.1 Available under license

1.1164 bson 3.12.4
   1.1164.1 Available under license
1.1165 js-yaml 3.14.0
  1.1165.1 Available under license
1.1166 has-bigints 1.0.1
  1.1166.1 Available under license
1.1167 is-yarn-global 0.3.0
  1.1167.1 Available under license
1.1168 jsreport-core 2.10.1
  1.1168.1 Available under license
1.1169 json-schema-traverse 0.3.1
  1.1169.1 Available under license
1.1170 babel-plugin-transform-es2015-for-of 6.23.0 6.23.0
  1.1170.1 Available under license
1.1171 logback-core 1.2.3
  1.1171.1 Available under license
1.1172 orc-core 1.5.6
  1.1172.1 Available under license
1.1173 woodstox-core 5.0.3
  1.1173.1 Available under license
1.1174 is-number 2.1.0
  1.1174.1 Available under license
1.1175 extract-zip 1.7.0
  1.1175.1 Available under license
1.1176 tunnel-agent 0.6.0
  1.1176.1 Available under license
1.1177 type-is 1.6.18
  1.1177.1 Available under license
1.1178 babel-helper-optimise-call-expression 6.24.1
  1.1178.1 Available under license
1.1179 clean-stack 2.2.0
  1.1179.1 Available under license
1.1180 hive-llap-common 3.1.2
  1.1180.1 Available under license
1.1181 bytes 2.3.0 2.3.0
  1.1181.1 Available under license
1.1182 app-root-path 2.0.1
  1.1182.1 Available under license
1.1183 p-limit 2.3.0
  1.1183.1 Available under license
1.1184 kafka-schema-registry-client 6.1.1
1.1185 @babel/plugin-proposal-private-methods 7.13.0
   1.1185.1 Available under license

1.1186 libxext 1.3.3-1
   1.1186.1 Available under license

1.1187 proto-google-common-protos 1.17.0
   1.1187.1 Available under license

1.1188 bluebird 3.7.2 3.7.2
   1.1188.1 Available under license

1.1189 klaw 1.3.1
   1.1189.1 Available under license

1.1190 commons-cli 1.4
   1.1190.1 Available under license

1.1191 jsreport-authentication 2.6.1
   1.1191.1 Available under license

1.1192 jsreport-public-templates 2.2.0
   1.1192.1 Available under license

1.1193 y18n 4.0.3 4.0.3
   1.1193.1 Available under license

1.1194 generate-function 2.0.0 2.0.0
   1.1194.1 Available under license

1.1195 babel-plugin-transform-es2015-modules-systemjs 6.24.1
   1.1195.1 Available under license

1.1196 paranamer 2.5.6

1.1197 re2j 1.1

1.1198 send 0.13.2
   1.1198.1 Available under license

1.1199 types-color-name 1.1.1
   1.1199.1 Available under license

1.1200 babel-helper-replace-supers 6.24.1 6.24.1
   1.1200.1 Available under license

1.1201 memory 0.9.0
   1.1201.1 Available under license

1.1202 babel-helper-explode-assignable-expression 6.24.1 6.24.1
   1.1202.1 Available under license

1.1203 component-emitter 1.3.0 1.3.0
   1.1203.1 Available under license

1.1204 node-fetch 1.7.3 1.7.3
   1.1204.1 Available under license
1.1205 generate-object-property 1.2.0 1.2.0
1.1205.1 Available under license

1.1206 define-property 0.2.5
1.1206.1 Available under license

1.1207 uzaygezen-core 0.2
1.1207.1 Available under license

1.1208 json-schema-ref-parser 3.3.1 3.3.1
1.1208.1 Available under license

1.1209 es6-map 0.1.5
1.1209.1 Available under license

1.1210 accepts 1.3.3 1.3.3
1.1210.1 Available under license

1.1211 negotiator 0.6.2
1.1211.1 Available under license

1.1212 private 0.1.8 0.1.8
1.1212.1 Available under license

1.1213.1 Available under license

1.1214 pseudomap 1.0.2
1.1214.1 Available under license

1.1215 ms 2.1.1 2.1.1
1.1215.1 Available under license

1.1216 js-tokens 4.0.0 4.0.0
1.1216.1 Available under license

1.1217 revalidator 0.1.8
1.1217.1 Available under license

1.1218 liftoff 3.0.0
1.1218.1 Available under license

1.1219 form-data 2.5.1 2.5.1
1.1219.1 Available under license

1.1220 lodash-pickby 4.6.0
1.1220.1 Available under license

1.1221 package-json 6.5.0
1.1221.1 Available under license

1.1222 cairo 1.14.6 1
1.1222.1 Available under license

1.1223 @babel/plugin-proposal-async-generator-functions 7.13.8
1.1223.1 Available under license

1.1224 json 1.8
Open Source Used In DNAC Platform Antelope
1.1264.1 Available under license
1.1265 calcite-linq4j 1.26.0
   1.1265.1 Available under license
1.1266 ms 0.7.1
   1.1266.1 Available under license
1.1267 @babel/plugin-syntax-numeric-separator 7.10.4
   1.1267.1 Available under license
1.1268 cli-boxes 2.2.1
   1.1268.1 Available under license
1.1269 xsdlib 2013.6.1
   1.1269.1 Available under license
1.1270 react-ace 5.10.0 5.10.0
   1.1270.1 Available under license
1.1271 jackson-module-paranamer 2.10.5
   1.1271.1 Available under license
1.1272 byte-buddy-agent 1.9.3
   1.1272.1 Available under license
1.1273 hbase-protocol-shaded 2.4.2
   1.1273.1 Available under license
1.1274 @babel/plugin-transform-modules-systemjs 7.13.8
   1.1274.1 Available under license
1.1275 velocity 2.0
   1.1275.1 Available under license
1.1276 babel-core 6.26.3 6.26.3
   1.1276.1 Available under license
1.1277 websocket-servlet 9.3.27.v20190418
1.1278 swagger-models 2.0.7
   1.1278.1 Available under license
1.1279 hbase-hadoop-compat 2.4.2
   1.1279.1 Available under license
1.1280 babel-helpers 6.24.1
   1.1280.1 Available under license
1.1281 jsreport-import-export 2.1.1
   1.1281.1 Available under license
1.1282 fast-deep-equal 3.1.3 3.1.3
   1.1282.1 Available under license
1.1283 utile 0.3.0
   1.1283.1 Available under license
1.1284 babel-plugin-transform-es2015-function-name 6.24.1
1.1284.1 Available under license
1.1285 snappy-java 1.1.2.6
1.1286 contains-path 0.1.0
    1.1286.1 Available under license
1.1287 etag 1.7.0 1.7.0
    1.1287.1 Available under license
1.1288 traverse 0.6.6 0.6.6
    1.1288.1 Available under license
1.1289 htmlparser2 3.9.2 3.9.2
    1.1289.1 Available under license
1.1290 sanitize-html-react 1.13.0 1.13.0
    1.1290.1 Available under license
1.1291 object-component 0.0.3
    1.1291.1 Available under license
1.1292 commons-compiler 3.0.11
1.1293 global-prefix 1.0.2
    1.1293.1 Available under license
1.1294 papa-parse 5.1.1 5.1.1
    1.1294.1 Available under license
1.1295 hikaricp-java7 2.4.13
1.1296 expat 2.1.0 7ubuntu0.16.04.5
    1.1296.1 Available under license
1.1297 commons-digester 1.8
1.1298 form-data 2.3.3
    1.1298.1 Available under license
1.1299 bluebird 3.5.1 3.5.1
    1.1299.1 Available under license
1.1300 sac 1.3
    1.1300.1 Available under license
1.1301 lodash-max 4.0.1
    1.1301.1 Available under license
1.1302 inquirer 0.12.0
    1.1302.1 Available under license
1.1303 rimraf 2.6.3
    1.1303.1 Available under license
1.1304 reap2 1.0.1
    1.1304.1 Available under license
1.1305 to-regex 3.0.2
    1.1305.1 Available under license
1.1306 test-exclude 6.0.0
   1.1306.1 Available under license
1.1307 to-utf8 0.0.1
   1.1307.1 Available under license
1.1308 log4j-api 2.14.0
   1.1308.1 Available under license
1.1309 upath 1.2.0
   1.1309.1 Available under license
1.1310 argparse 0.7.0
1.1311 debug 4.3.2
   1.1311.1 Available under license
1.1312 async 2.6.0
   1.1312.1 Available under license
1.1313 source-map 0.6.1
   1.1313.1 Available under license
1.1314 mute-stream 0.0.5
   1.1314.1 Available under license
1.1315 to-buffer 1.1.1
   1.1315.1 Available under license
1.1316 babel-plugin-transform-es2015-typeof-symbol 6.23.0 6.23.0
   1.1316.1 Available under license
1.1317 atk 2.18.1-2ubuntu1
   1.1317.1 Available under license
1.1318 serve-static 1.13.2 1.13.2
   1.1318.1 Available under license
1.1319 boxen 4.2.0
   1.1319.1 Available under license
1.1320 jersey-container-jetty-http 2.25.1
   1.1320.1 Available under license
1.1321 parseqs 0.0.5 0.0.5
   1.1321.1 Available under license
1.1322 isarray 1.0.0
   1.1322.1 Available under license
1.1323 find-up 4.1.0
   1.1323.1 Available under license
1.1324 parquet-hadoop-bundle 1.10.0
   1.1324.1 Available under license
1.1325 groovy-macro 3.0.8
   1.1325.1 Available under license
1.1326 murmur 1.0.0
  1.1326.1 Available under license
1.1327 regjsparser 0.6.9
  1.1327.1 Available under license
1.1328 profiler 1.0.2
1.1329 scheduler 0.19.1 0.19.1
  1.1329.1 Available under license
1.1330 resolve-from 1.0.1
  1.1330.1 Available under license
1.1331 jaeger-core 0.23.0
  1.1331.1 Available under license
1.1332 passport-strategy 1.0.0
  1.1332.1 Available under license
1.1333 websocket-server 9.4.17.v20190418
  1.1333.1 Available under license
1.1334 jsreport-base 2.0.2
  1.1334.1 Available under license
1.1335 has-cors 1.1.0 1.1.0
  1.1335.1 Available under license
1.1336 copy-to-clipboard 3.3.1
  1.1336.1 Available under license
1.1337 protostuff-api 1.7.1
  1.1337.1 Available under license
1.1338 influx 5.0.7 5.0.7
  1.1338.1 Available under license
1.1339 readable-stream 1.1.14 1.1.14
  1.1339.1 Available under license
1.1340 jetty-http 9.4.41.v20210516
  1.1340.1 Available under license
1.1341 rechoir 0.6.2
  1.1341.1 Available under license
1.1342 p-try 2.2.0 2.2.0
  1.1342.1 Available under license
1.1343 concat-stream 1.6.2 1.6.2
  1.1343.1 Available under license
1.1344 jsreport-pdf-utils 1.10.1
  1.1344.1 Available under license
1.1345 jersey-client 2.25.1
  1.1345.1 Available under license
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1346</td>
<td>depd 1.1.0</td>
<td>1.1346.1 Available under license</td>
</tr>
<tr>
<td>1.1347</td>
<td>colorette 1.2.2</td>
<td>1.1347.1 Available under license</td>
</tr>
<tr>
<td>1.1348</td>
<td>aircompressor 0.10</td>
<td></td>
</tr>
<tr>
<td>1.1349</td>
<td>ini 1.3.5</td>
<td>1.1349.1 Available under license</td>
</tr>
<tr>
<td>1.1350</td>
<td>babel-helper-create-class-features-plugin 7.13.11</td>
<td>1.1350.1 Available under license</td>
</tr>
<tr>
<td>1.1351</td>
<td>js-string-escape 1.0.1</td>
<td>1.1351.1 Available under license</td>
</tr>
<tr>
<td>1.1352</td>
<td>trim-right 1.0.1</td>
<td>1.1352.1 Available under license</td>
</tr>
<tr>
<td>1.1353</td>
<td>joi 6.10.1</td>
<td>1.1353.1 Available under license</td>
</tr>
<tr>
<td>1.1354</td>
<td>babel-helper-get-function-arity 6.24.1</td>
<td>1.1354.1 Available under license</td>
</tr>
<tr>
<td>1.1355</td>
<td>is-absolute-url 2.1.0</td>
<td>1.1355.1 Available under license</td>
</tr>
<tr>
<td>1.1356</td>
<td>libthai 0.1.24 2</td>
<td>1.1356.1 Available under license</td>
</tr>
<tr>
<td>1.1357</td>
<td>lodash-repeat 4.1.0</td>
<td>1.1357.1 Available under license</td>
</tr>
<tr>
<td>1.1358</td>
<td>bunyan-format 0.2.1 0.2.1</td>
<td>1.1358.1 Available under license</td>
</tr>
<tr>
<td>1.1359</td>
<td>jws 3.1.5 3.1.5</td>
<td>1.1359.1 Available under license</td>
</tr>
<tr>
<td>1.1360</td>
<td>node-forge 0.10.0</td>
<td>1.1360.1 Available under license</td>
</tr>
<tr>
<td>1.1361</td>
<td>latest-version 5.1.0</td>
<td>1.1361.1 Available under license</td>
</tr>
<tr>
<td>1.1362</td>
<td>isstream 0.1.2 0.1.2</td>
<td>1.1362.1 Available under license</td>
</tr>
<tr>
<td>1.1363</td>
<td>react-base16-styling 0.6.0 0.6.0</td>
<td>1.1363.1 Available under license</td>
</tr>
<tr>
<td>1.1364</td>
<td>batik-squiggle-ext 1.13</td>
<td>1.1364.1 Available under license</td>
</tr>
<tr>
<td>1.1365</td>
<td>hide-powered-by 1.0.0 1.0.0</td>
<td>1.1365.1 Available under license</td>
</tr>
</tbody>
</table>
1.1366 ignore-by-default 1.0.1
  1.1366.1 Available under license
1.1367 at-spi2-core 2.18.3-4ubuntu1
  1.1367.1 Available under license
1.1368 ncp 1.0.1
  1.1368.1 Available under license
1.1369 define-property 1.0.0
  1.1369.1 Available under license
1.1370 buffer-from 1.1.1 1.1.1
  1.1370.1 Available under license
1.1371 lodash.get 4.4.2 4.4.2
  1.1371.1 Available under license
1.1372 loader-utils 1.2.3 1.2.3
  1.1372.1 Available under license
1.1373 jetty 9.3.27.v20190418
  1.1373.1 Available under license
1.1374 eventemitter2 0.4.14
  1.1374.1 Available under license
1.1375 opentracing-util 0.31.0
1.1376 ansistyles 0.1.3 0.1.3
  1.1376.1 Available under license
1.1377 libxdmcp 1.1.2-1.1
  1.1377.1 Available under license
1.1378 foreground-child 2.0.0
  1.1378.1 Available under license
1.1379 parquet-common 1.10.0
  1.1379.1 Available under license
1.1380 socket.io 1.7.4
  1.1380.1 Available under license
1.1381 serve-favicon 2.5.0
  1.1381.1 Available under license
1.1382 unorm 1.6.0
  1.1382.1 Available under license
1.1383 prepend-http 1.0.4 1.0.4
  1.1383.1 Available under license
1.1384 readdirp 3.3.0
  1.1384.1 Available under license
1.1385 javax-annotation-api 1.2
  1.1385.1 Available under license
1.1386 ms 0.7.2
  1.1386.1 Available under license

1.1387 objenesis 2.6
  1.1387.1 Available under license

1.1388 xmldom 0.2.1
  1.1388.1 Available under license

1.1389 drill-jdbc-storage 1.19.0
  1.1389.1 Available under license

1.1390 forever-agent 0.6.1 0.6.1
  1.1390.1 Available under license

1.1391 zip 1.2.0
  1.1391.1 Available under license

1.1392 parquet-column 1.12.0
  1.1392.1 Available under license

1.1393 jackson-datatype-jdk8 2.10.5
  1.1393.1 Available under license

1.1394 har-schema 2.0.0 2.0.0
  1.1394.1 Available under license

1.1395 drange 1.1.1 1.1.1
  1.1395.1 Available under license

1.1396 iconv-lite 0.4.24 0.4.24
  1.1396.1 Available under license

1.1397 rxjs 6.6.3
  1.1397.1 Available under license

1.1398 jackson-datatype-joda 2.12.1
  1.1398.1 Available under license

1.1399 asm-tree 7.3.1
  1.1399.1 Available under license

1.1400 joda-time 2.10.5
  1.1400.1 Available under license

1.1401 babel-messages 6.23.0
  1.1401.1 Available under license

1.1402 converter-jackson 2.8.1
  1.1402.1 Available under license

1.1403 xmlhttprequest-ssl 1.5.3
  1.1403.1 Available under license

1.1404 is-resolvable 1.1.0
  1.1404.1 Available under license

1.1405 object.omit 2.0.1
1.1405.1 Available under license
1.1406 **type-of-is** 3.5.1 3.5.1
    1.1406.1 Available under license
1.1407 **balanced-match** 1.0.0
    1.1407.1 Available under license
1.1408 **wide-align** 1.1.3
    1.1408.1 Available under license
1.1409 **commons-io** 1.3.2
    1.1409.1 Available under license
1.1410 **asm** 5.2
1.1411 **hive-llap-client** 3.1.2
    1.1411.1 Available under license
1.1412 **commons-collections** 3.2.2
    1.1412.1 Available under license
1.1413 **commander** 2.20.3
    1.1413.1 Available under license
1.1414 **crc32-stream** 2.0.0
    1.1414.1 Available under license
1.1415 **javax-inject** 2.2.0-b21
    1.1415.1 Available under license
1.1416 **batik-gui-util** 1.13
    1.1416.1 Available under license
1.1417 **jsbn** 0.1.1 0.1.1
    1.1417.1 Available under license
1.1418 **qdox** 1.12.1
1.1419 **object.values** 1.1.1
    1.1419.1 Available under license
1.1420 **batik-xml** 1.13
    1.1420.1 Available under license
1.1421 **regenerate** 1.4.1
    1.1421.1 Available under license
1.1422 **jython** 2.7.1
    1.1422.1 Available under license
1.1423 **snappy** 1.0.5
    1.1423.1 Available under license
1.1424 **jsch** 0.1.55
1.1425 **babel-plugin-transform-async-generator-functions** 6.24.1
    1.1425.1 Available under license
1.1426 **camelcase** 5.3.1 5.3.1
<table>
<thead>
<tr>
<th>Version</th>
<th>Module Name</th>
<th>Version</th>
<th>Available under license</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1426.1</td>
<td>Antelope</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1427</td>
<td>resource locator</td>
<td>1.0.1</td>
<td></td>
</tr>
<tr>
<td>1.1427.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1428</td>
<td>metrics-health-checks</td>
<td>4.0.2</td>
<td></td>
</tr>
<tr>
<td>1.1429</td>
<td>fs-extra</td>
<td>1.0.0</td>
<td></td>
</tr>
<tr>
<td>1.1429.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1430</td>
<td>jquery</td>
<td>3.5.1</td>
<td></td>
</tr>
<tr>
<td>1.1430.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1431</td>
<td>jansi-osx</td>
<td>1.8</td>
<td></td>
</tr>
<tr>
<td>1.1431.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1432</td>
<td>html-escaper</td>
<td>2.0.2</td>
<td></td>
</tr>
<tr>
<td>1.1432.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1433</td>
<td>parse-color</td>
<td>1.0.0</td>
<td></td>
</tr>
<tr>
<td>1.1433.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1434</td>
<td>es5-ext</td>
<td>0.10.53</td>
<td></td>
</tr>
<tr>
<td>1.1434.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1435</td>
<td>bcrypt-pbkdf</td>
<td>1.0.2</td>
<td></td>
</tr>
<tr>
<td>1.1435.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1436</td>
<td>ratelimitj-core</td>
<td>0.4.0</td>
<td></td>
</tr>
<tr>
<td>1.1436.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1437</td>
<td>json-buffer</td>
<td>3.0.0</td>
<td></td>
</tr>
<tr>
<td>1.1437.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1438</td>
<td>jackson-jaxrs-xml-provider</td>
<td>2.10.4</td>
<td></td>
</tr>
<tr>
<td>1.1438.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1439</td>
<td>component-bind</td>
<td>1.0.0</td>
<td></td>
</tr>
<tr>
<td>1.1439.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1440</td>
<td>helmet</td>
<td>2.3.0</td>
<td></td>
</tr>
<tr>
<td>1.1440.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1441</td>
<td>babel-plugin-transform-es2015-destructuring</td>
<td>6.23.0 6.23.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1442</td>
<td>color-convert</td>
<td>2.0.1</td>
<td></td>
</tr>
<tr>
<td>1.1442.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1443</td>
<td>jersey-servlet</td>
<td>1.19</td>
<td></td>
</tr>
<tr>
<td>1.1444</td>
<td>nanoid</td>
<td>1.0.2</td>
<td></td>
</tr>
<tr>
<td>1.1444.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1445</td>
<td>invariant</td>
<td>2.2.4</td>
<td></td>
</tr>
<tr>
<td>1.1445.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1446</td>
<td>release-zalgo</td>
<td>1.0.0</td>
<td></td>
</tr>
<tr>
<td>1.1446.1</td>
<td></td>
<td></td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.1447 http2-client 1.3.3 1.3.3
   1.1447.1 Available under license

1.1448 babylon 6.18.0
   1.1448.1 Available under license

1.1449 object-component 0.0.3 0.0.3
   1.1449.1 Available under license

1.1450 stax-ex 1.8

1.1451 lucene-sandbox 7.4.0
   1.1451.1 Available under license

1.1452 batik-gvt 1.13
   1.1452.1 Available under license

1.1453 drill-format-hdf5 1.19.0
   1.1453.1 Available under license

1.1454 jsonpath-plus 3.0.0 3.0.0
   1.1454.1 Available under license

1.1455 babel-plugin-transform-es2015-block-scoped-functions 6.22.0
   1.1455.1 Available under license

1.1456 getpass 0.1.7 0.1.7
   1.1456.1 Available under license

1.1457 websocket-server 9.3.27.v20190418

1.1458 is-data-descriptor 0.1.4
   1.1458.1 Available under license

1.1459 which-module 2.0.0
   1.1459.1 Available under license

1.1460 jackson 1.9.13
   1.1460.1 Available under license

1.1461 options 0.0.6
   1.1461.1 Available under license

1.1462 async-each 1.0.3
   1.1462.1 Available under license

1.1463 okhttp 3.9.1
   1.1463.1 Available under license

1.1464 run-async 0.1.0
   1.1464.1 Available under license

1.1465 has 1.0.3 1.0.3
   1.1465.1 Available under license

1.1466 hive 3.1.2
   1.1466.1 Available under license

1.1467 lodash-padstart 4.6.1
1.1467.1 Available under license

1.1468 *mongodb* **2.2.36**
1.1468.1 Available under license

1.1469 *jsreport-handlebars* **2.1.0**
1.1469.1 Available under license

1.1470 *regjsparser* **0.1.5**
1.1470.1 Available under license

1.1471 *metrics-servlets* **4.0.2**

1.1472 *lowercase-keys* **1.0.1 1.0.1**
1.1472.1 Available under license

1.1473 *cycle* **1.0.3**
1.1473.1 Available under license

1.1474 *opentype-js* **1.1.0**
1.1474.1 Available under license

1.1475.1 Available under license

1.1476 *snmp-4j* **2.6.3**
1.1476.1 Available under license

1.1477 *model-mapper* **1.1.0**

1.1478 *which* **2.0.2**
1.1478.1 Available under license

1.1479 *os-locale* **3.1.0**
1.1479.1 Available under license

1.1480 *mitt* **1.1.3**
1.1480.1 Available under license

1.1481 *@babel/generator* **7.13.9**
1.1481.1 Available under license

1.1482 *@babel/helper-function-name* **7.12.13**
1.1482.1 Available under license

1.1483 *object-copy* **0.1.0**
1.1483.1 Available under license

1.1484 *bitsyntax* **0.1.0**
1.1484.1 Available under license

1.1485 *jackson-annotations* **2.11.4**
1.1485.1 Available under license

1.1486 *chalk* **1.1.3**
1.1486.1 Available under license

1.1487 *transaction-api* **1.1**
1.1487.1 Available under license
1.1488 jakarta-validation-api 2.0.2
   1.1488.1 Available under license
1.1489 ansi-regex 2.1.1
   1.1489.1 Available under license
1.1490 ncp 2.0.0 2.0.0
   1.1490.1 Available under license
1.1491 jsreport-scheduling 2.5.0
   1.1491.1 Available under license
1.1492 jarchivelib 1.0.0
   1.1492.1 Available under license
1.1493 jaxb-runtime 2.3.1
1.1494 jetty-http-spi 9.3.27.v20190418
1.1495 chalk 2.4.2
   1.1495.1 Available under license
1.1496 compression 1.6.2
   1.1496.1 Available under license
1.1497 ignite-shmem 1.0.0
   1.1497.1 Available under license
1.1498 registry-auth-token 4.2.1
   1.1498.1 Available under license
1.1499 anymatch 1.3.2
   1.1499.1 Available under license
1.1500 js-tokens 3.0.2 3.0.2
   1.1500.1 Available under license
1.1501 lodash 4.17.15 4.17.15
   1.1501.1 Available under license
1.1502 is-negative-zero 2.0.1
   1.1502.1 Available under license
1.1503 mime 1.4.1 1.4.1
   1.1503.1 Available under license
1.1504 axios 0.18.1 0.18.1
   1.1504.1 Available under license
1.1505 unzipper 0.9.15
   1.1505.1 Available under license
1.1506 ejs 2.6.2
   1.1506.1 Available under license
1.1507 babel-helper-remap-async-to-generator 6.24.1 6.24.1
   1.1507.1 Available under license
1.1508 negotiator 0.6.1 0.6.1
<table>
<thead>
<tr>
<th>Version</th>
<th>Package Name</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1508.1</td>
<td>Available under license</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1509</td>
<td>coffee-script 2.5.1</td>
<td>2.5.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1510</td>
<td>pause-stream 0.0.11</td>
<td>0.0.11</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1511</td>
<td>commons-compress 1.14</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1512</td>
<td>js-cookie 2.1.1</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1513</td>
<td>tomcat 8.5.46</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1514</td>
<td>antlr-runtime 4.8-1</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1515</td>
<td>cookie-parser 1.4.4</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1516</td>
<td>through2 2.0.5</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1517</td>
<td>babel-plugin-syntax-async-functions 6.13.0</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1518</td>
<td>babel-plugin-transform-es2015-classes 6.24.1</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1519</td>
<td>babel-helper-define-map 6.26.0</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1520</td>
<td>wrap-ansi 6.2.0</td>
<td>6.2.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1521</td>
<td>ansi-escapes 3.2.0</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1522</td>
<td>activation 1.1</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1523</td>
<td>undefsafe 2.0.3</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1524</td>
<td>regexpu-core 4.7.1</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1525</td>
<td>jsonwebtoken 7.4.3</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1526</td>
<td>jersey-hk2 2.32</td>
<td></td>
<td>Available under license</td>
</tr>
<tr>
<td>1.1527</td>
<td>thrift 0.9.3</td>
<td></td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.1528 mongodb-core 2.1.20 2.1.20
   1.1528.1 Available under license
1.1529 jackson 2.12.1
   1.1529.1 Available under license
1.1530 fs-readfile-promise 2.0.1 2.0.1
   1.1530.1 Available under license
1.1531 glob-parent 3.1.0
   1.1531.1 Available under license
1.1532 @babel/plugin-proposal-class-properties 7.13.0
   1.1532.1 Available under license
1.1533 default-require-extensions 3.0.0
   1.1533.1 Available under license
1.1534 @babel/plugin-proposal-optional-catch-binding 7.13.8
   1.1534.1 Available under license
1.1535 junit-jupiter-api 5.7.0
   1.1535.1 Available under license
1.1536 babel-helper-call-delegate 6.24.1
   1.1536.1 Available under license
1.1537 should.js 13.2.3 13.2.3
   1.1537.1 Available under license
1.1538 readdirp 3.2.0
   1.1538.1 Available under license
1.1539 lodash-some 4.6.0
   1.1539.1 Available under license
1.1540 password-hash 1.2.2
   1.1540.1 Available under license
1.1541 puppeteer 1.11.0
   1.1541.1 Available under license
1.1542 hpkp 1.2.0 1.2.0
   1.1542.1 Available under license
1.1543 crypto-random-string 2.0.0
   1.1543.1 Available under license
1.1544 jetty-xml 9.4.17.v20190418
   1.1544.1 Available under license
1.1545 execa 1.0.0
   1.1545.1 Available under license
1.1546 once 1.4.0
   1.1546.1 Available under license
1.1547 babel-plugin-transform-es2015-parameters 6.24.1 6.24.1
1.1547.1 Available under license

1.1548 **websocket-servlet 9.4.17.v20190418**
   1.1548.1 Available under license

1.1549 **servlet-api 3.1.0**
   1.1549.1 Available under license

1.1550 **flagged-respawn 1.0.1**
   1.1550.1 Available under license

1.1551 **summernote 0.8.15 0.8.15**
   1.1551.1 Available under license

1.1552 **is-extglob 1.0.0**
   1.1552.1 Available under license

1.1553 **setimmediate 1.0.5 1.0.5**
   1.1553.1 Available under license

1.1554 **decamelize 1.2.0 1.2.0**
   1.1554.1 Available under license

1.1555 **core-js 2.5.7 2.5.7**
   1.1555.1 Available under license

1.1556 **commons-io 2.5**
   1.1556.1 Available under license

1.1557 **batik-bridge 1.13**
   1.1557.1 Available under license

1.1558 **immer 8.0.1 8.0.1**
   1.1558.1 Available under license

1.1559 **parquet-hadoop 1.12.0**
   1.1559.1 Available under license

1.1560 **fromentries 1.3.2**
   1.1560.1 Available under license

1.1561 **guava 30.0-jre**
   1.1561.1 Available under license

1.1562 **sprintf-js 1.0.3**
   1.1562.1 Available under license

1.1563 **flat 4.1.1**
   1.1563.1 Available under license

1.1564 **ret 0.2.2 0.2.2**
   1.1564.1 Available under license

1.1565 **performance-now 2.1.0 2.1.0**
   1.1565.1 Available under license

1.1566 **memoize-one 4.1.0 4.1.0**
   1.1566.1 Available under license
1.1567 librsvg 2.40.13-3
   1.1567.1 Available under license
1.1568 lowercase-keys 2.0.0
   1.1568.1 Available under license
1.1569 babel-polyfill 6.26.0
1.1570 accepts 1.3.7
   1.1570.1 Available under license
1.1571 options 0.0.6 0.0.6
   1.1571.1 Available under license
1.1572 component-emitter 1.1.2
   1.1572.1 Available under license
1.1573 cliui 6.0.0 6.0.0
   1.1573.1 Available under license
1.1574 babel-helper-call-delegate 6.24.1 6.24.1
   1.1574.1 Available under license
1.1575 unbox-primitive 1.0.1
   1.1575.1 Available under license
1.1576 latencyutils 2.0.3
   1.1576.1 Available under license
1.1577 hasha 5.2.2
   1.1577.1 Available under license
1.1578 de.huxhorn.lilith.data.logging.protobuf 0.9.44
   1.1578.1 Available under license
1.1579 object-inspect 1.9.0
   1.1579.1 Available under license
1.1580 ws 6.1.4
   1.1580.1 Available under license
1.1581 v8flags 3.2.0
   1.1581.1 Available under license
1.1582 ansi-regex 5.0.0 5.0.0
   1.1582.1 Available under license
1.1583 babel-plugin-syntax-trailing-function-commas 6.22.0 6.22.0
   1.1583.1 Available under license
1.1584 groovy-swing 3.0.8
   1.1584.1 Available under license
1.1585 is-my-ip-valid 1.0.0 1.0.0
   1.1585.1 Available under license
1.1586 safe-buffer 5.2.1 5.2.1
   1.1586.1 Available under license
1.1587 hsts 1.0.0 1.0.0
   1.1587.1 Available under license
1.1588 stealthy-require 1.1.1
   1.1588.1 Available under license
1.1589 uglify-js 3.11.3
   1.1589.1 Available under license
1.1590 parquet-common 1.8.1
   1.1590.1 Available under license
1.1591 1to2 1.0.0
   1.1591.1 Available under license
1.1592 crc32-stream 3.0.1
   1.1592.1 Available under license
1.1593 har-validator 2.1.3 2.1.3
   1.1593.1 Available under license
1.1594 @babel/parser 7.11.5
   1.1594.1 Available under license
   1.1595.1 Available under license
1.1596 swagger-jaxrs 1.5.0
1.1597 lodash-find 4.6.0
   1.1597.1 Available under license
1.1598 through 2.3.8 2.3.8
   1.1598.1 Available under license
1.1599 async 1.5.2
   1.1599.1 Available under license
1.1600 classmate 0.8.0
   1.1600.1 Available under license
1.1601 cookie 0.1.5 0.1.5
   1.1601.1 Available under license
1.1602 callsites 0.2.0
   1.1602.1 Available under license
1.1603 atk 2.18.0 1
   1.1603.1 Available under license
1.1604 grpc-api 1.27.1
   1.1604.1 Available under license
1.1605 asm 2.2.0-b21
   1.1605.1 Available under license
1.1606 minlog 1.3.0
   1.1606.1 Available under license
1.1607 uuid 3.3.2
  1.1607.1 Available under license
1.1608 readable-stream 2.3.7 2.3.7
  1.1608.1 Available under license
1.1609 avro 1.9.1
  1.1609.1 Available under license
1.1610 set-value 2.0.1
  1.1610.1 Available under license
1.1611 clone-response 1.0.2
  1.1611.1 Available under license
1.1612 libvpx 1.7.0-1367-gac3eccdc2
  1.1612.1 Available under license
1.1613 path-to-regexp 1.8.0 1.8.0
  1.1613.1 Available under license
1.1614 concat-stream 1.6.2
  1.1614.1 Available under license
1.1615 regjsgen 0.2.0 0.2.0
  1.1615.1 Available under license
1.1616 is-my-json-valid 2.17.2 2.17.2
  1.1616.1 Available under license
1.1617 css-select 1.2.0
  1.1617.1 Available under license
1.1618 hbase 2.4.2
  1.1618.1 Available under license
1.1619 @babel/helper-validator-identifier 7.10.4
  1.1619.1 Available under license
1.1620 jetty-util-ajax 9.4.34.v20201102
  1.1620.1 Available under license
1.1621 metrics-jvm 4.0.2
1.1622 multer 1.4.1
  1.1622.1 Available under license
1.1623 ansi-colors 3.2.3
  1.1623.1 Available under license
1.1624 after 0.8.2 0.8.2
  1.1624.1 Available under license
1.1625 jackson-mapper-asl 1.9.13
  1.1625.1 Available under license
1.1626 aws-sign2 0.7.0
  1.1626.1 Available under license
1.1627 arraybuffer.slice 0.0.6
   1.1627.1 Available under license
1.1628 connect 3.4.1
   1.1628.1 Available under license
1.1629 jackson-databind 2.12.1
   1.1629.1 Available under license
1.1630 oauth-sign 0.8.2 0.8.2
   1.1630.1 Available under license
1.1631 node-ensure 0.0.0
   1.1631.1 Available under license
1.1632 source-map-resolve 0.5.3
   1.1632.1 Available under license
1.1633 isarray 2.0.1
   1.1633.1 Available under license
1.1634 @babel/plugin-transform-exponentiation-operator 7.12.13
   1.1634.1 Available under license
1.1635 brace-expansion 1.1.11
   1.1635.1 Available under license
1.1636 anymatch 3.1.1
   1.1636.1 Available under license
1.1637 1to2 1.0.0 1.0.0
   1.1637.1 Available under license
1.1638 wrap-ansi 5.1.0
   1.1638.1 Available under license
1.1639 graceful-fs 4.2.4
   1.1639.1 Available under license
1.1640 is-npm 4.0.0
   1.1640.1 Available under license
1.1641 cron-parser 2.7.3
   1.1641.1 Available under license
1.1642 array-slice 1.1.0
   1.1642.1 Available under license
1.1643 deep-is 0.1.3
   1.1643.1 Available under license
1.1644 string-prototype-trimstart 1.0.1
   1.1644.1 Available under license
1.1645 randomatic 3.1.1
   1.1645.1 Available under license
1.1646 locate-path 5.0.0 5.0.0
1.1646.1 Available under license

1.1647 babel-helper-get-function-arity 6.24.1
   1.1647.1 Available under license

1.1648 path-is-absolute 1.0.1 1.0.1
   1.1648.1 Available under license

1.1649 protobuf-java 2.5.31
   1.1649.1 Available under license

1.1650 jsreport-cli 2.2.5
   1.1650.1 Available under license

1.1651 is-extglob 2.1.1
   1.1651.1 Available under license

1.1652 joni 2.1.31
   1.1652.1 Available under license

1.1653 events 2.1.0
   1.1653.1 Available under license

1.1654 fontconfig 2.11.94 0ubuntu1.1
   1.1654.1 Available under license

1.1655 babel-plugin-transform-reserved-words 7.12.13
   1.1655.1 Available under license

1.1656 utils-merge 1.0.1 1.0.1
   1.1656.1 Available under license

1.1657 is-finite 1.1.0
   1.1657.1 Available under license

1.1658 scylla-driver-core 3.10.2-scylla-1
   1.1658.1 Available under license

1.1659 abbrev 1.1.1
   1.1659.1 Available under license

1.1660 lowercase-keys 1.0.1
   1.1660.1 Available under license

1.1661 istanbul-lib-coverage 3.0.0
   1.1661.1 Available under license

1.1662 jackson 1.0.0
   1.1662.1 Available under license

1.1663 oas-resolver 2.5.4 2.5.4
   1.1663.1 Available under license

1.1664 mv 2.1.1 2.1.1
   1.1664.1 Available under license

1.1665 babel-plugin-syntax-async-generators 6.13.0
   1.1665.1 Available under license
<table>
<thead>
<tr>
<th>Version</th>
<th>Package Name</th>
<th>Version</th>
<th>Available under license</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1666</td>
<td>shelljs</td>
<td>0.3.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.3.0</td>
<td></td>
</tr>
<tr>
<td>1.1667</td>
<td>domhandler</td>
<td>2.4.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1668</td>
<td>lodash.isundefined</td>
<td>3.0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1669</td>
<td>is-data-descriptor</td>
<td>1.0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1670</td>
<td>safe-regex</td>
<td>1.1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1671</td>
<td>syslog-java-client</td>
<td>1.1.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1672</td>
<td>netty</td>
<td>4.1.46.Final</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1673</td>
<td>readable-stream</td>
<td>3.6.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1674</td>
<td>jsreport-office</td>
<td>1.2.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1675</td>
<td>color-name</td>
<td>1.1.4</td>
<td>1.1.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1676</td>
<td>netty-transport-native-epoll</td>
<td>4.1.42.Final</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1677</td>
<td>qs</td>
<td>6.5.2</td>
<td>6.5.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1678</td>
<td>parser-core</td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1679</td>
<td>swagger-schema-official</td>
<td>2.0.0-bab6bed</td>
<td>2.0.0-bab6bed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1680</td>
<td>parquet-jackson</td>
<td>1.12.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1681</td>
<td>istack-commons</td>
<td>3.0.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1682</td>
<td>loose-envify</td>
<td>1.4.0</td>
<td>1.4.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1683</td>
<td>httpclient5</td>
<td>5.0.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1684</td>
<td>babel-plugin-polyfill-corejs3</td>
<td>0.1.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1685</td>
<td>socket.io-adapter</td>
<td>1.1.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>find-up</td>
<td>3.0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>micromatch</td>
<td>3.1.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mimic-fn</td>
<td>1.2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mime-types</td>
<td>2.1.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>socket.io-parser</td>
<td>2.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tomcat-annotations-api</td>
<td>8.5.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jsreport-resources</td>
<td>2.1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>httpasyncclient</td>
<td>4.1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>batik-css</td>
<td>1.13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lodash-isfunction</td>
<td>3.0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jersey-entity-filtering</td>
<td>2.25.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>whatwg-fetch</td>
<td>3.4.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tephra-api</td>
<td>0.6.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jcommander</td>
<td>1.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lodash.flattendeep</td>
<td>4.4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c3p0</td>
<td>0.9.5.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>commons-codec</td>
<td>1.14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>promise</td>
<td>2.1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ansi-regex</td>
<td>3.0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.1705.1 Available under license
1.1706 @babel/runtime 7.0.0-beta.44 7.0.0-beta.44
   1.1706.1 Available under license
1.1707 yup 0.29.3 0.29.3
   1.1707.1 Available under license
1.1708 p-locate 4.1.0
   1.1708.1 Available under license
1.1709 quartz 2.3.2
   1.1709.1 Available under license
1.1710 request 2.88.2 2.88.2
   1.1710.1 Available under license
1.1711 batik-dom 1.13
   1.1711.1 Available under license
1.1712 pathval 1.1.1
   1.1712.1 Available under license
1.1713 chalk 0.4.0 0.4.0
   1.1713.1 Available under license
1.1714 buffer-from 1.1.1
   1.1714.1 Available under license
1.1715 glob-parent 2.0.0
   1.1715.1 Available under license
1.1716 jersey-common 2.5.1
   1.1716.1 Available under license
1.1717 colord 1.2.12-1ubuntu1
   1.1717.1 Available under license
1.1718 jna 3.5.1
   1.1718.1 Available under license
1.1719 bl 2.2.1 2.2.1
   1.1719.1 Available under license
1.1720 kind-of 6.0.3
   1.1720.1 Available under license
1.1721 babel-register 6.26.0 6.26.0
   1.1721.1 Available under license
1.1722 proxy-addr 1.0.10
   1.1722.1 Available under license
1.1723 estraverse 1.9.3
   1.1723.1 Available under license
1.1724 drill-kudu-storage 1.19.0
   1.1724.1 Available under license
1.1725 pupa 2.1.1
  1.1725.1 Available under license
1.1726 ini 1.3.7
  1.1726.1 Available under license
1.1727 should-equal 2.0.0 2.0.0
  1.1727.1 Available under license
1.1728 react-intl 2.4.0 2.4.0
  1.1728.1 Available under license
1.1729 jackson-xc 2.9.10
  1.1729.1 Available under license
1.1730 @babel/helper-define-polyfill-provider 0.1.5
  1.1730.1 Available under license
1.1731 drill-rdbms-metastore 1.19.0
  1.1731.1 Available under license
1.1732 hbaselogging 2.4.2
  1.1732.1 Available under license
1.1733 node-forge-flash 0.0.0
  1.1733.1 Available under license
1.1734 fast-safe-stringify 2.0.7 2.0.7
  1.1734.1 Available under license
1.1735 jline-style 3.12.1
  1.1735.1 Available under license
1.1736 string-prototype-trimend 1.0.1
  1.1736.1 Available under license
1.1737 gtk 3.18.9-1ubuntu3.3
  1.1737.1 Available under license
1.1738 @babel/plugin-transform-computed-properties 7.13.0
  1.1738.1 Available under license
1.1739 source-map-url 0.4.0
  1.1739.1 Available under license
1.1740 qs 4.0.0 4.0.0
  1.1740.1 Available under license
1.1741 immutability-helper 2.7.1 2.7.1
  1.1741.1 Available under license
1.1742 tmp 0.0.33
  1.1742.1 Available under license
1.1743 guava 19.0
1.1744 extsprintf 1.3.0 1.3.0
  1.1744.1 Available under license
1.1745 is-stream 2.0.0
   1.1745.1 Available under license
1.1746 node-extend 2.0.2
   1.1746.1 Available under license
1.1747 readline2 1.0.1
   1.1747.1 Available under license
1.1748 sparse-bitfield 3.0.3 3.0.3
   1.1748.1 Available under license
1.1749 xalan 2.7.2
   1.1749.1 Available under license
1.1750 babel-plugin-transform-es2015-for-of 6.23.0
   1.1750.1 Available under license
1.1751 calcite-cassandra 1.21.0-drill-r2
   1.1751.1 Available under license
1.1752 content-disposition 0.5.1 0.5.1
   1.1752.1 Available under license
1.1753 emoji-regex 8.0.0 8.0.0
   1.1753.1 Available under license
1.1754 avro 1.7.6
   1.1754.1 Available under license
1.1755 object-assign 3.0.0 3.0.0
   1.1755.1 Available under license
1.1756 cypress 1.0.0 1.0.0
   1.1756.1 Available under license
1.1757 jersey-container-servlet 2.5.1
   1.1757.1 Available under license
1.1758 glob 6.0.4
   1.1758.1 Available under license
1.1759 proxy-addr 2.0.4 2.0.4
   1.1759.1 Available under license
1.1760 thrift 0.7.0
   1.1760.1 Available under license
1.1761 is-buffer 1.1.6
   1.1761.1 Available under license
1.1762 resolve 1.18.1
   1.1762.1 Available under license
1.1763 sprintf-js 1.1.0
   1.1763.1 Available under license
1.1764 jna 5.8.0
1.1764.1 Available under license
1.1765 pcapngdecoder 1.2
  1.1765.1 Available under license
1.1766 expand-range 1.8.2
  1.1766.1 Available under license
1.1767 babel-plugin-syntax-class-properties 6.13.0
  1.1767.1 Available under license
1.1768 okio 1.13.0
1.1769 apr 1.6.5
1.1770 istanbul-lib-report 3.0.0
  1.1770.1 Available under license
1.1771 mail 1.4.7
  1.1771.1 Available under license
1.1772 pirates 4.0.1
  1.1772.1 Available under license
1.1773 libxi 1.7.6-1
  1.1773.1 Available under license
1.1774 mute-stream 0.0.7
  1.1774.1 Available under license
1.1775 commons-lang3 3.8.1
  1.1775.1 Available under license
1.1776 calcite-example-function 1.21.0
  1.1776.1 Available under license
1.1777 json-simple 1.1.1
1.1778 amqp-client 3.6.6
  1.1778.1 Available under license
1.1779 fast-json-stable-stringify 2.1.0
  1.1779.1 Available under license
1.1780 babel-plugin-transform-es2015-block-scoped-functions 6.22.0 6.22.0
  1.1780.1 Available under license
1.1781 gson 2.8.6
  1.1781.1 Available under license
1.1782 passport 0.4.0
  1.1782.1 Available under license
1.1783 debug 4.3.1
  1.1783.1 Available under license
1.1784 activation-api 1.2.1
  1.1784.1 Available under license
1.1785 t-digest 3.2
1.1786 levn 0.3.0
  1.1786.1 Available under license
1.1787 es6-promise 3.2.1
  1.1787.1 Available under license
1.1788 jline-terminal 3.12.1
  1.1788.1 Available under license
1.1789 jetty-util 9.4.17.v20190418
  1.1789.1 Available under license
1.1790 log4j-api 2.8.2
  1.1790.1 Available under license
1.1791 de.huxhorn.sulky.io 0.9.17
  1.1791.1 Available under license
1.1792 ee-first 1.1.1
  1.1792.1 Available under license
  1.1793.1 Available under license
1.1794 eslint-plugin-import 2.22.1
  1.1794.1 Available under license
1.1795 lodash-merge 4.6.2
  1.1795.1 Available under license
1.1796 libx11 1.6.3-1ubuntu2.1
  1.1796.1 Available under license
1.1797 form-data 2.5.0
  1.1797.1 Available under license
1.1798 axios 0.19.2
  1.1798.1 Available under license
1.1799 metrics-servlets 4.1.0
1.1800 node-fetch-h 2.3.0 2.3.0
  1.1800.1 Available under license
1.1801 formidable 1.2.1
  1.1801.1 Available under license
1.1802 commons-dbcp 2.6.0
  1.1802.1 Available under license
1.1803 jsreport-tags 2.5.0
  1.1803.1 Available under license
1.1804 libxcb 1.11.1 1ubuntu1
  1.1804.1 Available under license
1.1805 https-proxy-agent 2.2.4
  1.1805.1 Available under license
1.1806 commons-io 2.6
   1.1806.1 Available under license
1.1807 ansicolors 0.2.1 0.2.1
   1.1807.1 Available under license
1.1808 aws4 1.8.0 1.8.0
   1.1808.1 Available under license
1.1809 buffer-indexof-polyfill 1.0.2
   1.1809.1 Available under license
1.1810 commons-dbcp 1.4
   1.1810.1 Available under license
1.1811 follow-redirects 1.14.5 1.14.5
   1.1811.1 Available under license
1.1812 drill-protocol 1.19.0
   1.1812.1 Available under license
1.1813 apib-include-directive 0.1.0 0.1.0
   1.1813.1 Available under license
1.1814 readdirp 2.2.1
   1.1814.1 Available under license
1.1815 cookies 0.7.3
   1.1815.1 Available under license
1.1816 httpdlog-parser 5.7
   1.1816.1 Available under license
1.1817 escape-html 1.0.3
   1.1817.1 Available under license
1.1818 icu 63
   1.1818.1 Available under license
1.1819 ip-address 5.8.9
1.1820 react-is 16.12.0 16.12.0
   1.1820.1 Available under license
1.1821 tmp 0.1.0
   1.1821.1 Available under license
1.1822 @babel/helper-annotate-as-pure 7.12.13
   1.1822.1 Available under license
1.1823 escalade 3.1.1
   1.1823.1 Available under license
1.1824 end-of-stream 1.4.4
   1.1824.1 Available under license
1.1825 jackson-module-afterburner 2.12.1
   1.1825.1 Available under license
1.1845.1 Available under license

1.1846 commons-el 5.5.23
1.1846.1 Available under license

1.1847 batik 1.13
1.1847.1 Available under license

1.1848 is-symbol 1.0.3
1.1848.1 Available under license

1.1849 ci-info 2.0.0
1.1849.1 Available under license

1.1850 y18n 5.0.8 5.0.8
1.1850.1 Available under license

1.1851 uri-js 4.4.0
1.1851.1 Available under license

1.1852 mv 2.1.1
1.1852.1 Available under license

1.1853 unset-value 1.0.0
1.1853.1 Available under license

1.1854 kite-data-core 1.1.0
1.1854.1 Available under license

1.1855 is-my-ip-valid 1.0.0
1.1855.1 Available under license

1.1856 excel-streaming-reader 3.0.4
1.1856.1 Available under license

1.1857 drill-storage-hive-core 1.19.0
1.1857.1 Available under license

1.1858 ajv 5.5.2 5.5.2
1.1858.1 Available under license

1.1859 babel-traverse 6.26.0
1.1859.1 Available under license

1.1860 deep-sort-object 1.0.2 1.0.2
1.1860.1 Available under license

1.1861 arr-diff 2.0.0
1.1861.1 Available under license

1.1862 jersey 2.25.1
1.1862.1 Available under license

1.1863 groovy-sql 3.0.8
1.1863.1 Available under license

1.1864 minimatch 3.0.4 3.0.4
1.1864.1 Available under license
Open Source Used In DNAC Platform Antelope

1.1865 css-what 2.1.3  
1.1865.1 Available under license

1.1866 httpsasyncclient 4.1.4  
1.1866.1 Available under license

1.1867 cliui 3.2.0  
1.1867.1 Available under license

1.1868 moment 2.27.0  
1.1868.1 Available under license

1.1869 firstline 1.3.1 1.3.1  
1.1869.1 Available under license

1.1870 react-copy-to-clipboard 5.0.2  
1.1870.1 Available under license

1.1871 array prototype flat 1.2.3  
1.1871.1 Available under license

1.1872 concat-map 0.0.1 0.0.1  
1.1872.1 Available under license

1.1873 batik-shared-resources 1.13  
1.1873.1 Available under license

1.1874 is-buffer 2.0.5  
1.1874.1 Available under license

1.1875 pify 2.3.0  
1.1875.1 Available under license

1.1876 commons-codec 1.12  
1.1876.1 Available under license

1.1877 p-map 3.0.0  
1.1877.1 Available under license

1.1878 hbase-client 2.4.2  
1.1878.1 Available under license

1.1879 wrappy 1.0.2  
1.1879.1 Available under license

1.1880 netty-buffer 4.0.48.Final  
1.1880.1 Available under license

1.1881 fs-constants 1.0.0  
1.1881.1 Available under license

1.1882 debug 3.1.0  
1.1882.1 Available under license

1.1883 jakarta-inject 2.6.1  
1.1883.1 Available under license

1.1884 restore-cursor 1.0.1
1.1884.1 Available under license

1.1885 **jackson 2.2.3**
1.1885.1 Available under license

1.1886 **prelude-ls 1.1.2**
1.1886.1 Available under license

1.1887 **nopt 1.0.10**
1.1887.1 Available under license

1.1888 **statuses 1.2.1 1.2.1**
1.1888.1 Available under license

1.1889 **hk2-utils 2.5.0-b32**
1.1889.1 Available under license

1.1890 **slash 2.0.0**
1.1890.1 Available under license

1.1891 **fast-deep-equal 2.0.1 2.0.1**
1.1891.1 Available under license

1.1892 **babel-plugin-transform-es2015-modules-amd 6.24.1**
1.1892.1 Available under license

1.1893 **find-cache-dir 3.3.1**
1.1893.1 Available under license

1.1894 **slash 1.0.0**
1.1894.1 Available under license

1.1895 **jasper-runtime 5.5.12**
1.1895.1 Available under license

1.1896 **reflections 0.9.10**
1.1896.1 Available under license

1.1897 **guice-servlet 4.0**
1.1897.1 Available under license

1.1898 **istanbul-lib-instrument 4.0.3**
1.1898.1 Available under license

1.1899 **tough-cookie 2.3.4 2.3.4**
1.1899.1 Available under license

1.1900 **map-stream 0.1.0 0.1.0**
1.1900.1 Available under license

1.1901 **checker-compat-qual 2.0.0**
1.1902 **utils-merge 1.0.0**
1.1902.1 Available under license

1.1903 **define-properties 1.1.3**
1.1903.1 Available under license

1.1904 **read 1.0.7**
1.1904.1 Available under license
1.1905 fs-readdir-recursive 1.1.0
   1.1905.1 Available under license
1.1906 should-type 1.4.0 1.4.0
   1.1906.1 Available under license
1.1907 typedarray-to-buffer 3.1.5
   1.1907.1 Available under license
1.1908 invert-kv 1.0.0
   1.1908.1 Available under license
1.1909 kerb-client 1.0.0
   1.1909.1 Available under license
1.1910 jboss-logging 3.1.1.GA
   1.1910.1 Available under license
1.1911 babel-plugin-transform-es2015-computed-properties 6.24.1
   1.1911.1 Available under license
1.1912 browserslist 4.16.3
   1.1912.1 Available under license
1.1913 drill-iceberg-metastore 1.19.0
   1.1913.1 Available under license
1.1914 kafka-schema-serializer 6.1.1
   1.1914.1 Available under license
1.1915 is-boolean-object 1.1.0
   1.1915.1 Available under license
1.1916 ms 2.1.2
   1.1916.1 Available under license
1.1917 hbase-http 2.4.2
   1.1917.1 Available under license
1.1918 animal-sniffer-annotation 1.18
1.1919 babel-plugin-transform-es2015-sticky-regex 6.24.1
   1.1919.1 Available under license
1.1920 yargs 16.2.0 16.2.0
   1.1920.1 Available under license
1.1921 foodmart-data-json 0.4
   1.1921.1 Available under license
1.1922 tiger-types 1.4
1.1923 jersey-guava 2.25.1
   1.1923.1 Available under license
1.1924 async-replace-with-limit 1.1.0
   1.1924.1 Available under license
1.1925 babel-plugin-transform-es2015-literals 6.22.0
   1.1925.1 Available under license

1.1926 @babel/plugin-transform-named-capturing-groups-regex 7.12.13
   1.1926.1 Available under license

1.1927 @babel/plugin-transform-unicode-regex 7.12.13
   1.1927.1 Available under license

1.1928 swagger-annotations 1.6.2
   1.1928.1 Available under license

1.1929 parseqs 0.0.6
   1.1929.1 Available under license

1.1930 onetime 2.0.1
   1.1930.1 Available under license

1.1931 nomnom 1.8.1 1.8.1
   1.1931.1 Available under license

1.1932 swagger2openapi 7.0.3 7.0.3
   1.1932.1 Available under license

1.1933 bean-validation-api 2.0.1.Final
   1.1933.1 Available under license

1.1934 babel-plugin-transform-async-to-generator 6.24.1 6.24.1
   1.1934.1 Available under license

1.1935 readable-stream 2.2.7 2.2.7
   1.1935.1 Available under license

1.1936 negotiator 0.6.1
   1.1936.1 Available under license

1.1937 reactapp 1.0.0 1.0.0
   1.1937.1 Available under license

1.1938 tweetnacl 0.14.5 0.14.5
   1.1938.1 Available under license

1.1939 nopt 3.0.6 3.0.6
   1.1939.1 Available under license

1.1940 hoek 2.16.3 2.16.3
   1.1940.1 Available under license

1.1941 emoji-regex 8.0.0
   1.1941.1 Available under license

1.1942 @babel/code-frame 7.5.5
   1.1942.1 Available under license

1.1943 @babel/runtime 7.0.0-beta.44
   1.1943.1 Available under license

1.1944 xml-apis-ext 1.3.04
1.1944.1 Available under license
1.1945 q 0.9.7 0.9.7
1.1945.1 Available under license
1.1946 swagger-annotations 1.5.0
1.1946.1 Available under license
1.1947 streams 2.7.0
1.1948 json-schema-validator 0.1.16
1.1948.1 Available under license
1.1949 jackson-dataformat-yaml 2.9.9
1.1949.1 Available under license
1.1950 sax 1.2.4
1.1950.1 Available under license
1.1951 type-fest 0.8.1
1.1951.1 Available under license
1.1952 @babel/types 7.11.5
1.1952.1 Available under license
1.1953 gensync 1.0.0-beta.2
1.1953.1 Available under license
1.1954 jsreport 2.11.0
1.1954.1 Available under license
1.1955 dom-serializer 0.1.0 0.1.0
1.1955.1 Available under license
1.1956 component-inherit 0.0.3 0.0.3
1.1956.1 Available under license
1.1957 lodash.isequal 4.5.0 4.5.0
1.1957.1 Available under license
1.1958 passport-http-bearer 1.0.1
1.1958.1 Available under license
1.1959 lodash-foreach 4.5.0
1.1959.1 Available under license
1.1960 callsite 1.0.0 1.0.0
1.1960.1 Available under license
1.1961 handlebars 4.7.6
1.1961.1 Available under license
1.1962 @babel/runtime 7.12.1 7.12.1
1.1962.1 Available under license
1.1963 react-pdf 3.0.6 3.0.6
1.1963.1 Available under license
1.1964 metrics-json 4.1.0
1.1965 dont-sniff-mimetype 1.0.0
   1.1965.1 Available under license
1.1966 @babel/plugin-transform-for-of 7.13.0
   1.1966.1 Available under license
1.1967 content-disposition 0.5.1
   1.1967.1 Available under license
1.1968 p-locate 4.1.0 4.1.0
   1.1968.1 Available under license
1.1969 object-defaults 1.1.0
   1.1969.1 Available under license
1.1970 debug 4.1.0
   1.1970.1 Available under license
1.1971 path-type 2.0.0
   1.1971.1 Available under license
1.1972 expand-tilde 2.0.2
   1.1972.1 Available under license
1.1973 content-security-policy-builder 1.0.0
   1.1973.1 Available under license
1.1974 detect-indent 4.0.0
   1.1974.1 Available under license
1.1975 split-string 3.1.0
   1.1975.1 Available under license
1.1976 negotiator 0.5.3 0.5.3
   1.1976.1 Available under license
1.1977 write-file-atomic 3.0.3
   1.1977.1 Available under license
1.1978 decompress-response 3.3.0
   1.1978.1 Available under license
1.1979 fresh 0.5.2 0.5.2
   1.1979.1 Available under license
1.1980 jetty 9.4.34.v20201102
   1.1980.1 Available under license
1.1981 woodstox-core 5.3.0
   1.1981.1 Available under license
1.1982 hibernate-validator 5.0.0.Final
   1.1982.1 Available under license
1.1983 babel-helpers 6.24.1 6.24.1
   1.1983.1 Available under license
1.1984 socket.io 2.3.0
<table>
<thead>
<tr>
<th>Year</th>
<th>Package Name</th>
<th>Version</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>1984</td>
<td>Antelope</td>
<td>1.02</td>
<td></td>
</tr>
<tr>
<td>1985</td>
<td>guava</td>
<td>30.1-jre</td>
<td></td>
</tr>
<tr>
<td>1986</td>
<td>kind-of</td>
<td>3.2.2</td>
<td></td>
</tr>
<tr>
<td>1987</td>
<td>swagger-core</td>
<td>2.0.7</td>
<td></td>
</tr>
<tr>
<td>1988</td>
<td>json5</td>
<td>1.0.1</td>
<td></td>
</tr>
<tr>
<td>1989</td>
<td>de.huxhorn.lilith.data.converter</td>
<td>0.9.44</td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td>glibc</td>
<td>1.1.0</td>
<td></td>
</tr>
<tr>
<td>1991</td>
<td>drill-druid-storage</td>
<td>1.19.0</td>
<td></td>
</tr>
<tr>
<td>1992</td>
<td>findbugs-annotations</td>
<td>1.3.9-1</td>
<td></td>
</tr>
<tr>
<td>1993</td>
<td>async</td>
<td>0.9.2</td>
<td></td>
</tr>
<tr>
<td>1994</td>
<td>jcommander</td>
<td>1.78</td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td>has-value</td>
<td>0.3.1</td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>pinkie</td>
<td>2.0.4</td>
<td></td>
</tr>
<tr>
<td>1997</td>
<td>jersey-client</td>
<td>2.5.1</td>
<td></td>
</tr>
<tr>
<td>1998</td>
<td>commons-compress</td>
<td>1.2.2</td>
<td></td>
</tr>
<tr>
<td>1999</td>
<td>babel-eslint</td>
<td>6.1.2</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>emoji-regex</td>
<td>7.0.3</td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>method-override</td>
<td>2.3.10 2.3.10</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>codemirror</td>
<td>5.39.2</td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td>commons-text</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>ee-first</td>
<td>1.1.1</td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Package Name</td>
<td>Version</td>
<td>Notes</td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>2005</td>
<td>txw2</td>
<td>2.3.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>2006</td>
<td>babel-plugin-transform-es2015-computed-properties</td>
<td>6.24.1 6.24.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>2007</td>
<td>once</td>
<td>1.4.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>2008</td>
<td>orc-shims</td>
<td>1.5.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>2009</td>
<td>qs</td>
<td>4.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>2010</td>
<td>regexp-quote</td>
<td>0.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>2011</td>
<td>string_decoder</td>
<td>1.1.1 1.1.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>2012</td>
<td>braces</td>
<td>3.0.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>2013</td>
<td>buffer-more-ints</td>
<td>0.0.2 0.0.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>2014</td>
<td>jsec</td>
<td>1.3.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>2015</td>
<td>moment</td>
<td>2.22.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>2016</td>
<td>encodeurl</td>
<td>1.0.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>2017</td>
<td>junit-platform-launcher</td>
<td>1.7.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>2018</td>
<td>generate-function</td>
<td>2.3.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>2019</td>
<td>drill-format-xml</td>
<td>1.19.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>2020</td>
<td>strip-bom</td>
<td>4.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>2021</td>
<td>js-base64</td>
<td>2.4.8</td>
<td>Available under license</td>
</tr>
<tr>
<td>2022</td>
<td>mocha</td>
<td>7.2.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>2023</td>
<td>elfutils</td>
<td>0.165 3ubuntu1.2</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.2024 semver 6.3.0
  1.2024.1 Available under license
1.2025 aggdesigner-algorithm 6.0
1.2026 mongodb-driver 3.12.4
  1.2026.1 Available under license
1.2027 accepts 1.2.13
  1.2027.1 Available under license
1.2028 jsonwebtoken 7.4.3 7.4.3
  1.2028.1 Available under license
1.2029 lodash.curry 4.1.1 4.1.1
  1.2029.1 Available under license
1.2030 drill-format-esri 1.19.0
  1.2030.1 Available under license
1.2031 drill-format-excel 1.19.0
  1.2031.1 Available under license
1.2032 babel-plugin-transform-object-rest-spread 6.26.0
  1.2032.1 Available under license
1.2033 path-to-regexp 0.1.7 0.1.7
  1.2033.1 Available under license
1.2034 asn1 0.2.4 0.2.4
  1.2034.1 Available under license
1.2035 is-unc-path 1.0.0
  1.2035.1 Available under license
1.2036 websocket-api 9.3.27.v20190418
1.2037 to-fast-properties 1.0.3
  1.2037.1 Available under license
1.2038 string-width 1.0.2
  1.2038.1 Available under license
1.2039 repeating 2.0.1
  1.2039.1 Available under license
1.2040 after 0.8.2
  1.2040.1 Available under license
1.2041 on-headers 1.0.2
  1.2041.1 Available under license
1.2042 atob 2.1.2
  1.2042.1 Available under license
1.2043 has-flag 4.0.0
  1.2043.1 Available under license
1.2044 unescape 1.0.1
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>array-unique</td>
<td>0.3.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>slf4j</td>
<td>1.7.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>busboy</td>
<td>0.2.14</td>
<td>Available under license</td>
</tr>
<tr>
<td>strip-ansi</td>
<td>0.1.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>babel-generator</td>
<td>6.26.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>jakarta xml bind api</td>
<td>2.3.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>@babel/plugin-proposal-json-strings</td>
<td>7.13.8</td>
<td>Available under license</td>
</tr>
<tr>
<td>lcms</td>
<td>2.6</td>
<td>3ubuntu2.1</td>
</tr>
<tr>
<td>source-map-support</td>
<td>0.4.18</td>
<td>0.4.18</td>
</tr>
<tr>
<td>get-stream</td>
<td>4.1.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>jsoup</td>
<td>1.9.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>sketches-core</td>
<td>0.9.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>aws-java-sdk-sts</td>
<td>1.11.837</td>
<td>Available under license</td>
</tr>
<tr>
<td>junit</td>
<td>4.13.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>pdfjs-dist</td>
<td>2.0.489</td>
<td>Available under license</td>
</tr>
<tr>
<td>jnr-posix</td>
<td>3.0.44</td>
<td>Available under license</td>
</tr>
<tr>
<td>big-integer</td>
<td>1.6.48</td>
<td>Available under license</td>
</tr>
<tr>
<td>isarray</td>
<td>0.0.1</td>
<td>0.0.1</td>
</tr>
<tr>
<td>color-name</td>
<td>1.1.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>Package Name</td>
<td>Version</td>
<td>License</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>text-table</td>
<td>0.2.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>proxy-from-env</td>
<td>1.1.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>jersey-mvc-freemarker</td>
<td>2.25.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>jersey-mvc</td>
<td>2.25.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>decamelize</td>
<td>1.2.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>sqlite</td>
<td>3.21.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>antlr</td>
<td>3.5.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>safe-buffer</td>
<td>5.1.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>is-fullwidth-code-point</td>
<td>3.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>denque</td>
<td>1.5.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>async</td>
<td>2.6.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>hadoop</td>
<td>1.2.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>has</td>
<td>1.0.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>saslprep</td>
<td>1.0.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>hbase-shaded-miscellaneous</td>
<td>3.4.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>co</td>
<td>4.6.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>lodash-flatten</td>
<td>4.4.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>string-prototype-trimstart</td>
<td>1.0.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>dnsjava</td>
<td>2.1.7</td>
<td>Available under license</td>
</tr>
<tr>
<td>path-loader</td>
<td>1.0.10</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.2083.1 Available under license
1.2084.1 Available under license
1.2085.1 Available under license
1.2086.1 Available under license
1.2087.1 Available under license
1.2088.1 Available under license
1.2089.1 Available under license
1.2090.1 Available under license
1.2091.1 Available under license
1.2092.1 Available under license
1.2093.1 Available under license
1.2094.1 Available under license
1.2095.1 Available under license
1.2096.1 Available under license
1.2097.1 Available under license
1.2098.1 Available under license
1.2099.1 Available under license
1.2100.1 Available under license
1.2101.1 Available under license
1.2102.1 Available under license
Open Source Used In DNAC Platform Antelope

1.2103 @babel/helper-validator-identifier 7.12.11
   1.2103.1 Available under license
1.2104 keyv 3.1.0
   1.2104.1 Available under license
1.2105 supports-color 7.2.0
   1.2105.1 Available under license
1.2106 merge-descriptors 1.0.1 1.0.1
   1.2106.1 Available under license
1.2107 hadoop-hdfs 3.2.2
   1.2107.1 Available under license
1.2108 normalize-path 2.1.1
   1.2108.1 Available under license
1.2109 winston 2.4.5
   1.2109.1 Available under license
1.2110 posix-character-classes 0.1.1
   1.2110.1 Available under license
1.2111 esprima 4.0.1
   1.2111.1 Available under license
1.2112 binary-extensions 2.1.0
   1.2112.1 Available under license
1.2113 harfbuzz 1.0.1-1ubuntu0.1
   1.2113.1 Available under license
1.2114 express 4.13.4
   1.2114.1 Available under license
1.2115 lodash.isequal 4.5.0
   1.2115.1 Available under license
1.2116 big.js 5.2.2 5.2.2
   1.2116.1 Available under license
1.2117 jetty-servlets 9.4.17.v20190418
   1.2117.1 Available under license
1.2118 d 1.0.1
   1.2118.1 Available under license
1.2119 commons-lang3 2.4
   1.2119.1 Available under license
1.2120 node-extend-without-arrays 1.1.6
   1.2120.1 Available under license
1.2121 jsreport-reports 2.5.1
   1.2121.1 Available under license
1.2122 buffers 0.1.1
1.2122.1 Available under license
1.2123 bluebird 3.4.7
    1.2123.1 Available under license
1.2124 fill-range 7.0.1
    1.2124.1 Available under license
1.2125 hive-shims-common 3.1.2
    1.2125.1 Available under license
1.2126 regenerator-transform 0.10.1
    1.2126.1 Available under license
1.2127 ansi-align 3.0.0
    1.2127.1 Available under license
1.2128 run-async 2.4.1
    1.2128.1 Available under license
1.2129 form-data 1.0.0-rc4
    1.2129.1 Available under license
1.2130 strip-ansi 5.2.0
    1.2130.1 Available under license
1.2131 regenerate-unicode-properties 8.2.0
    1.2131.1 Available under license
1.2132 tunnel-agent 0.6.0 0.6.0
    1.2132.1 Available under license
1.2133 lodash-filter 4.6.0
    1.2133.1 Available under license
1.2134 brace-expansion 1.1.11 1.1.11
    1.2134.1 Available under license
1.2135 strip-bom 3.0.0
    1.2135.1 Available under license
1.2136 fragment-cache 0.2.1
    1.2136.1 Available under license
1.2137 commons-configuration2 2.1.1
    1.2137.1 Available under license
1.2138 xmlenc 0.52
    1.2138.1 Available under license
1.2139 dns-prefetch-control 0.1.0
    1.2139.1 Available under license
1.2140 brace 0.11.1 0.11.1
    1.2140.1 Available under license
1.2141 aws-java-sdk-dynamodb 1.11.375
    1.2141.1 Available under license
<table>
<thead>
<tr>
<th>Version</th>
<th>Package Name</th>
<th>Version</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2142</td>
<td>parquet-jackson</td>
<td>1.8.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2142.1</td>
<td></td>
</tr>
<tr>
<td>1.2143</td>
<td>static-eval</td>
<td>2.0.2</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.0.2.1</td>
<td></td>
</tr>
<tr>
<td>1.2144</td>
<td>lodash-omit</td>
<td>4.5.0</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2144.1</td>
<td></td>
</tr>
<tr>
<td>1.2145</td>
<td>@babel/plugin-transform-unicode-escapes</td>
<td>7.12.13</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2145.1</td>
<td></td>
</tr>
<tr>
<td>1.2146</td>
<td>yeast</td>
<td>0.1.2</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1.2.1</td>
<td></td>
</tr>
<tr>
<td>1.2147</td>
<td>hbase-shaded-jersey</td>
<td>3.4.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2147.1</td>
<td></td>
</tr>
<tr>
<td>1.2148</td>
<td>libevent</td>
<td>1.3.6</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2148.1</td>
<td></td>
</tr>
<tr>
<td>1.2149</td>
<td>@nicolo-ribaudo/chokidar-2</td>
<td>2.1.8-no-fsevents</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2149.1</td>
<td></td>
</tr>
<tr>
<td>1.2150</td>
<td>esprima</td>
<td>4.0.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.0.1.1</td>
<td></td>
</tr>
<tr>
<td>1.2151</td>
<td>hppc</td>
<td>0.7.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2151.1</td>
<td></td>
</tr>
<tr>
<td>1.2152</td>
<td>yargs</td>
<td>15.4.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.4.1.1</td>
<td></td>
</tr>
<tr>
<td>1.2153</td>
<td>pako</td>
<td>1.0.11</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2153.1</td>
<td></td>
</tr>
<tr>
<td>1.2154</td>
<td>jetty-continuation</td>
<td>9.3.27.v20190418</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2154.1</td>
<td></td>
</tr>
<tr>
<td>1.2155</td>
<td>get-intrinsic</td>
<td>1.1.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.1.1.1</td>
<td></td>
</tr>
<tr>
<td>1.2156</td>
<td>array-each</td>
<td>1.0.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2156.1</td>
<td></td>
</tr>
<tr>
<td>1.2157</td>
<td>base64-arraybuffer</td>
<td>0.1.4</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2157.1</td>
<td></td>
</tr>
<tr>
<td>1.2158</td>
<td>hk2-locator</td>
<td>2.6.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2158.1</td>
<td></td>
</tr>
<tr>
<td>1.2159</td>
<td>babel-helper-replace-supers</td>
<td>6.24.1</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2159.1</td>
<td></td>
</tr>
<tr>
<td>1.2160</td>
<td>url-parse-lax</td>
<td>3.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2160.1</td>
<td></td>
</tr>
<tr>
<td>1.2161</td>
<td>package-hash</td>
<td>4.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2161.1</td>
<td></td>
</tr>
<tr>
<td>Component Name</td>
<td>Version</td>
<td>License</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>Antelope</td>
<td>1.2161.1</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>jetty-security</td>
<td>9.4.41.v20210516</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>code-point-at</td>
<td>1.1.0</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>is-regex</td>
<td>1.1.1</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>resolve-from</td>
<td>5.0.0</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>cookie-parser</td>
<td>1.4.3</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>on-finished</td>
<td>2.3.0</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>jsreport-docx</td>
<td>2.9.0</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>http-errors</td>
<td>1.3.1</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>streamsearch</td>
<td>0.1.2</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>lodash</td>
<td>4.17.21</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>fbemitter</td>
<td>2.1.1</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>groovy-xml</td>
<td>3.0.8</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>config</td>
<td>1.3.3</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>acorn</td>
<td>2.7.0</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>batik-svgrasterizer</td>
<td>1.13</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>has-binary</td>
<td>0.1.7</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>path-key</td>
<td>3.1.1</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>groovy</td>
<td>3.0.8</td>
<td>Available under license</td>
<td></td>
</tr>
<tr>
<td>jsreport-assets</td>
<td>1.7.0</td>
<td>Available under license</td>
<td></td>
</tr>
</tbody>
</table>
1.2181 metrics 4.0.2
1.2182 process-nextick-args 1.0.7
  1.2182.1 Available under license
1.2183 moment 2.22.1
  1.2183.1 Available under license
1.2184 swagger-integration 2.0.7
  1.2184.1 Available under license
1.2185 xml-beans 4.0.0
  1.2185.1 Available under license
1.2186 cups 2.1.3 4ubuntu0.10
  1.2186.1 Available under license
  1.2187.1 Available under license
1.2188 uuid 3.4.0
1.2189 decode-uri-component 0.2.0 0.2.0
  1.2189.1 Available under license
1.2190 hbase-metrics 2.4.2
  1.2190.1 Available under license
1.2191 annotations 3.0.1
1.2192 toposort 2.0.2 2.0.2
  1.2192.1 Available under license
1.2193 ignite-indexing 2.11.0
  1.2193.1 Available under license
1.2194 react 16.4.2 16.4.2
  1.2194.1 Available under license
1.2195 rimraf 2.4.5
  1.2195.1 Available under license
1.2196 remark 1.0.0
  1.2196.1 Available under license
1.2197 xml2js-preserve-spaces 0.0.1
  1.2197.1 Available under license
1.2198 asynckit 0.4.0 0.4.0
  1.2198.1 Available under license
1.2199 libxtst 1.2.2-1
  1.2199.1 Available under license
1.2200 jersey-bean-validation 2.5.1
1.2201 aws-java-sdk-s3 1.11.704
  1.2201.1 Available under license
1.2202 is-bigint 1.0.1
1.2202.1 Available under license

1.2203 jaxb-api 2.3.1
   1.2203.1 Available under license

1.2204 toidentifier 1.0.0
   1.2204.1 Available under license

1.2205 fast-deep-equal 2.0.1
   1.2205.1 Available under license

1.2206 proxy-addr 1.0.10 1.0.10
   1.2206.1 Available under license

1.2207 math-random 1.0.4
   1.2207.1 Available under license

1.2208 is-callable 1.2.3
   1.2208.1 Available under license

1.2209 xml2js 0.4.23 0.4.23
   1.2209.1 Available under license

1.2210 jcip-annotation 1.0
   1.2210.1 Available under license

1.2211 json-path 1.1.1 1.1.1
   1.2211.1 Available under license

1.2212 jackson-jaxrs-xml-provider 2.9.10
   1.2212.1 Available under license

1.2213 nocache 1.0.1 1.0.1
   1.2213.1 Available under license

1.2214 zjsonpatch 0.3.6
   1.2214.1 Available under license

1.2215 lucene-analyzers-common 7.4.0
   1.2215.1 Available under license

1.2216 synchronous-promise 2.0.15 2.0.15
   1.2216.1 Available under license

1.2217 batik-svggen 1.13
   1.2217.1 Available under license

1.2218 ecc-jsbn 0.1.2 0.1.2
   1.2218.1 Available under license

1.2219 onetime 1.1.0
   1.2219.1 Available under license

   1.2220.1 Available under license

1.2221 fresh 0.3.0
   1.2221.1 Available under license
1.2222 has-values 0.1.4
   1.2222.1 Available under license
1.2223 jacoco-agent 0.7.9
   1.2223.1 Available under license
1.2224 parseurl 1.3.2 1.3.2
   1.2224.1 Available under license
1.2225 netty-resolver 4.1.42.Final
   1.2225.1 Available under license
1.2226 netty-buffer 4.1.42.Final
   1.2226.1 Available under license
1.2227 kotlin 1.3.72
   1.2227.1 Available under license
1.2228 hbase-shaded-gson 3.4.1
   1.2228.1 Available under license
1.2229 pinkie-promise 1.0.0 1.0.0
   1.2229.1 Available under license
1.2230 ip-regex 1.0.3
   1.2230.1 Available under license
1.2231 openssl 1.0.2m
   1.2231.1 Available under license
1.2232 @jsreport/mingo 2.4.1
   1.2232.1 Available under license
1.2233 fs-writefile-promise 1.0.3 1.0.3
   1.2233.1 Available under license
1.2234 openjpeg 1.6.35
   1.2234.1 Available under license
1.2235 p-reduce 2.1.0
   1.2235.1 Available under license
1.2236 kudu-client 1.14.0
   1.2236.1 Available under license
1.2237 domelementtype 1.3.1
   1.2237.1 Available under license
1.2238 semver-diff 3.1.1
   1.2238.1 Available under license
1.2239 guava 30.1-jre
   1.2239.1 Available under license
1.2240 kind-of 4.0.0
   1.2240.1 Available under license
1.2241 opentype-js 0.8.0
1.2241.1 Available under license

1.2242 parquet-hadoop 1.8.1
1.2242.1 Available under license

1.2243 babel-helper-hoist-variables 6.24.1 6.24.1
1.2243.1 Available under license

1.2244 libjpeg-turbo 1.4.2-0ubuntu3.3
1.2244.1 Available under license

1.2245 hoek 2.16.3
1.2245.1 Available under license

1.2246 cookie-signature 1.0.6
1.2246.1 Available under license

1.2247 growl 1.10.5
1.2247.1 Available under license

1.2248 oas-schema-walker 1.1.5 1.1.5
1.2248.1 Available under license

1.2249 shimmer 1.0.0 1.0.0
1.2249.1 Available under license

1.2250 babel-plugin-polyfill-regenerator 0.1.6
1.2250.1 Available under license

1.2251 hk2-locator 2.2.0-b21
1.2251.1 Available under license

1.2252 y18n 4.0.0
1.2252.1 Available under license

1.2253 oas-validator 5.0.5 5.0.5
1.2253.1 Available under license

1.2254 joda-time 2.9.9
1.2254.1 Available under license

1.2255 netty-transport-native-unix-common 4.1.42.Final
1.2255.1 Available under license

1.2256 process-on-spawn 1.0.0
1.2256.1 Available under license

1.2257 is-property 1.0.2
1.2257.1 Available under license

1.2258 reactive-streams 1.0.2
1.2258.1 Available under license

1.2259 batik-rasterizer 1.13
1.2259.1 Available under license

1.2260 hbase-shaded-netty 3.4.1
1.2260.1 Available under license
1.2261 stream-combiner 0.0.4 0.0.4
   1.2261.1 Available under license
1.2262 sshpk 1.16.1
   1.2262.1 Available under license
1.2263 node-environment-flags 1.0.6
   1.2263.1 Available under license
1.2264 y18n 3.2.1
   1.2264.1 Available under license
1.2265 object-inspect 1.8.0
   1.2265.1 Available under license
1.2266 form-data 2.3.3 2.3.3
   1.2266.1 Available under license
1.2267 dot-prop 5.3.0
   1.2267.1 Available under license
1.2268 escalate 3.1.1 3.1.1
   1.2268.1 Available under license
1.2269 string-width 4.2.2
   1.2269.1 Available under license
   1.2270.1 Available under license
1.2271 javax.servlet.jsp 2.3.2
   1.2271.1 Available under license
1.2272 fontbox 2.0.22
   1.2272.1 Available under license
1.2273 oauth-sign 0.9.0
   1.2273.1 Available under license
1.2274 jline-terminal-jna 3.12.1
   1.2274.1 Available under license
1.2275 fresh 0.3.0 0.3.0
   1.2275.1 Available under license
1.2276 jsr305 3.0.2
   1.2276.1 Available under license
1.2277 ignore 3.3.10
   1.2277.1 Available under license
1.2278 finalhandler 0.4.1
   1.2278.1 Available under license
1.2279 libbsd 0.8.2 1ubuntu0.1
   1.2279.1 Available under license
1.2280 json-refs 3.0.15 3.0.15
1.2280.1 Available under license

1.2281 lodash-pick 4.4.0
   1.2281.1 Available under license

1.2282 hpkp 1.2.0
   1.2282.1 Available under license

1.2283 javax-inject 1
   1.2283.1 Available under license

1.2284 reftools 1.1.8 1.1.8
   1.2284.1 Available under license

1.2285 destroy 1.0.4 1.0.4
   1.2285.1 Available under license

1.2286 iconv-lite 0.4.23 0.4.23
   1.2286.1 Available under license

1.2287 deepmerge 2.1.0
   1.2287.1 Available under license

1.2288 pre-push 0.1.1 0.1.1
   1.2288.1 Available under license

1.2289 asm-all-repackaged 2.2.0-b21
   1.2289.1 Available under license

1.2290 follow-redirects 1.5.10 1.5.10
   1.2290.1 Available under license

1.2291 ansi-styles 4.3.0 4.3.0
   1.2291.1 Available under license

1.2292 es-abstract 1.17.7
   1.2292.1 Available under license

1.2293 vector 1.19.0
   1.2293.1 Available under license

1.2294 esri-geometry-api 2.2.2
   1.2294.1 Available under license

1.2295 commons-pool 2.4.3
   1.2295.1 Available under license

1.2296 @babel/plugin-proposal-numeric-separator 7.12.13
   1.2296.1 Available under license

1.2297 detect-indent 4.0.0 4.0.0
   1.2297.1 Available under license

1.2298 window-size 0.1.4
   1.2298.1 Available under license

1.2299 scala-reflect 2.13.5
   1.2299.1 Available under license
<table>
<thead>
<tr>
<th>Version</th>
<th>Package Name</th>
<th>Version</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2300</td>
<td>nice-try</td>
<td>1.0.5</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2301</td>
<td>etag</td>
<td>1.7.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2302</td>
<td>colors.js</td>
<td>1.4.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2303</td>
<td>caller-path</td>
<td>0.1.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2304</td>
<td>is-number-object</td>
<td>1.0.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2305</td>
<td>activation-api</td>
<td>1.2.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2306</td>
<td>react-lifecycles-compat</td>
<td>3.0.4</td>
<td>3.0.4  Available under license</td>
</tr>
<tr>
<td>1.2307</td>
<td>base64-js</td>
<td>0.0.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2308</td>
<td>p-finally</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2309</td>
<td>@babel/helper-wrap-function</td>
<td>7.13.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2310</td>
<td>okhttp</td>
<td>4.9.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2311</td>
<td>jetty-xml</td>
<td>9.3.27.v20190418</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2312</td>
<td>scala</td>
<td>2.11.12</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2313</td>
<td>bson</td>
<td>4.2.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2314</td>
<td>thrift</td>
<td>0.9.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2315</td>
<td>kerb-util</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2316</td>
<td>setimmediate</td>
<td>1.0.5</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2317</td>
<td>acorn</td>
<td>5.7.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2318</td>
<td>reactor-core</td>
<td>3.0.7.RELEASE</td>
<td>Available under license</td>
</tr>
<tr>
<td>1.2319</td>
<td>aws-sign2</td>
<td>0.7.0</td>
<td>0.7.0 Available under license</td>
</tr>
</tbody>
</table>
1.2320 fn-name 3.0.0 3.0.0
   1.2320.1 Available under license
1.2321 esri-geometry-api 2.2.0
   1.2321.1 Available under license
1.2322 mkdir-promise 1.1.0 1.1.0
   1.2322.1 Available under license
1.2323 drill-memory-base 1.19.0
   1.2323.1 Available under license
1.2324 arr-union 3.1.0
   1.2324.1 Available under license
1.2325 batik-extension 1.13
   1.2325.1 Available under license
1.2326 kerb-simplekdc 1.0.0
   1.2326.1 Available under license
1.2327 lcid 1.0.0
   1.2327.1 Available under license
1.2328 @babel/helper-member-expression-to-functions 7.13.12
   1.2328.1 Available under license
1.2329 libsoup 2.52.2-1ubuntu0.3
   1.2329.1 Available under license
1.2330 lazystream 1.0.0
   1.2330.1 Available under license
1.2331 homedir-polyfill 1.0.3
   1.2331.1 Available under license
1.2332 annotations 13.0
   1.2332.1 Available under license
1.2333 caseless 0.12.0 0.12.0
   1.2333.1 Available under license
1.2334 lodash 4.17.10 4.17.10
   1.2334.1 Available under license
1.2335 chalk 3.0.0
   1.2335.1 Available under license
1.2336 extsprintf 1.4.0 1.4.0
   1.2336.1 Available under license
1.2337 sax 1.2.4 1.2.4
   1.2337.1 Available under license
1.2338 xmlbuilder 9.0.4
   1.2338.1 Available under license
1.2339 isstream 0.1.2
1.2339.1 Available under license
1.2340 **abbrev 1.1.1** 1.1.1
   1.2340.1 Available under license
1.2341 **babel-plugin-dynamic-import-node 2.3.3**
   1.2341.1 Available under license
1.2342 **calcite-core 1.21.0-drill-r2**
   1.2342.1 Available under license
1.2343 **escodegen 1.14.3** 1.14.3
   1.2343.1 Available under license
1.2344 **spdx-license-ids 3.0.6**
   1.2344.1 Available under license
1.2345 **is-string 1.0.5**
   1.2345.1 Available under license
1.2346 **touch 3.1.0**
   1.2346.1 Available under license
1.2347 **express 4.17.1**
   1.2347.1 Available under license
1.2348 **jsreport-child-templates 1.4.0**
   1.2348.1 Available under license
1.2349 **glob-parent 5.1.1**
   1.2349.1 Available under license
1.2350 **node-promise 0.5.13 0.5.13**
   1.2350.1 Available under license
1.2351 **safer-buffer 2.1.2**
   1.2351.1 Available under license
1.2352 **jsreport-version-control 1.6.0**
   1.2352.1 Available under license
1.2353 **jquery 3.4.0 3.4.0**
   1.2353.1 Available under license
1.2354 **get-stream 5.1.0**
   1.2354.1 Available under license
1.2355 **commons-codec 1.13**
   1.2355.1 Available under license
1.2356 **babel-plugin-transform-es2015-destructuring 6.23.0**
   1.2356.1 Available under license
1.2357 **hadoop-auth 3.2.2**
   1.2357.1 Available under license
1.2358 **babel-code-frame 6.26.0 6.26.0**
   1.2358.1 Available under license
1.2359 my-scope-package-a 0.0.0
   1.2359.1 Available under license

1.2360 uuid 3.2.1
   1.2360.1 Available under license

1.2361 babel-helper-compilation-targets 7.13.13
   1.2361.1 Available under license

1.2362 xpp 1.1.4c
   1.2362.1 Notifications
   1.2362.2 Available under license

1.2363 path-dirname 1.0.2
   1.2363.1 Available under license

1.2364 pi-spec-converter-web 0.0.0 0.0.0
   1.2364.1 Available under license

1.2365 kryo-shaded 3.0.3

1.2366 @babel/traverse 7.11.5
   1.2366.1 Available under license

1.2367 jsprim 1.4.1 1.4.1
   1.2367.1 Available under license

1.2368 es6-symbol 3.1.3
   1.2368.1 Available under license

1.2369 use 3.1.1
   1.2369.1 Available under license

1.2370 sqlite 3.25.3
   1.2370.1 Available under license

1.2371 which 1.3.1
   1.2371.1 Available under license

1.2372 semver 5.5.0 5.5.0
   1.2372.1 Available under license

1.2373 entities 1.1.2
   1.2373.1 Available under license

1.2374 accepts 1.3.3
   1.2374.1 Available under license

1.2375 ipaddr.js 1.9.1
   1.2375.1 Available under license

1.2376 har-validator 5.1.5 5.1.5
   1.2376.1 Available under license

1.2377 ant 1.10.9
   1.2377.1 Available under license

1.2378 pluralize 1.1.6 1.1.6
1.2378.1 Available under license
1.2379 **hk2-utils 2.2.0-b21**
   1.2379.1 Available under license
1.2380 **amdefine 1.0.1**
   1.2380.1 Available under license
1.2381 **read-all-stream 1.0.2 1.0.2**
   1.2381.1 Available under license
1.2382 **hadoop-mapreduce-client-core 3.2.2**
   1.2382.1 Available under license
1.2383 **istanbul-reports 3.0.2**
   1.2383.1 Available under license
1.2384 **@babel/plugin-transform-spread 7.13.0**
   1.2384.1 Available under license
1.2385 **for-in 1.0.2**
   1.2385.1 Available under license
1.2386 **raw-body 2.4.0**
   1.2386.1 Available under license
1.2387 **object-keys 1.1.1**
   1.2387.1 Available under license
1.2388 **readable-stream 2.3.7**
   1.2388.1 Available under license
1.2389 **pure-color 1.3.0 1.3.0**
   1.2389.1 Available under license
1.2390 **babel-runtime 6.26.0 6.26.0**
   1.2390.1 Available under license
1.2391 **is-accessor-descriptor 1.0.0**
   1.2391.1 Available under license
1.2392 **to-regex-range 5.0.1**
   1.2392.1 Available under license
1.2393 **lockfile 1.0.4**
   1.2393.1 Available under license
1.2394 **source-map 0.2.0**
   1.2394.1 Available under license
1.2395 **annotations 2.0.1**
   1.2395.1 Available under license
1.2396 **json-stringify-safe 5.0.1**
   1.2396.1 Available under license
1.2397 **core 3.1.1**
   1.2397.1 Available under license
1.2398 qs 6.5.1 6.5.1
  1.2398.1 Available under license
1.2399 kite-hadoop-compatibility 1.1.0
  1.2399.1 Available under license
1.2400 cross-spawn 7.0.3
  1.2400.1 Available under license
1.2401 eslint-config-airbnb-base 9.0.0
  1.2401.1 Available under license
1.2402 kotlin 1.4.0
  1.2402.1 Available under license
1.2403 ajv 4.11.8
  1.2403.1 Available under license
1.2404 esutils 2.0.3
  1.2404.1 Available under license
1.2405 redux-thunk 2.3.0 2.3.0
  1.2405.1 Available under license
1.2406 circular-json 0.3.3
  1.2406.1 Available under license
1.2407 listenablefuture 9999.0-empty-to-avoid-conflict-with-guava
  1.2407.1 Available under license
1.2408 amqplib-tutorials 0.0.1 0.0.1
  1.2408.1 Available under license
1.2409 set-immediate-shim 1.0.1
  1.2409.1 Available under license
1.2410 object-map 1.0.1
  1.2410.1 Available under license
1.2411 js-base64 2.6.4
  1.2411.1 Available under license
1.2412 pause 0.0.1
  1.2412.1 Available under license
1.2413 cookie 0.3.1
  1.2413.1 Available under license
1.2414 grpc-netty-shaded 1.27.1
  1.2414.1 Available under license
1.2415 @babel/code-frame 7.12.13
  1.2415.1 Available under license
1.2416 has-symbols 1.0.1 1.0.1
  1.2416.1 Available under license
1.2417 content-disposition 0.5.2 0.5.2
<table>
<thead>
<tr>
<th>1.2417.1 Available under license</th>
<th>1.2418 yargs-parser 20.2.7 20.2.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2419 parquet-avro 1.8.1</td>
<td>1.2419.1 Available under license</td>
</tr>
<tr>
<td>1.2420 url-parse 1.4.7</td>
<td>1.2420.1 Available under license</td>
</tr>
<tr>
<td>1.2421 hive-contrib 3.1.2</td>
<td>1.2421.1 Available under license</td>
</tr>
<tr>
<td>1.2422 ansi-regex 2.1.1 2.1.1</td>
<td>1.2422.1 Available under license</td>
</tr>
<tr>
<td>1.2423 babel-preset-es2015 6.24.1</td>
<td>1.2423.1 Available under license</td>
</tr>
<tr>
<td>1.2424 react-tagsinput 3.19.0 3.19.0</td>
<td>1.2424.1 Available under license</td>
</tr>
<tr>
<td>1.2425 path-is-absolute 1.0.1</td>
<td>1.2425.1 Available under license</td>
</tr>
<tr>
<td>1.2426 lodash-set 4.3.2</td>
<td>1.2426.1 Available under license</td>
</tr>
<tr>
<td>1.2427 object-keys 0.4.0 0.4.0</td>
<td>1.2427.1 Available under license</td>
</tr>
<tr>
<td>1.2428 json3 3.3.2 3.3.2</td>
<td>1.2428.1 Available under license</td>
</tr>
<tr>
<td>1.2429 string_decoder 1.1.1</td>
<td>1.2429.1 Available under license</td>
</tr>
<tr>
<td>1.2430 jackson-annotations 2.12.3</td>
<td>1.2430.1 Available under license</td>
</tr>
<tr>
<td>1.2431 forwarded 0.1.2</td>
<td>1.2431.1 Available under license</td>
</tr>
<tr>
<td>1.2432 commons-digester 2.1</td>
<td>1.2432.1 Available under license</td>
</tr>
<tr>
<td>1.2433 es6-promise 4.2.8</td>
<td>1.2433.1 Available under license</td>
</tr>
<tr>
<td>1.2434 repeat-element 1.1.3</td>
<td>1.2434.1 Available under license</td>
</tr>
<tr>
<td>1.2435 parquet-hadoop 1.10.0</td>
<td>1.2435.1 Available under license</td>
</tr>
<tr>
<td>1.2436 es6-iterator 2.0.3</td>
<td>1.2436.1 Available under license</td>
</tr>
</tbody>
</table>
1.2437 poi 5.0.0
   1.2437.1 Available under license
1.2438 rimraf 2.7.1
   1.2438.1 Available under license
1.2439 socket.io-adapter 0.5.0
   1.2439.1 Available under license
1.2440 parse-json 2.2.0
   1.2440.1 Available under license
1.2441 @babel/plugin-proposal-logical-assignment-operators 7.13.8
   1.2441.1 Available under license
1.2442 doctrine 1.5.0
   1.2442.1 Available under license
1.2443 dns-prefetch-control 0.1.0 0.1.0
   1.2443.1 Available under license
   1.2444.1 Available under license
1.2445 is-my-json-valid 2.20.5
   1.2445.1 Available under license
1.2446 kerby-asn1 1.0.0
   1.2446.1 Available under license
1.2447 ansi-styles 4.3.0
   1.2447.1 Available under license
1.2448 express 4.13.4 4.13.4
   1.2448.1 Available under license
1.2449 react-highlight-words 0.17.0 0.17.0
   1.2449.1 Available under license
1.2450 zstd-jni 1.4.9-1
   1.2450.1 Available under license
1.2451 ret 0.1.15
   1.2451.1 Available under license
1.2452 extglob 0.3.2
   1.2452.1 Available under license
1.2453 figures 1.7.0
   1.2453.1 Available under license
1.2454 mockito-core 2.23.4
   1.2454.1 Available under license
1.2455 lodash.flow 3.5.0 3.5.0
   1.2455.1 Available under license
1.2456 aspect-j 1.8.13
1.2456.1 Available under license
1.2457 `elasticsearch-rest-client 7.0.1`
   1.2457.1 Available under license
1.2458 `@babel/plugin-transform-regenerator 7.12.13`
   1.2458.1 Available under license
1.2459 `amqplib 0.5.2 0.5.2`
   1.2459.1 Available under license
1.2460 `curl 7.51.0`
   1.2460.1 Available under license
1.2461 `xmpcore 6.0.6`
   1.2461.1 Available under license
1.2462 `rx-lite 3.1.2`
   1.2462.1 Available under license
1.2463 `is-path-inside 3.0.3`
   1.2463.1 Available under license
1.2464 `@babel/helper-remap-async-to-generator 7.13.0`
   1.2464.1 Available under license
1.2465 `asynckit 0.4.0`
   1.2465.1 Available under license
1.2466 `ipaddr.js 1.0.5 1.0.5`
   1.2466.1 Available under license
1.2467 `combined-stream 1.0.8`
   1.2467.1 Available under license
1.2468 `simple-odata-server 1.1.1`
   1.2468.1 Available under license
1.2469 `@babel/plugin-transform-async-to-generator 7.13.0`
   1.2469.1 Available under license
1.2470 `@babel/plugin-transform-new-target 7.12.13`
   1.2470.1 Available under license
1.2471 `drill-format-httpd 1.19.0`
   1.2471.1 Available under license
1.2472 `jetty-servlets 9.3.27.v20190418`
1.2473 `byte-buddy 1.9.3`
1.2474 `derby 10.14.1.0`
   1.2474.1 Available under license
1.2475 `base64id 2.0.0`
   1.2475.1 Available under license
1.2476 `react-redux 5.0.7 5.0.7`
   1.2476.1 Available under license
1.2477 p-cancelable 1.1.0
   1.2477.1 Available under license
1.2478 metrics 4.1.0
   1.2478.1 Available under license
1.2479 object.assign 4.1.0
   1.2479.1 Available under license
1.2480 serve-static 1.10.3
   1.2480.1 Available under license
1.2481 lodash 4.17.20 4.17.20
   1.2481.1 Available under license
1.2482 univocity-parsers 2.8.3
   1.2482.1 Available under license
1.2483 llvm 6.0-1ubuntu2~16.04.1
   1.2483.1 Available under license
1.2484 sprintf-js 1.0.3 1.0.3
   1.2484.1 Available under license
1.2485 @babel/plugin-transform-parameters 7.13.0
   1.2485.1 Available under license
1.2486 pip 9.0.1
   1.2486.1 Available under license
1.2487 mimic-response 2.1.0
   1.2487.1 Available under license
1.2488 aws4 1.11.0 1.11.0
   1.2488.1 Available under license
1.2489 jetty-webapp 9.4.34.v20201102
   1.2489.1 Available under license
1.2490 locate-path 5.0.0
   1.2490.1 Available under license
1.2491 json3 3.3.2
   1.2491.1 Available under license
1.2492 lodash.get 4.4.2
   1.2492.1 Available under license
1.2493 groovy-json 3.0.8
   1.2493.1 Available under license
1.2494 express 4.16.3 4.16.3
   1.2494.1 Available under license
1.2495 avatica-metrics 1.15.0
   1.2495.1 Available under license
1.2496 tephra-hbase-compat-1.0 0.6.0

Open Source Used In DNAC Platform Antelope 127
1.2496.1 Available under license

1.2497 **groovyjsr 3.0.8**
   1.2497.1 Available under license

1.2498 **jsesc 1.3.0**
   1.2498.1 Available under license

1.2499 **ipaddr.js 1.0.5**
   1.2499.1 Available under license

1.2500 **@babel/highlight 7.0.0**
   1.2500.1 Available under license

1.2501 **guava 22.0-android**
   1.2501.1 Available under license

1.2502 **swagger-core 1.5.0**

1.2503 **dropwizard-metrics-hadoop-metrics2-reporter 0.1.2**
   1.2503.1 Available under license

1.2504 **boolbase 1.0.0**
   1.2504.1 Available under license

1.2505 **spdx-exceptions 2.3.0**
   1.2505.1 Available under license

1.2506 **@babel/helper-module-imports 7.13.12**
   1.2506.1 Available under license

1.2507 **ignite-core 2.11.0**
   1.2507.1 Available under license

1.2508 **babel-helper-regex 6.26.0 6.26.0**
   1.2508.1 Available under license

1.2509 **formidable 1.2.2 1.2.2**
   1.2509.1 Available under license

1.2510 **cache-api 1.0.0**
   1.2510.1 Available under license

1.2511 **lodash.reduce 4.5.0**
   1.2511.1 Available under license

1.2512 **jctools-core 3.1.0**
   1.2512.1 Available under license

1.2513 **influx 5.6.3**
   1.2513.1 Available under license

1.2514 **ms 2.1.1**
   1.2514.1 Available under license

1.2515 **espree 3.5.4**
   1.2515.1 Available under license

1.2516 **bouncy-castle 1.57**
1.2516.1 Available under license

1.2517 array-unique 0.2.1
1.2517.1 Available under license

1.2518 string-prototype-trimend 1.0.4
1.2518.1 Available under license

1.2519 istanbul-lib-hook 3.0.0
1.2519.1 Available under license

1.2520 split-on-first 1.1.0 1.1.0
1.2520.1 Available under license

1.2521 hk2-locator 2.5.0-b32
1.2521.1 Available under license

1.2522 inflight 1.0.6
1.2522.1 Available under license

1.2523 babel-helper-define-map 6.26.0 6.26.0
1.2523.1 Available under license

1.2524 util-deprecate 1.0.2
1.2524.1 Available under license

1.2525 is-number 3.0.0
1.2525.1 Available under license

1.2526 string_decoder 1.0.3
1.2526.1 Available under license

1.2527 request-promise 4.2.6
1.2527.1 Available under license

1.2528 binary 0.3.0
1.2528.1 Available under license

1.2529 semver 7.0.0
1.2529.1 Available under license

1.2530 p-limit 2.3.0 2.3.0
1.2530.1 Available under license

1.2531 jsrender 1.0.4
1.2531.1 Available under license

1.2532 has-color 0.1.7 0.1.7
1.2532.1 Available under license

1.2533 setprototypeof 1.0.3 1.0.3
1.2533.1 Available under license

1.2534 jline 2.9

1.2535 jetty-servlet 9.4.34.v20201102
1.2535.1 Available under license

1.2536 html-to-xlsx 2.0.3
1.2536.1 Available under license
1.2537 depd 1.1.2 1.1.2
1.2537.1 Available under license
1.2538 parseurl 1.3.3
1.2538.1 Available under license
1.2539 type 2.1.0
1.2539.1 Available under license
1.2540 map-visit 1.0.0
1.2540.1 Available under license
1.2541 regenerator-runtime 0.11.1
1.2541.1 Available under license
1.2542 bonecp 0.8.0.RELEASE
1.2542.1 Available under license
1.2543 batik-ttf2svg 1.13
1.2543.1 Available under license
1.2544 error-ex 1.3.2
1.2544.1 Available under license
1.2545 shebang-regex 1.0.0
1.2545.1 Available under license
1.2546 is-glob 4.0.1
1.2546.1 Available under license
1.2547 from 0.1.7 0.1.7
1.2547.1 Available under license
1.2548 ansi-regex 4.1.0
1.2548.1 Available under license
1.2549 ws 1.1.5
1.2549.1 Available under license
1.2550 has-symbols 1.0.0
1.2550.1 Available under license
1.2551 estraverse 4.3.0 4.3.0
1.2551.1 Available under license
1.2552 range-parser 1.2.0 1.2.0
1.2552.1 Available under license
1.2553 drill-format-pcapng 1.19.0
1.2553.1 Available under license
1.2554 content-type 1.0.4
1.2554.1 Available under license
1.2555 nth-check 1.0.2
1.2555.1 Available under license
1.2556 node-promise 0.5.14
   1.2556.1 Available under license
1.2557 is-typedarray 1.0.0 1.0.0
   1.2557.1 Available under license
1.2558 to-array 0.1.4 0.1.4
   1.2558.1 Available under license
1.2559 libpng 1.6.35
   1.2559.1 Available under license
1.2560 user-home 2.0.0
   1.2560.1 Available under license
1.2561 jackson-dataformat-csv 2.10.5
   1.2561.1 Available under license
1.2562 minimist 1.2.0 1.2.0
   1.2562.1 Available under license
1.2563 is-finite 1.0.2 1.0.2
   1.2563.1 Available under license
1.2564 defer-to-connect 1.1.3
   1.2564.1 Available under license
1.2565 guice 5.0.1
   1.2565.1 Available under license
1.2566 parquet-column 1.10.0
   1.2566.1 Available under license
1.2567 is-core-module 2.0.0
   1.2567.1 Available under license
1.2568 process-nextick-args 2.0.1 2.0.1
   1.2568.1 Available under license
1.2569 moment-timezone 0.5.33 0.5.33
   1.2569.1 Available under license
1.2570 hamcrest 1.3
   1.2570.1 Available under license
1.2571 commons-math 3.1.1
   1.2571.1 Available under license
1.2572 jetty-xml 9.4.41.v20210516
   1.2572.1 Available under license
1.2573 parseuri 0.0.5
   1.2573.1 Available under license
1.2574 extglob 2.0.4
   1.2574.1 Available under license
1.2575 yargs 13.2.4
1.2575.1 Available under license
1.2576 z-schema 3.25.1 3.25.1
  1.2576.1 Available under license
1.2577 nss 3.28.4-0ubuntu0.16.04.10
  1.2577.1 Available under license
1.2578 is-fullwidth-code-point 2.0.0
  1.2578.1 Available under license
1.2579 intl-relativeformat 2.1.0 2.1.0
  1.2579.1 Available under license
1.2580 babel-plugin-proposal-optional-chaining 7.13.12
  1.2580.1 Available under license
1.2581 @babel/plugin-transform-dotall-regex 7.12.13
  1.2581.1 Available under license
1.2582 jsreport-static-pdf 0.4.0
  1.2582.1 Available under license
1.2583 is-binary-path 1.0.1
  1.2583.1 Available under license
1.2584 apache-log4j 2.8.2
  1.2584.1 Available under license
1.2585 listenablefuture 1.0
  1.2585.1 Available under license
1.2586 aspectj-tools 1.8.13
1.2587 ajv 6.10.0
  1.2587.1 Available under license
1.2588 verror 1.10.0
  1.2588.1 Available under license
1.2589 esquery 1.3.1
  1.2589.1 Available under license
1.2590 superagent 3.8.3 3.8.3
  1.2590.1 Available under license
1.2591 babel-plugin-syntax-top-level-await 7.12.13
  1.2591.1 Available under license
1.2592 isemail 1.2.0 1.2.0
  1.2592.1 Available under license
1.2593 cookie 0.4.0
  1.2593.1 Available under license
1.2594 guava 25.0-jre
  1.2594.1 Available under license
1.2595 ono 4.0.11 4.0.11
Open Source Used In DNAC Platform Antelope

1.2595.1 Available under license
**1.2596 chalk 2.4.1**
1.2596.1 Available under license
**1.2597 jersey-media-json-jackson 2.25.1**
1.2597.1 Available under license
**1.2598 electron-to-chromium 1.3.690**
1.2598.1 Available under license
**1.2599 @babel/helper-plugin-utils 7.13.0**
1.2599.1 Available under license
**1.2600 ipaddr.js 1.8.0 1.8.0**
1.2600.1 Available under license
**1.2601 commander 4.1.1**
1.2601.1 Available under license
**1.2602 deep-eql 3.0.1**
1.2602.1 Available under license
**1.2603 jersey-guice 1.19**
**1.2604 raw-body 2.3.2 2.3.2**
1.2604.1 Available under license
**1.2605 drill-storage-http 1.19.0**
1.2605.1 Available under license
**1.2606 wrappy 1.0.2 1.0.2**
1.2606.1 Available under license
**1.2607 memo-parser 0.2.1**
1.2607.1 Available under license
**1.2608 streamifier 0.1.1**
1.2608.1 Available under license
**1.2609 object.pick 1.3.0**
1.2609.1 Available under license
**1.2610 kerby-config 1.0.0**
1.2610.1 Available under license
**1.2611 yargs-unparser 1.6.0**
1.2611.1 Available under license
**1.2612 drill-udfs 1.19.0**
1.2612.1 Available under license
**1.2613 call-bind 1.0.2 1.0.2**
1.2613.1 Available under license
**1.2614 debug 2.6.9 2.6.9**
1.2614.1 Available under license
**1.2615 is-regex 1.1.2**
1.2615.1 Available under license

1.2616 estraverse 4.3.0
1.2616.1 Available under license

1.2617 dont-sniff-mimetype 1.0.0 1.0.0
1.2617.1 Available under license

1.2618 buffer 5.6.0
1.2618.1 Available under license

1.2619 is-extendable 1.0.1
1.2619.1 Available under license

1.2620 snapdragon 0.8.2
1.2620.1 Available under license

1.2621 pango 1.38.1-1
1.2621.1 Available under license

1.2622 jackson-jaxrs 2.9.10
1.2622.1 Available under license

1.2623 sf4j-ext 1.7.25
1.2624 gson4jackson 2.7.0
1.2624.1 Available under license

1.2625 batik-i18n 1.13
1.2625.1 Available under license

1.2626 metrics-jvm 4.1.0
1.2627 okio 2.8.0
1.2627.1 Available under license

1.2628 es6-weak-map 2.0.3
1.2628.1 Available under license

1.2629 @babel/cli 7.13.14
1.2629.1 Available under license

1.2630 @babel/plugin-transform-destructuring 7.13.0
1.2630.1 Available under license

1.2631 babel-plugin-transform-exponentiation-operator 6.24.1
1.2631.1 Available under license

1.2632 hosted-git-info 2.8.8
1.2632.1 Available under license

1.2633 json-schema-traverse 0.4.1
1.2633.1 Available under license

1.2634 minimist 1.2.5
1.2634.1 Available under license

1.2635 babel-plugin-syntax-exponentiation-operator 6.13.0
1.2635.1 Available under license
1.2636 v8flags 2.1.1
1.2636.1 Available under license

1.2637 inflight 1.0.6 1.0.6
1.2637.1 Available under license

1.2638 @babel/helper-function-name 7.10.4
1.2638.1 Available under license

1.2639 amqp-client 4.8.0
1.2639.1 Available under license

1.2640 type-check 0.3.2 0.3.2
1.2640.1 Available under license

1.2641 underscore 1.13.1 1.13.1
1.2641.1 Available under license

1.2642 json-schema 0.2.3
1.2642.1 Available under license

1.2643 @babel/helper-optimise-call-expression 7.12.13
1.2643.1 Available under license

1.2644 javax-ws-rs-api 2.0.1
1.2644.1 Available under license

1.2645 optional-require 1.0.3 1.0.3
1.2645.1 Available under license

1.2646 jersey 2.5.1
1.2646.1 Available under license

1.2647 should-type-adaptors 1.1.0 1.1.0
1.2647.1 Available under license

1.2648 @babel/preset-env 7.13.12
1.2648.1 Available under license

1.2649 estraverse 5.2.0
1.2649.1 Available under license

1.2650 should-format 3.0.3 3.0.3
1.2650.1 Available under license

1.2651 es6-promisify 5.0.0
1.2651.1 Available under license

1.2652 lm-sensors 3.4.0-2
1.2652.1 Available under license

1.2653 helmet 2.3.0
1.2653.1 Available under license

1.2654 jsreport-client 1.2.0
1.2654.1 Available under license

1.2655 babel-plugin-transform-es2015-spread 6.22.0
1.2655.1 Available under license

1.2656 **orc-core 1.5.1**
1.2656.1 Available under license

1.2657 **joda-time 2.9.1**
1.2657.1 Available under license

1.2658 **is-number 4.0.0**
1.2658.1 Available under license

1.2659 **junit 4.10**
1.2659.1 Available under license

1.2660 **hadoop-mapreduce-client-jobclient 3.2.2**
1.2660.1 Available under license

1.2661 **strip-ansi 6.0.0 6.0.0**
1.2661.1 Available under license

1.2662 **mkdirp 0.3.0**
1.2662.1 Available under license

1.2663 **engine.io-parser 1.3.2 1.3.2**
1.2663.1 Available under license

1.2664 **es-to-primitive 1.2.1**
1.2664.1 Available under license

1.2665 **metrics-health-checks 4.1.0**

1.2666 **commons-math 2.2**
1.2666.1 Available under license

1.2667 **jopt-simple 5.0.4**
1.2667.1 Available under license

1.2668 **zero-allocation-hashing 0.9**
1.2668.1 Available under license

1.2669 **core-js 3.9.1**
1.2669.1 Available under license

1.2670 **netty-transport 4.1.42.Final**
1.2670.1 Available under license

1.2671 **braces 2.3.2**
1.2671.1 Available under license

1.2672 **is-fullwidth-code-point 3.0.0**
1.2672.1 Available under license

1.2673 **glob-base 0.3.0**
1.2673.1 Available under license

1.2674 **color-convert 1.9.3**
1.2674.1 Available under license

1.2675 **next-tick 1.0.0**
1.2675.1 Available under license
1.2676 traverse 0.3.9
1.2676.1 Available under license
1.2677 require-main-filename 2.0.0
1.2677.1 Available under license
1.2678 mixin-deep 1.3.2
1.2678.1 Available under license
1.2679 engine.io 1.8.5
1.2679.1 Available under license
1.2680 hadoop-mapreduce-client-common 3.2.2
1.2680.1 Available under license
1.2681 hide-powered-by 1.0.0
1.2681.1 Available under license
1.2682 mesa 18.0.5-0ubuntu0~16.04.1
1.2682.1 Available under license
1.2683 braces 1.8.5
1.2683.1 Available under license
1.2684 babel-plugin-transform-exponentiation-operator 6.24.1 6.24.1
1.2684.1 Available under license
1.2685 update-notifier 4.1.3
1.2685.1 Available under license
1.2686 opentracing-noop 0.31.0
1.2687 component-emitter 1.2.1
1.2687.1 Available under license
1.2688 caching-transform 4.0.0
1.2688.1 Available under license
1.2689 jaeger-core 1.5.0
1.2689.1 Available under license
1.2690 lodash-union 4.6.0
1.2690.1 Available under license
1.2691 parquet-encoding 1.12.0
1.2691.1 Available under license
1.2692 websocket-common 9.4.17.v20190418
1.2692.1 Available under license
1.2693 browser-stdout 1.3.1
1.2693.1 Available under license
1.2694 core-js 2.6.12 2.6.12
1.2694.1 Available under license
1.2695 find-up 2.1.0
1.2695.1 Available under license

1.2696 **chokidar** 3.5.1
   1.2696.1 Available under license

1.2697 **avro** 1.7.7
   1.2697.1 Available under license

1.2698 **wrap-ansi** 2.1.0
   1.2698.1 Available under license

1.2699 **annotations** 16.0.3
   1.2699.1 Available under license

1.2700 **tough-cookie** 2.5.0 2.5.0
   1.2700.1 Available under license

1.2701 **os-tmpdir** 1.0.2 1.0.2
   1.2701.1 Available under license

1.2702 **xerces-j** 2.9.1
   1.2702.1 Available under license

1.2703 **antlr-runtime** 4.9
   1.2703.1 Available under license

1.2704 **jsreport-sample-template** 2.6.1
   1.2704.1 Available under license

1.2705 **regenerate** 1.4.0 1.4.0
   1.2705.1 Available under license

1.2706 **json-schema-ref-parser** 6.1.0 6.1.0
   1.2706.1 Available under license

1.2707 **babel-plugin-transform-es2015-arrow-functions** 6.22.0
   1.2707.1 Available under license

1.2708 **parquet-avro** 1.12.0
   1.2708.1 Available under license

1.2709 **jetty-xml** 9.4.34.v20201102
   1.2709.1 Available under license

1.2710 **indexof** 0.0.1
   1.2710.1 Available under license

1.2711 **isorelax** 20090621
   1.2711.1 Available under license

1.2712 **metrics** 2.2.0
   1.2712.1 Available under license

1.2713 **jheaps** 0.9
   1.2713.1 Available under license

1.2714 **commons-text** 1.6
   1.2714.1 Available under license
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>drill-opentsdb-storage</td>
<td>1.19.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>metrics-core</td>
<td>2.2.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>de.huxhorn.sulky.codec</td>
<td>0.9.17</td>
<td>Available under license</td>
</tr>
<tr>
<td>libxcursor</td>
<td>1.1.14-1ubuntu0.16.04.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>escape-string-regexp</td>
<td>1.0.5</td>
<td>1.0.5 Available under license</td>
</tr>
<tr>
<td>sqlite-jdbc</td>
<td>3.30.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>prop-types</td>
<td>15.7.2</td>
<td>15.7.2 Available under license</td>
</tr>
<tr>
<td>es6-symbol</td>
<td>3.1.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>graphite</td>
<td>1.3.10-0ubuntu0.16.04.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>ant-launcher</td>
<td>1.10.9</td>
<td>Available under license</td>
</tr>
<tr>
<td>camelize</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>iee754</td>
<td>1.1.13</td>
<td>Available under license</td>
</tr>
<tr>
<td>asm</td>
<td>7.3.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>ws</td>
<td>6.2.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>base64-js</td>
<td>1.3.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>secure-keys</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>avro-ipc-jetty</td>
<td>1.9.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>side-channel</td>
<td>1.0.4</td>
<td>1.0.4 Available under license</td>
</tr>
<tr>
<td>concat-stream</td>
<td>2.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>commons-configuration</td>
<td>2.1.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.2734.1 Available under license
1.2735 metrics-jmx 4.0.2
1.2736 to-object-path 0.3.0
   1.2736.1 Available under license
1.2737 collection-visit 1.0.0
   1.2737.1 Available under license
1.2738 chainsaw 0.1.0
   1.2738.1 Available under license
1.2739 natty 0.13
   1.2739.1 Available under license
1.2740 is-fullwidth-code-point 1.0.0
   1.2740.1 Available under license
1.2741 kind-of 5.1.0
   1.2741.1 Available under license
1.2742 hbase-zookeeper 2.4.2
   1.2742.1 Available under license
1.2743 @babel/helpers 7.13.10
   1.2743.1 Available under license
1.2744 es6-error 4.1.1
   1.2744.1 Available under license
1.2745 is-posix-bracket 0.1.1
   1.2745.1 Available under license
1.2746 pump 3.0.0
   1.2746.1 Available under license
1.2747 mylib 0.0.0
   1.2747.1 Available under license
1.2748 babel-helper-function-name 6.24.1
   1.2748.1 Available under license
1.2749 jsreport-chrome-pdf 1.10.0
   1.2749.1 Available under license
1.2750 gson 2.8.2
1.2751 jetty-util 9.4.34.v20201102
   1.2751.1 Available under license
1.2752 archy 1.0.0
   1.2752.1 Available under license
1.2753 batik-rasterizer-ext 1.13
   1.2753.1 Available under license
1.2754 user-home 1.1.1
   1.2754.1 Available under license
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>platform</td>
<td>1.3.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>extend</td>
<td>3.0.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>jline-builtins</td>
<td>3.12.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>hive-standalone-metastore</td>
<td>3.1.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>avro-mapred</td>
<td>1.9.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>node-modules-regexp</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>jetty-http</td>
<td>9.4.34.v20201102</td>
<td>Available under license</td>
</tr>
<tr>
<td>batik-svgbrowser</td>
<td>1.13</td>
<td>Available under license</td>
</tr>
<tr>
<td>jsreport-freeze</td>
<td>2.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>methods</td>
<td>1.1.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>debug</td>
<td>2.2.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>socket.io-parser</td>
<td>2.3.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>maxmind-db</td>
<td>1.4.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>orc-shims</td>
<td>1.5.6</td>
<td>Available under license</td>
</tr>
<tr>
<td>body-parser</td>
<td>1.18.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>is-callable</td>
<td>1.1.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>mime</td>
<td>1.3.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>yargs</td>
<td>15.4.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>antlr</td>
<td>3.1.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>send</td>
<td>0.17.1</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.2774.1 Available under license
1.2775 jackson-databind 2.11.1
  1.2775.1 Available under license
1.2776 append-transform 2.0.0
  1.2776.1 Available under license
1.2777 hive-serde 3.1.2
  1.2777.1 Available under license
1.2778 xmlsec-java 2.2.1
  1.2778.1 Available under license
1.2779 classnames 2.2.6 2.2.6
  1.2779.1 Available under license
1.2780 jnr-ffi 2.1.5
1.2781 findup-sync 3.0.0
  1.2781.1 Available under license
1.2782 scala 2.11.0
  1.2782.1 Available under license
1.2783 j2objc-annotations 1.3
1.2784 icu 55.1 7ubuntu0.4
  1.2784.1 Available under license
1.2785 is-installed-globally 0.3.2
  1.2785.1 Available under license
1.2786 interpret 1.4.0
  1.2786.1 Available under license
1.2787 jersey 2.32
  1.2787.1 Available under license
1.2788 lucene-queryparser 7.4.0
  1.2788.1 Available under license
1.2789 http-errors 1.3.1
  1.2789.1 Available under license
1.2790 iconv-lite 0.4.19 0.4.19
  1.2790.1 Available under license
1.2791 accepts 1.3.5 1.3.5
  1.2791.1 Available under license
1.2792 regenerator-runtime 0.13.7
  1.2792.1 Available under license
1.2793 babel-plugin-transform-es2015-shorthand-properties 6.24.1
  1.2793.1 Available under license
1.2794 asn1 0.2.4
  1.2794.1 Available under license
1.2795 inherits 2.0.4 2.0.4
  1.2795.1 Available under license
1.2796 trim-right 1.0.1 1.0.1
  1.2796.1 Available under license
1.2797 twill-common 0.6.0-incubating
  1.2797.1 Available under license
1.2798 xmlbuilder 11.0.1 11.0.1
  1.2798.1 Available under license
1.2799 jsreport-debug 2.1.3
  1.2799.1 Available under license
1.2800 intl-format-cache 2.1.0 2.1.0
  1.2800.1 Available under license
1.2801 engine.io 3.4.2
  1.2801.1 Available under license
1.2802 async 2.6.0 2.6.0
  1.2802.1 Available under license
1.2803 got 2.4.0 2.4.0
  1.2803.1 Available under license
1.2804 @babel/plugin-syntax-logical-assignment-operators 7.10.4
  1.2804.1 Available under license
1.2805 my-scope-package-b 0.0.0
  1.2805.1 Available under license
1.2806 cliui 5.0.0
  1.2806.1 Available under license
1.2807 bouncy-castle 1.60
  1.2807.1 Available under license
1.2808 @babel/helper-get-function-arity 7.12.13
  1.2808.1 Available under license
1.2809 node-preload 0.2.1
  1.2809.1 Available under license
1.2810 shebang-regex 3.0.0
  1.2810.1 Available under license
1.2811 protostuff-json 1.7.1
  1.2811.1 Available under license
1.2812 oro 2.0.8
1.2813 jackson-annotations 2.10.5
  1.2813.1 Available under license
1.2814 aop-alliance 1.0
  1.2814.1 Available under license
<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>argparse</td>
<td>1.0.10</td>
<td>Available under license</td>
</tr>
<tr>
<td>hbase-procedure</td>
<td>2.4.2</td>
<td>Available under license</td>
</tr>
<tr>
<td>config</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>mime</td>
<td>1.3.4</td>
<td>Available under license</td>
</tr>
<tr>
<td>readdirp</td>
<td>3.5.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>netty-codec-socks</td>
<td>4.1.42.Final</td>
<td>Available under license</td>
</tr>
<tr>
<td>urijs</td>
<td>1.19.6</td>
<td>Available under license</td>
</tr>
<tr>
<td>electron-to-chromium</td>
<td>1.3.55</td>
<td>Available under license</td>
</tr>
<tr>
<td>htrace-core4</td>
<td>4.1.0-incubating</td>
<td>Available under license</td>
</tr>
<tr>
<td>avro-mapred</td>
<td>1.7.7</td>
<td>Available under license</td>
</tr>
<tr>
<td>kerb-admin</td>
<td>1.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>wrap-ansi</td>
<td>7.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>locate-path</td>
<td>3.0.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>@exodus/schemasafe</td>
<td>1.0.0-rc.3</td>
<td>Available under license</td>
</tr>
<tr>
<td>media-typer</td>
<td>0.3.0</td>
<td>Available under license</td>
</tr>
<tr>
<td>batik-codec</td>
<td>1.13</td>
<td>Available under license</td>
</tr>
<tr>
<td>nocache</td>
<td>1.0.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>compressible</td>
<td>2.0.14</td>
<td>Available under license</td>
</tr>
<tr>
<td>babel-helper-bindify-decorators</td>
<td>6.24.1</td>
<td>Available under license</td>
</tr>
<tr>
<td>http-cache-semantics</td>
<td>4.1.0</td>
<td>Available under license</td>
</tr>
</tbody>
</table>
1.2834.1 Available under license

1.2835 request-promise-core 1.1.4
1.2835.1 Available under license

1.2836 json-glib 1.1.2-0ubuntu1
1.2836.1 Available under license

1.2837 require-main-filename 2.0.0 2.0.0
1.2837.1 Available under license

1.2838 cliui 4.1.0
1.2838.1 Available under license

1.2839 resolve-from 2.0.0 2.0.0
1.2839.1 Available under license

1.2840 shelljs 0.8.4
1.2840.1 Available under license

1.2841 big.js 5.2.2
1.2841.1 Available under license

1.2842 serve-static 1.14.1
1.2842.1 Available under license

1.2843 exit-hook 1.1.1
1.2843.1 Available under license

1.2844 preserve 0.2.0
1.2844.1 Available under license

1.2845 zookeeper 3.5.7
1.2845.1 Available under license

1.2846 worker-loader 1.1.1
1.2846.1 Available under license

1.2847 netty-tcnative 2.0.1.Final
1.2847.1 Available under license

1.2848 is-obj 2.0.0
1.2848.1 Available under license

1.2849 curvesapi 1.06

1.2850 picomatch 2.2.2
1.2850.1 Available under license

1.2851 jersey-media-multipart 2.25.1
1.2851.1 Available under license

1.2852 regjsparser 0.1.5 0.1.5
1.2852.1 Available under license

1.2853 unicode-match-property-value-ecmascript 1.2.0
1.2853.1 Available under license

1.2854 zlib 1.2.8
1.2854.1 Available under license
1.2855 ajv 6.12.6
   1.2855.1 Available under license
1.2856 groovy-nio 3.0.8
   1.2856.1 Available under license
1.2857 isexe 2.0.0
   1.2857.1 Available under license
1.2858 chokidar 3.3.1
   1.2858.1 Available under license
1.2859 archiver 3.0.0
   1.2859.1 Available under license
1.2860 color-convert 0.5.3
   1.2860.1 Available under license
1.2861 swagger-converter 1.5.1 1.5.1
   1.2861.1 Available under license
1.2862 vm2 3.8.3
   1.2862.1 Available under license
1.2863 google-http-client-xml 1.20.0
   1.2863.1 Available under license
1.2864 boost 1.58.0+dfsg 5ubuntu3.1
   1.2864.1 Available under license
1.2865 readable-stream 2.2.7
   1.2865.1 Available under license
1.2866 hk2-api 2.2.0-b21
   1.2866.1 Available under license
1.2867 jsprim 1.4.1
   1.2867.1 Available under license
1.2868 node-ensure 0.0.0 0.0.0
   1.2868.1 Available under license
1.2869 aop-alliance 1.3.1.7-5.ph2
   1.2869.1 Available under license
1.2870 msexcel-builder-extended 0.0.8
   1.2870.1 Available under license
1.2871 babel-plugin-syntax-dynamic-import 6.18.0
   1.2871.1 Available under license
1.2872 audience-annotations 0.5.0
   1.2872.1 Available under license
1.2873 safe-json-stringify 1.2.0
   1.2873.1 Available under license
1.2874 ajv-keywords 3.5.2
   1.2874.1 Available under license
1.2875 @babel/helper-explode-assignable-expression 7.13.0
   1.2875.1 Available under license
1.2876 mongodb 2.2.36 2.2.36
   1.2876.1 Available under license
1.2877 guava 11.0.2
   1.2877.1 Available under license
1.2878 babel-plugin-transform-es2015-literals 6.22.0 6.22.0
   1.2878.1 Available under license
1.2879 mongodb-core 2.1.20
   1.2879.1 Available under license
1.2880 superagent 2.3.0
   1.2880.1 Available under license
1.2881 java-sizeof 0.0.5
   1.2881.1 Available under license
1.2882 okio 2.4.3
   1.2882.1 Available under license
1.2883 batik-parser 1.13
   1.2883.1 Available under license
1.2884 call-bind 1.0.2
   1.2884.1 Available under license
1.2885 nanomatch 1.2.13
   1.2885.1 Available under license
1.2886 istanbul-lib-source-maps 4.0.0
   1.2886.1 Available under license
1.2887 babel-template 6.26.0
   1.2887.1 Available under license
1.2888 socket.io-client 1.7.4 1.7.4
   1.2888.1 Available under license
1.2889 libflac 1.3.1
   1.2889.1 Available under license
1.2890 find-up 4.1.0 4.1.0
   1.2890.1 Available under license
1.2891 encodeurl 1.0.2
   1.2891.1 Available under license
1.2892 regexpu-core 2.0.0 2.0.0
   1.2892.1 Available under license
1.2893 is-plain-object 2.0.4
Open Source Used In DNAC Platform Antelope 148

1.2893.1 Available under license
1.2894 babel-plugin-syntax-exponentiation-operator 6.13.0 6.13.0
   1.2894.1 Available under license
1.2895 avatica-core 1.17.0
   1.2895.1 Available under license
1.2896 supports-color 2.0.0 2.0.0
   1.2896.1 Available under license
1.2897 babel-plugin-transform-class-properties 6.24.1
   1.2897.1 Available under license
1.2898 summernote 0.8.11 0.8.11
   1.2898.1 Available under license
1.2899 filename-regex 2.0.1
   1.2899.1 Available under license
1.2900 engine.io 1.8.5 1.8.5
   1.2900.1 Available under license
1.2901 @babel/runtime 7.13.10
   1.2901.1 Available under license
1.2902 jline 3.12.1
   1.2902.1 Available under license
1.2903 activation-api 1.1
   1.2903.1 Available under license
1.2904 blob 0.0.4
   1.2904.1 Available under license
1.2905 send 0.13.1
   1.2905.1 Available under license
1.2906 timed-out 2.0.0 2.0.0
   1.2906.1 Available under license
1.2907 jetty-servlet 9.4.41.v20210516
   1.2907.1 Available under license
1.2908 slash 3.0.0 3.0.0
   1.2908.1 Available under license
1.2909 bouncy-castle 1.64
   1.2909.1 Available under license
1.2910 babel-template 6.26.0 6.26.0
   1.2910.1 Available under license
1.2911 parse-duration 0.1.2
   1.2911.1 Available under license
1.2912 lodash-assignin 4.2.0
   1.2912.1 Available under license
1.2913 fast-deep-equal 1.1.0
   1.2913.1 Available under license
1.2914 buffer-fill 1.0.0
   1.2914.1 Available under license
1.2915 glew 1.3.1-1ubuntu0.16.04.2
   1.2915.1 Available under license
1.2916 jnr-netdb 1.1.6
1.2917 zip-stream 2.1.3
   1.2917.1 Available under license
1.2918 hadoop-yarn-common 3.2.2
   1.2918.1 Available under license
1.2919 aws4 1.10.1
   1.2919.1 Available under license
1.2920 delayed-stream 1.0.0 1.0.0
   1.2920.1 Available under license
1.2921 jsreport-text 2.0.0
   1.2921.1 Available under license
1.2922 apache-log4j 1.2.17
   1.2922.1 Available under license
1.2923 depd 1.1.2
   1.2923.1 Available under license
1.2924 jju 1.2.1 1.2.1
   1.2924.1 Available under license
1.2925 bluebird 3.5.1
   1.2925.1 Available under license
1.2926 extend-shallow 2.0.1
   1.2926.1 Available under license
1.2927 commons-http-client 3.1
   1.2927.1 Available under license
1.2928 safer-buffer 2.1.2 2.1.2
   1.2928.1 Available under license
1.2929 lodash-bind 4.2.1
   1.2929.1 Available under license
1.2930 nimbus-jose-jwt 4.41.3
   1.2930.1 Available under license
1.2931 hbase-server 2.4.2
   1.2931.1 Available under license
1.2932 propagate 2.0.1
   1.2932.1 Available under license
1.2933 jackson-xc 2.12.1  
   1.2933.1 Available under license
1.2934 p-try 2.2.0  
   1.2934.1 Available under license
1.2935 raml-parser 0.8.18 0.8.18  
   1.2935.1 Available under license
1.2936 jersey-common 2.25.1  
   1.2936.1 Available under license
1.2937 nconf 0.10.0  
   1.2937.1 Available under license
1.2938 avahi 0.6.32~rc+dfsg-1ubuntu2.3  
   1.2938.1 Available under license
1.2939 @babel/plugin-transform-literals 7.12.13  
   1.2939.1 Available under license
1.2940 jedis 2.10.2  
1.2941 node-releases 1.1.71  
   1.2941.1 Available under license
1.2942 object.assign 4.1.2  
   1.2942.1 Available under license
1.2943 ms 2.0.0  
   1.2943.1 Available under license
1.2944 javassist 3.18.2-GA  
   1.2944.1 Available under license
1.2945 buffer-crc32 0.2.13  
   1.2945.1 Available under license
1.2946 has-symbols 1.0.1  
   1.2946.1 Available under license
1.2947 fs 0.0.2  
   1.2947.1 Available under license
1.2948 de.huxhorn.lilith.data.logging 0.9.44  
   1.2948.1 Available under license
1.2949 mongodb-driver-core 3.12.4  
   1.2949.1 Available under license
1.2950 json5 1.0.1  
   1.2950.1 Available under license
1.2951 buffer-equal-constant-time 1.0.1  
   1.2951.1 Available under license
1.2952 json-schema-compatibility 1.1.0  
   1.2952.1 Available under license
1.2953 compression 1.6.2 1.6.2
   1.2953.1 Available under license
1.2954 qt 5.7.0
   1.2954.1 Available under license
1.2955 crc 3.8.0
   1.2955.1 Available under license
1.2956 https-proxy-agent 3.0.1
   1.2956.1 Available under license
1.2957 netty 4.1.61.Final
   1.2957.1 Available under license
1.2958 external-editor 3.1.0
   1.2958.1 Available under license
1.2959 nanoid 1.0.7
   1.2959.1 Available under license
1.2960 deep-is 0.1.3 0.1.3
   1.2960.1 Available under license
1.2961 fill-range 4.0.0
   1.2961.1 Available under license
1.2962 netty-handler 4.1.59.Final
   1.2962.1 Available under license
1.2963 jansi-native 1.8
   1.2963.1 Available under license
1.2964 jbig-kit 2.1-3.1
   1.2964.1 Available under license
1.2965 jcip-annotation 1.0.1
   1.2965.1 Available under license
1.2966 supports-color 2.0.0
   1.2966.1 Available under license
1.2967 find-cache-dir 2.1.0
   1.2967.1 Available under license
1.2968 read-pkg 2.0.0
   1.2968.1 Available under license
1.2969 word-wrap 1.2.3 1.2.3
   1.2969.1 Available under license
1.2970 etag 1.8.1 1.8.1
   1.2970.1 Available under license
1.2971 term-size 2.2.1
   1.2971.1 Available under license
1.2972 ajv-keywords 3.4.1 3.4.1
1.2972.1 Available under license

1.2973 jsesc 0.5.0
   1.2973.1 Available under license

1.2974 zlib 1.2.3
   1.2974.1 Available under license

1.2975 isobject 3.0.1
   1.2975.1 Available under license

1.2976 prelude-is 1.1.2 1.1.2
   1.2976.1 Available under license

1.2977 calcite-linq4j 1.21.0-drill-r2
   1.2977.1 Available under license

1.2978 ion-java 1.0.2
   1.2978.1 Available under license

1.2979 is 3.3.0
   1.2979.1 Available under license

1.2980 @babel/plugin-transform-duplicate-keys 7.12.13
   1.2980.1 Available under license

1.2981 set-blocking 2.0.0 2.0.0
   1.2981.1 Available under license

1.2982 get-value 2.0.6
   1.2982.1 Available under license

1.2983 platform 1.3.1
   1.2983.1 Available under license

1.2984 passport-local 1.0.0
   1.2984.1 Available under license

1.2985 nodemon 2.0.7
   1.2985.1 Available under license

1.2986 avatica-core 1.15.0
   1.2986.1 Available under license

1.2987 protostuff 1.7.1
   1.2987.1 Available under license

1.2988 cookiejar 2.0.6
   1.2988.1 Available under license

1.2989 jsreport-studio 2.10.1
   1.2989.1 Available under license

1.2990 jersey-client 2.32
   1.2990.1 Available under license

1.2991 archiver-utils 2.1.0
   1.2991.1 Available under license
1.2992 parquet-encoding 1.8.1
   1.2992.1 Available under license
1.2993 parseuri 0.0.6
   1.2993.1 Available under license
1.2994 drafter.js 2.6.7 2.6.7
   1.2994.1 Available under license
1.2995 aws-java-sdk-kms 1.11.704
   1.2995.1 Available under license
1.2996 isarray 1.0.0 1.0.0
   1.2996.1 Available under license
1.2997 jackson-dataformat-xml 2.10.4
   1.2997.1 Available under license
1.2998 private 0.1.8
   1.2998.1 Available under license
   1.2999.1 Available under license
1.3000 fstream 1.0.12
   1.3000.1 Available under license
1.3001 blob 0.0.5
   1.3001.1 Available under license
   1.3002.1 Available under license
1.3003 define-property 2.0.2
   1.3003.1 Available under license
1.3004 de.huxhorn.lilith.sender 0.9.44
   1.3004.1 Available under license
1.3005 @babel/helper-module-transforms 7.13.14
   1.3005.1 Available under license
1.3006 netty-codec 4.1.59.Final
   1.3006.1 Available under license
1.3007 debug 2.6.9
   1.3007.1 Available under license
1.3008 es6-set 0.1.5
   1.3008.1 Available under license
1.3009 hk2-api 2.6.1
   1.3009.1 Available under license
1.3010 lodash.isplainobject 4.0.6
   1.3010.1 Available under license
1.3011 function-bind 1.1.1 1.1.1
1.3011.1 Available under license
1.3012 zookeeper-jute 3.5.7
  1.3012.1 Available under license
1.3013 jnr-x86asm 1.0.2
  1.3013.1 Available under license
1.3014 dicer 0.2.5
  1.3014.1 Available under license
1.3015 jetty-security 9.4.17.v20190418
  1.3015.1 Available under license
1.3016 msgpack 0.6.6
  1.3016.1 Available under license
1.3017 bytes 3.1.0
  1.3017.1 Available under license
1.3018 buffer-shims 1.0.0 1.0.0
  1.3018.1 Available under license
1.3019 typedarray 0.0.6 0.0.6
  1.3019.1 Available under license
1.3020 better-assert 1.0.2 1.0.2
  1.3020.1 Available under license
1.3021 paranamer 2.3
  1.3021.1 Available under license
1.3022 java-jwt 3.8.3
  1.3022.1 Available under license
1.3023 jetty 9.4.17.v20190418
  1.3023.1 Available under license
1.3024 grpc-protobuf-lite 1.27.1
  1.3024.1 Available under license
1.3025 source-map 0.5.7 0.5.7
  1.3025.1 Available under license
1.3026 detect-file 1.0.0
  1.3026.1 Available under license
1.3027 jsreport-jsrender 2.1.1
  1.3027.1 Available under license
1.3028 fast-deep-equal 1.1.0 1.1.0
  1.3028.1 Available under license
1.3029 pixman 0.33.6 1
  1.3029.1 Available under license
1.3030 follow-redirects 1.5.10
  1.3030.1 Available under license
1.3031 jwa 1.1.6 1.1.6
   1.3031.1 Available under license

1.3032 esutils 2.0.2 2.0.2
   1.3032.1 Available under license

1.3033 agent-base 4.3.0
   1.3033.1 Available under license

1.3034 unc-path-regex 0.1.2
   1.3034.1 Available under license

1.3035 json5 0.5.1
   1.3035.1 Available under license

1.3036 jsreport-scripts 2.6.0
   1.3036.1 Available under license

1.3037 client-sessions 0.8.0
   1.3037.1 Available under license

1.3038 babel-plugin-transform-es2015-block-scoping 6.26.0
   1.3038.1 Available under license

1.3039 axios 0.21.4 0.21.4
   1.3039.1 Available under license

1.3040 jackson-jaxrs 2.10.4
   1.3040.1 Available under license

1.3041 indent-string 4.0.0
   1.3041.1 Available under license

1.3042 home-or-tmp 2.0.0
   1.3042.1 Available under license

1.3043 babel-plugin-transform-property-literals 7.12.13
   1.3043.1 Available under license

1.3044 js-yaml 3.13.1
   1.3044.1 Available under license

1.3045 yargs-parser 13.1.2
   1.3045.1 Available under license

1.3046 lodash.isboolean 3.0.3
   1.3046.1 Available under license

1.3047 hive-spark-client 3.1.2
   1.3047.1 Available under license

1.3048 statuses 1.5.0
   1.3048.1 Available under license

1.3049 jackson-dataformat-hocon 1.1.1
   1.3049.1 Available under license

1.3050 base64id 1.0.0
1.3050.1 Available under license

1.3051 **utf-8-validate** 1.2.1 1.2.1
1.3051.1 Available under license

1.3052 **jansi** 1.8
1.3052.1 Available under license

1.3053 **free-type** 2.6.1 0.1ubuntu2.4
1.3053.1 Available under license

1.3054 **babel-plugin-syntax-decorators** 6.13.0
1.3054.1 Available under license

1.3055 **file-entry-cache** 2.0.0
1.3055.1 Available under license

1.3056 **jackson** 2.4.0
1.3056.1 Available under license

1.3057 **react-dom** 16.4.1 16.4.1
1.3057.1 Available under license

1.3058 **jffi** 1.0.1
1.3058.1 Available under license

1.3059 **commons-beanutils** 1.8.0
1.3059.1 Available under license

1.3060 **babel-plugin-transform-runtime** 6.23.0
1.3060.1 Available under license

1.3061 **ws** 1.1.1
1.3061.1 Available under license

1.3062 **babel-plugin-transform-es2015-typeof-symbol** 6.23.0
1.3062.1 Available under license

1.3063 **babel-helper-create-regexp-features-plugin** 7.12.17
1.3063.1 Available under license

1.3064 **filter-obj** 1.1.0 1.1.0
1.3064.1 Available under license

1.3065 **supports-color** 6.0.0
1.3065.1 Available under license

1.3066 **uri-js** 4.4.1 4.4.1
1.3066.1 Available under license

1.3067 **mongo-java-driver** 3.12.4
1.3067.1 Available under license

1.3068 **is-absolute** 1.0.0
1.3068.1 Available under license

1.3069 **xmlgraphics-commons** 2.4
1.3069.1 Available under license
1.3070 har-validator 5.0.3 5.0.3
   1.3070.1 Available under license
1.3071 jetty-continuation 9.4.41.v20210516
   1.3071.1 Available under license
1.3072 resolve-dir 1.0.1
   1.3072.1 Available under license
1.3073 json-schema 0.2.3 0.2.3
   1.3073.1 Available under license
1.3074 commons-lang3 3.10
   1.3074.1 Available under license
1.3075 ansi-regex 5.0.0
   1.3075.1 Available under license
1.3076 tar-stream 1.6.2
   1.3076.1 Available under license
1.3077 pstree-remy 1.1.8
   1.3077.1 Available under license
1.3078 shelljs 0.7.8
   1.3078.1 Available under license
1.3079 argparse 1.0.10
   1.3079.1 Available under license
1.3080 winston-transport 3.0.1
   1.3080.1 Available under license
1.3081 jackson-jaxrs 2.12.1
   1.3081.1 Available under license
1.3082 netty 4.1.59.Final
   1.3082.1 Available under license
1.3083 codemodel 2.6
1.3084 hoist-non-react-statics 2.5.5 2.5.5
   1.3084.1 Available under license
1.3085 os-homedir 1.0.2
   1.3085.1 Available under license
1.3086 babylon 6.18.0 6.18.0
   1.3086.1 Available under license
1.3087 babel-preset-env 1.7.0
   1.3087.1 Available under license
1.3088 tiny-inflate 1.0.3
   1.3088.1 Available under license
1.3089 os-tmpdir 1.0.2
   1.3089.1 Available under license
1.3090 commons-configuration 1.6
   1.3090.1 Available under license

1.3091 tsconfig-paths 3.9.0
   1.3091.1 Available under license

1.3092 highlight-words-core 1.2.2 1.2.2
   1.3092.1 Available under license

1.3093 number-is-nan 1.0.1 1.0.1
   1.3093.1 Available under license

1.3094 @babel/plugin-transform-typeof-symbol 7.12.13
   1.3094.1 Available under license

1.3095 cookie 0.3.1 0.3.1
   1.3095.1 Available under license

1.3096 curator-client 5.1.0
   1.3096.1 Available under license

1.3097 psl 1.8.0
   1.3097.1 Available under license

1.3098 base64-arraybuffer 0.1.5
   1.3098.1 Available under license

1.3099 content-type 1.0.4 1.0.4
   1.3099.1 Available under license

1.3100 lodash 4.17.20
   1.3100.1 Available under license

1.3101 kafka-metadata 2.8.0
   1.3101.1 Available under license

1.3102 icu 44
   1.3102.1 Available under license

1.3103 http-signature 1.2.0 1.2.0
   1.3103.1 Available under license

1.3104 commons-logging 1.2
   1.3104.1 Available under license

1.3105 libproxy 0.4.11-5ubuntu1
   1.3105.1 Available under license

1.3106 getpass 0.1.7
   1.3106.1 Available under license

1.3107 batik-svg-dom 1.13
   1.3107.1 Available under license

1.3108 drill-jdbc 1.19.0
   1.3108.1 Available under license

1.3109 qs 6.5.2
1.3109.1 Available under license

1.3110 worker-loader 1.1.1 1.1.1
1.3110.1 Available under license

1.3111 h2 1.4.200
1.3111.1 Available under license

1.3112 async 1.0.0
1.3112.1 Available under license

1.3113 velocity-engine-core 2.0
1.3113.1 Available under license

1.3114 jsreport-licensing 2.2.3
1.3114.1 Available under license

1.3115 react-is 16.13.1
1.3115.1 Available under license

1.3116 range-parser 1.0.3
1.3116.1 Available under license

1.3117 socket.io-parser 3.4.1
1.3117.1 Available under license

1.3118 xerces-j 2.12.0
1.3118.1 Available under license

1.3119 snake-yaml 1.29
1.3119.1 Available under license

1.3120 cglib 2.2.2
1.3120.1 Available under license

1.3121 drill-common 1.19.0
1.3121.1 Available under license

1.3122 async-limiter 1.0.1
1.3122.1 Available under license

1.3123 jackson-dataformat-yaml 2.9.8
1.3123.1 Available under license

1.3124 jetty-client 9.3.27.v20190418
1.3125 any-promise 1.3.0
1.3125.1 Available under license

1.3126 get-caller-file 2.0.5 2.0.5
1.3126.1 Available under license

1.3127 drill-format-syslog 1.19.0
1.3127.1 Available under license

1.3128 buffer-alloc 1.2.0
1.3128.1 Available under license

1.3129 httpcore5 5.0.2
1.3129.1 Available under license

1.3130 babel-plugin-transform-regenerator 6.26.0
   1.3130.1 Available under license

1.3131 ws 7.3.1
   1.3131.1 Available under license

1.3132 netty 4.1.42.Final
   1.3132.1 Available under license

1.3133 to-fast-properties 1.0.3 1.0.3
   1.3133.1 Available under license

1.3134 which-module 2.0.0 2.0.0
   1.3134.1 Available under license

1.3135 bean-validation-api 1.1.0.Final

1.3136 to-array 0.1.4
   1.3136.1 Available under license

1.3137 tinycolor2 1.4.1
   1.3137.1 Available under license

1.3138 bunyan 1.8.1 1.8.1
   1.3138.1 Available under license

1.3139 strict-uri-encode 2.0.0 2.0.0
   1.3139.1 Available under license

1.3140 wordwrap 1.0.0
   1.3140.1 Available under license

1.3141 javax-jdo 3.2.0-m3
   1.3141.1 Available under license

1.3142 janino 3.0.11

1.3143 @babel/plugin-transform-modules-amd 7.13.0
   1.3143.1 Available under license

1.3144 metrics 0.1.21
   1.3144.1 Available under license

1.3145 buffer-shims 1.0.0
   1.3145.1 Available under license

1.3146 ansi-styles 2.2.1
   1.3146.1 Available under license

1.3147 read-pkg-up 2.0.0
   1.3147.1 Available under license

1.3148 libvpx 1.8.1-299-gd2a5e26359
   1.3148.1 Available under license

1.3149 json5 2.2.0
   1.3149.1 Available under license
1.3150 strip-ansi 4.0.0
    1.3150.1 Available under license
1.3151 @babel/template 7.12.13
    1.3151.1 Available under license
1.3152 through 2.3.8
    1.3152.1 Available under license
1.3153 namespace 1.4.01
    1.3153.1 Available under license
1.3154 netty-resolver 4.1.59.Final
    1.3154.1 Available under license
1.3155 commons-pool 2.6.1
1.3156 okio 2.7.0
    1.3156.1 Available under license
1.3157 woodstox-core 6.2.0
    1.3157.1 Available under license
1.3158 bindings 1.2.1 1.2.1
    1.3158.1 Available under license
1.3159 @babel/code-frame 7.0.0
    1.3159.1 Available under license
1.3160 jaeger-thrift 0.23.0
    1.3160.1 Available under license
1.3161 jackson 2.11.1
    1.3161.1 Available under license
1.3162 minimatch 3.0.4
    1.3162.1 Available under license
1.3163 p-defer 1.0.0
    1.3163.1 Available under license
1.3164 path-to-regexp 0.1.7
    1.3164.1 Available under license
1.3165 base64id 1.0.0 1.0.0
    1.3165.1 Available under license
1.3166 string-prototype-codepointat 0.2.1
    1.3166.1 Available under license
1.3167 class-utils 0.3.6
    1.3167.1 Available under license
1.3168 hbase-mapreduce 2.0.0-alpha4
    1.3168.1 Available under license
1.3169 @babel/plugin-transform-shorthand-properties 7.12.13
    1.3169.1 Available under license
1.3170 string_decoder 0.10.31
    1.3170.1 Available under license
1.3171 htmlparser2 3.10.1
    1.3171.1 Available under license
1.3172 union-value 1.0.1
    1.3172.1 Available under license
1.3173 opentracing-api 0.31.0
    1.3173.1 Available under license
1.3174 generate-object-property 1.2.0
    1.3174.1 Available under license
1.3175 json-smart 2.3
    1.3175.1 Available under license
1.3176 kotlin 1.4.10
    1.3176.1 Available under license
1.3177 xdg-basedir 4.0.0
    1.3177.1 Available under license
1.3178 libpng 1.6.37
    1.3178.1 Available under license
1.3179 jsonpointer 4.1.0
    1.3179.1 Available under license
1.3180 babel-plugin-transform-es2015-template-literals 6.22.0 6.22.0
    1.3180.1 Available under license
1.3181 fined 1.2.0
    1.3181.1 Available under license
1.3182 http-signature 1.2.0
    1.3182.1 Available under license
1.3183 is-negative-zero 2.0.0
    1.3183.1 Available under license
1.3184 protobuf-java 3.13.0
    1.3184.1 Available under license
1.3185 python-setuptools 28.8.0
    1.3185.1 Available under license
1.3186 graceful-fs 4.1.11 4.1.11
    1.3186.1 Available under license
1.3187 is-date-object 1.0.2
    1.3187.1 Available under license
1.3188 make-dir 2.1.0
    1.3188.1 Available under license
1.3189 async 0.9.0
1.3189.1 Available under license
1.3190 babel-node 7.13.13
1.3190.1 Available under license
1.3191 commander 4.1.1 4.1.1
1.3191.1 Available under license
1.3192 groovy-test 3.0.8
1.3192.1 Available under license
1.3193 pkg-dir 2.0.0
1.3193.1 Available under license
1.3194 ajv 6.12.6 6.12.6
1.3194.1 Available under license
1.3195 joi 6.10.1 6.10.1
1.3195.1 Available under license
1.3196 domhandler 2.4.2 2.4.2
1.3196.1 Available under license
1.3197 bufferutil 1.2.1 1.2.1
1.3197.1 Available under license
1.3198 merge-descriptors 1.0.1
1.3198.1 Available under license
1.3199 yauzl 2.10.0
1.3199.1 Available under license
1.3200 stack-trace 0.0.10
1.3200.1 Available under license
1.3201 js-cookie 2.1.1 2.1.1
1.3201.1 Available under license
1.3202 jetty-servlet 9.3.27.v20190418
1.3203 jws 3.2.2
1.3203.1 Available under license
1.3204 http-errors 1.7.2
1.3204.1 Available under license
1.3205 memory-pager 1.5.0 1.5.0
1.3205.1 Available under license
1.3206 babel-register 7.13.14
1.3206.1 Available under license
1.3207 lodash.assign 4.2.0
1.3207.1 Available under license
1.3208 babel-runtime 6.26.0
1.3208.1 Available under license
1.3209 @babel/plugin-proposal-object-rest-spread 7.13.8

Open Source Used In DNAC Platform Antelope 163
1.3209.1 Available under license

1.3210 infinity-agent 1.0.2 1.0.2
1.3210.1 Available under license

1.3211 content-security-policy-builder 1.0.0 1.0.0
1.3211.1 Available under license

1.3212 bl 1.2.3
1.3212.1 Available under license

1.3213 widest-line 3.1.0
1.3213.1 Available under license

1.3214 type 1.2.0
1.3214.1 Available under license

1.3215 require_optional 1.0.1 1.0.1
1.3215.1 Available under license

1.3216 mongodb-migrations 1.0.1 1.0.1
1.3216.1 Available under license

1.3217 slf4j-log4j 2.8.2
1.3217.1 Available under license

1.3218 regex-not 1.0.2
1.3218.1 Available under license

1.3219 pako 0.2.9
1.3219.1 Available under license

1.3220 path-is-inside 1.0.2
1.3220.1 Available under license

1.3221 domutils 1.5.1
1.3221.1 Available under license

1.3222 dom 1.0
1.3222.1 Available under license

1.3223 commons-digester 1.8.1
1.3223.1 Available under license

1.3224 grpc-context 1.27.1
1.3224.1 Available under license

1.3225 better-assert 1.0.2
1.3225.1 Available under license

1.3226 jackson-annotations 2.4.0
1.3226.1 Available under license

1.3227 bytes 3.0.0 3.0.0
1.3227.1 Available under license

1.3228 junit-jupiter-engine 5.7.0
1.3228.1 Available under license
1.3229 decamelize 2.0.0
   1.3229.1 Available under license
1.3230 compressible 2.0.18
   1.3230.1 Available under license
1.3231 socket.io-adapter 0.5.0 0.5.0
   1.3231.1 Available under license
1.3232 isobject 3.0.1 3.0.1
   1.3232.1 Available under license
1.3233 parse-passwd 1.0.0
   1.3233.1 Available under license
1.3234 socket.io-parser 3.3.1
   1.3234.1 Available under license
1.3235 gson 2.2.4
1.3236 minimist 1.2.5 1.2.5
   1.3236.1 Available under license
1.3237 guice 4.0
   1.3237.1 Available under license
1.3238 drill-java-exec 1.19.0
   1.3238.1 Available under license
1.3239 pcre 2.48.2-0ubuntu4.4
   1.3239.1 Available under license
1.3240 @babel/helper-replace-supers 7.13.12
   1.3240.1 Available under license

1.1 bindings 1.2.1
1.1.1 Available under license:

No license file was found, but licenses were detected in source scan.

node-bindings
============

### Helper module for loading your native module's .node file

This is a helper module for authors of Node.js native addon modules. It is basically the "swiss army knife" of requiring your native module's .node file.

Throughout the course of Node's native addon history, addons have ended up being compiled in a variety of different places, depending on which build tool and which version of node was used. To make matters worse, now the _gyp_ build tool can produce either a _Release_ or _Debug_ build, each being built into different locations.
This module checks _all_ the possible locations that a native addon would be built at, and returns the first one that loads successfully.

Installation
------------

Install with `npm`

```bash
$ npm install bindings
```

Or add it to the "dependencies" section of your `_package.json_` file.

Example
-------

`require()`ing the proper bindings file for the current node version, platform and architecture is as simple as:

```js
var bindings = require('bindings')('binding.node')

// Use your bindings defined in your C files
bindings.your_c_function()
```

Nice Error Output
-----------------

When the `.node` file could not be loaded, `node-bindings` throws an Error with a nice error message telling you exactly what was tried. You can also check the `err.tries` Array property.

```
Error: Could not load the bindings file. Tried:
/Users/nrajlich/ref/build/binding.node
/Users/nrajlich/ref/build/Debug/binding.node
/Users/nrajlich/ref/build/Release/binding.node
/Users/nrajlich/ref/out/Debug/binding.node
/Users/nrajlich/ref/Debug/binding.node
/Users/nrajlich/ref/out/Release/binding.node
/Users/nrajlich/ref/Release/binding.node
/Users/nrajlich/ref/build/default/binding.node
/Users/nrajlich/ref/compiled/0.8.2/darwin/x64/binding.node
```
License
-------

(The MIT License)

Copyright (c) 2012 Nathan Rajlich &lt;nathan@tootallnate.net&gt;

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):
* /opt/cola/permits/1112921116_1607362434.89/0/bindings-1-2-1-2-tgz/package/README.md

1.2 babel-plugin-transform-decorators 6.24.1
1.2.1 Available under license :

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

MIT License

Copyright (c) 2014-2017 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3 babel-plugin-transform-es2015-modules-commonjs 6.26.2

1.3.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.4 babel-plugin-check-es2015-constants
6.22.0 6.22.0

1.4.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.5 httpcore-nio 4.4.6

1.5.1 Available under license:

Apache HttpCore NIO
Copyright 2005-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
Come back to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work.
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.6 @babel/helper-split-export-declaration

7.11.0

1.6.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.7 iconv-lite 0.4.24

1.7.1 Available under license:
Copyright (c) 2011 Alexander Shtuchkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
1.8 babel-plugin-transform-es2015-template-literals 6.22.0

1.8.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.9 netty 4.1.53.Final

1.9.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works.
“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
   1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
   2. You must cause any modified files to carry prominent notices stating that You changed the files; and
   3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
   4. If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any
additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.10 @babel/plugin-transform-modules-umd

7.13.0

1.10.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.11 es6-promise 3.2.1 3.2.1
1.11.1 Available under license :
Copyright (c) 2014 Yehuda Katz, Tom Dale, Stefan Penner and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.12 get-package-type 0.1.0
1.12.1 Available under license :
MIT License

Copyright (c) 2020 CF Ware, LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.13 babel-plugin-transform-strict-mode

6.24.1

1.13.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>

Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.14 pend 1.2.0
1.14.1 Available under license :
   The MIT License (Expat)

Copyright (c) 2014 Andrew Kelley

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation files
(the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge,
publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
1.15 parsejson 0.0.3
1.15.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.16 dtrace-provider 0.6.0 0.6.0
1.16.1 Available under license:
Copyright 2011 Chris Andrews. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY CHRIS ANDREWS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CHRIS ANDREWS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2012 Chris Andrews. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY CHRIS ANDREWS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CHRIS ANDREWS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.17 common-prefix 1.1.0 1.1.0

1.17.1 Available under license:

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.18 cross-spawn 6.0.5

1.18.1 Available under license:
   The MIT License (MIT)

   Copyright (c) 2018 Made With MOXY Lda <hello@moxy.studio>

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.19 cache-base 1.0.1

1.19.1 Available under license:
   The MIT License (MIT)

   Copyright (c) 2014-2017, Jon Schlinkert.

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.20 repeating 2.0.1 2.0.1

1.20.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.21 javax-el 3.0.0

1.21.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.
1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   
   B. Any new file that contains any part of the Original Software or previous Modification; or
   
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License.
You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item.. as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction’s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys’ fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.
The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on
the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives
a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE
PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW, EXCEPT WHEN OTHERWISE
STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM
"AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTicular PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY
COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE
PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL,
SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY
TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER
PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to
achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to
most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a
pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public
License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later
version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the
Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.22 assert-plus 1.0.0
1.22.1 Available under license:
No license file was found, but licenses were detected in source scan.

// Copyright (c) 2012, Mark Cavage. All rights reserved.

Found in path(s):
* /opt/cola/permits/1110631664_1607035113.64/0/assert-plus-1-0-0-4-tgz/package/assert.js
No license file was found, but licenses were detected in source scan.

The MIT License (MIT)
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
subject to the following conditions:
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1110631664_1607035113.64/0/assert-plus-1-0-0-4-tgz/package/README.md

1.23 netty-buffer 4.1.59.Final
1.23.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/UnsafeHeapSwappedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/UnsafeDirectSwappedByteBuf.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/AbstractPooledDerivedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/SimpleLeakAwareCompositeByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/UnpooledUnsafeNoCleanerDirectByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/PooledDuplicatedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/WrappedUnpooledUnsafeDirectByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/PooledSlicedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/AdvancedLeakAwareCompositeByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/WrappedUnpooledUnsafeDirectByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/PooledSlicedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/AdvancedLeakAwareCompositeByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/WrappedCompositeByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/AbstractUnpooledSlicedByteBuf.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE
~
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/META-INF/maven/io.netty/netty-buffer/pom.xml
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/CompositeByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/PooledByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/ByteBufUtil.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/SlicedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/UnpooledUnsafeDirectByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/ByteBufAllocator.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/package-info.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/AbstractByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/UnpooledDirectByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/Unpooled.java

Open Source Used In DNAC Platform Antelope 208
*/

* Copyright 2020 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
Open Source Used In DNAC Platform Antelope

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/search/package-info.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/LongPriorityQueue.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/SizeClassesMetric.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/SizeClasses.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/LongLongHashMap.java

No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/META-INF/native-image/io.netty/buffer/native-image.properties

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/PoolChunkListMetric.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/UnpooledDuplicatedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1.jar/io/netty/buffer/PooledUnsafeHeapByteBuf.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2020 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/search/AbstractSearchProcessorFactory.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/search/BitapSearchProcessorFactory.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/search/AhoCorasicSearchProcessorFactory.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/search/MultiSearchProcessor.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/search/MultiSearchProcessorFactory.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
  jar/io/netty/buffer/search/KmpSearchProcessorFactory.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/search/SearchProcessorFactory.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/search/AbstractMultiSearchProcessorFactory.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/search/SearchProcessor.java

* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/ReadOnlyUnsafeDirectByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/AdvancedLeakAwareByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/ByteBufProcessor.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/ReadOnlyByteBufferBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/UnreleasableByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/AbstractDerivedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/SimpleLeakAwareByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/EmptyByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/WrappedByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/FixedCompositeByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-
jar/io/netty/buffer/DefaultByteBufHolder.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/PooledUnsafeDirectByteBuf.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/AbstractReferenceCountedByteBuf.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
* 
* The Netty Project licenses this file to the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/PooledHeapByteBuf.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/PooledByteBufAllocatorMetric.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/ByteBufAllocatorMetric.java
* /opt/cola/permits/1145449558_1616156160.84/0/netty-buffer-4-1-59-final-sources-1-jar/io/netty/buffer/ByteBufAllocatorMetricProvider.java
1.24 babel-preset-stage-3 6.24.1

1.24.1 Available under license:

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

MIT License

Copyright (c) 2014-2017 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.25 cli-cursor 2.1.0

1.25.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.26 cookiejar 2.1.2 2.1.2

1.26.1 Available under license:

The MIT License (MIT)
Copyright (c) 2013 Bradley Meck

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.27 jetty-continuation 9.4.17.v20190418

1.27.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program. Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.
4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the
This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message
The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----
Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----
OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----
Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.28 netty-tcnative-boringssl-static
2.0.26.Final
1.28.1 Available under license:

No license file was found, but licenses were detected in source scan.

Bundle-License: http://www.apache.org/licenses/LICENSE-2.0

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/META-INF/MANIFEST.MF

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tnative/SSLPrivateKeyMethodSignTask.java
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tnative/SSLPrivateKeyMethodDecryptTask.java
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tnative/SSLPrivateKeyMethodTask.java
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tnative/SSLTask.java
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tnative/CertificateCallbackTask.java
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tnative/CertificateVerifierTask.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2.0-26-final-
sources-1-jar/io/netty/internal/tnative/CertificateCallback.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
Found in path(s):
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tcnative/SSLContext.java
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tcnative/Buffer.java
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tcnative/Library.java
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tcnative/SSL.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2014 The Netty Project
* *
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1068793931_1594430415.34/0/netty-tcnative-boringssl-static-2-0-26-final-sources-1-jar/io/netty/internal/tcnative/CertificateVerifier.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2016 The Netty Project
* *
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

1.29 inherits 2.0.3 2.0.3
1.29.1 Available under license :

The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.
1.30 commons-io 2.7
1.30.1 Available under license:

Apache Commons IO
Copyright 2002-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.31 ant-junit 1.10.9

1.31.1 Available under license:

    /*
     *    Apache License
     *      Version 2.0, January 2004
     *      http://www.apache.org/licenses/
     */

    TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

    1. Definitions.

    "License" shall mean the terms and conditions for use, reproduction,
    and distribution as defined by Sections 1 through 9 of this document.

    "Licensor" shall mean the copyright owner or entity authorized by
    the copyright owner that is granting the License.

    "Legal Entity" shall mean the union of the acting entity and all
    other entities that control, are controlled by, or are under common
    control with that entity. For the purposes of this definition,
    "control" means (i) the power, direct or indirect, to cause the
    direction or management of such entity, whether by contract or
    otherwise, or (ii) ownership of fifty percent (50%) or more of the
    outstanding shares, or (iii) beneficial ownership of such entity.

---

Open Source Used In DNAC Platform Antelope 240
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation, source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
* and charge a fee for, acceptance of support, warranty, indemnity,
* or other liability obligations and/or rights consistent with this
* License. However, in accepting such obligations, You may act only
* on Your own behalf and on Your sole responsibility, not on behalf
* of any other Contributor, and only if You agree to indemnify,
* defend, and hold each Contributor harmless for any liability
* incurred by, or claims asserted against, such Contributor by reason
* of your accepting any such warranty or additional liability.
*
* END OF TERMS AND CONDITIONS
*
* APPENDIX: How to apply the Apache License to your work.
*
* To apply the Apache License to your work, attach the following
* boilerplate notice, with the fields enclosed by brackets "[]"
* replaced with your own identifying information. (Don't include
* the brackets!) The text should be enclosed in the appropriate
* comment syntax for the file format. We also recommend that a
* file or class name and description of purpose be included on the
* same "printed page" as the copyright notice for easier
* identification within third-party archives.
*
* Copyright [yyyy] [name of copyright owner]
*
* Licensed under the Apache License, Version 2.0 (the "License);
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

W3C SOFTWARE NOTICE AND LICENSE
https://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other
related items) is being provided by the copyright holders under the following
license. By obtaining, using and/or copying this work, you (the licensee) agree
that you have read, understood, and will comply with the following terms and
conditions.

Permission to copy, modify, and distribute this software and its documentation,
with or without modification, for any purpose and without fee or royalty is
hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission.

Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered under http://saxproject.org/?selected=PD

SAX2 is Free!
I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com  
2000-05-05  
Apache Ant  
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (https://www.apache.org/).

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.

1.32 batik-constants 1.13

1.32.1 Available under license :

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

No license file was found, but licenses were detected in source scan.
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1135872551_1613620997.23/0/batik-constants-1-13-sources-1-
jar/org/apache/batik/constants/XMLConstants.java

1.33 prompt-tmp 1.0.0

1.33.1 Available under license :

Copyright (c) 2010 Nodejitsu Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.34 cliui 7.0.4 7.0.4

1.34.1 Available under license:
Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.35 hamcrest-library 1.3

1.35.1 Available under license:
BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

1.36 spdx-expression-parse 3.0.1

1.36.1 Available under license :
The MIT License

Copyright (c) 2015 Kyle E. Mitchell & other authors listed in AUTHORS

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.37 escape-string-regexp 1.0.5

1.37.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.38 optionator 0.8.3 0.8.3
1.38.1 Available under license:

Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.39 globals 9.18.0
1.39.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.40 @babel/core 7.13.14

1.40.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.41 hive-exec 3.1.2

1.41.1 Available under license:

Hive Query Language
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

(an example is provided in the Appendix below).
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.42 mkdirp 0.5.5
1.42.1 Available under license :
Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.43 acorn-jsx 3.0.1

1.43.1 Available under license:
Copyright (C) 2012-2014 by Ingvar Stepanyan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.44 eslint-module-utils 2.6.0

1.44.1 Available under license:
The MIT License (MIT)
Copyright (c) 2015 Ben Mosher

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.45 xmlhttprequest-ssl 1.5.3 1.5.3

1.45.1 Available under license:
Copyright (c) 2010 passive.ly LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.46 kerby-xdr 1.0.0

1.46.1 Available under license:

Kerby XDR Project
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
1.47 typedarray 0.0.6

1.47.1 Available under license:

/*
   Copyright (c) 2010, Linden Research, Inc.
   Copyright (c) 2012, Joshua Bell

   Permission is hereby granted, free of charge, to any person obtaining a copy
   of this software and associated documentation files (the "Software"), to deal
   in the Software without restriction, including without limitation the rights
   to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
   copies of the Software, and to permit persons to whom the Software is
   furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in
   all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
   IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
   FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
   AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
   LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
   OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
   THE SOFTWARE.

   */

   // Original can be found at:
   //   https://bitbucket.org/lindenlab/llsd
   // Modifications by Joshua Bell inexorabletash@gmail.com
   //   https://github.com/inexorabletash/polyfill

   // ES3/ES5 implementation of the Khronos Typed Array Specification
   // Ref: http://www.khronos.org/registry/typedarray/specs/latest/
   // Date: 2011-02-01

   // Variations:
   // * Allows typed_array.get/set() as alias for subscripts (typed_array[])
1.48.1 Available under license:

Copyright (c) 2013 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.49 uuid 3.4.0 3.4.0

1.49.1 Available under license:

The MIT License (MIT)

Copyright (c) 2010-2016 Robert Kieffer and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.50 camelcase 2.1.1

1.50.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.51 typetools 0.5.0

1.52 babel-plugin-transform-member-expression-literals 7.12.13

1.52.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.53 jnr-netdb 0.4

1.53.1 Available under license:

This code is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3 only, as published by the Free Software Foundation.

This code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License version 3 for more details.

You should have received a copy of the GNU Lesser General Public License version 3 along with this work. If not, see <http://www.gnu.org/licenses/>.

BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.
The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERM AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a
covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.
5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product. A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

"Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source. The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information. But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed. Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law. If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.
Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

1.54 pdfjs-dist 2.0.305 2.0.305
1.54.1 Available under license :

%%Copyright: -----------------------------------------------------------
%%Copyright: Copyright 1990-2009 Adobe Systems Incorporated.
%%Copyright: All rights reserved.
%%Copyright:
%%Copyright: Redistribution and use in source and binary forms, with or
%%Copyright: without modification, are permitted provided that the
%%Copyright: following conditions are met:
%%Copyright:
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.55 locate-path 2.0.0

1.55.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.56 har-validator 5.1.5

1.56.1 Available under license :

MIT License

Copyright (c) 2018 Ahmad Nassri <ahmad@ahmadnassri.com>
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.57 underscore 1.9.2 1.9.2

1.57.1 Available under license:

Copyright (c) 2009-2018 Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.58 ext 1.4.0
1.58.1 Available under license:
ISC License

Copyright (c) 2011-2019, Mariusz Nowak, @medikoo, medikoo.com

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.59 whatwg-fetch 2.0.4 2.0.4
1.59.1 Available under license:
Copyright (c) 2014-2016 GitHub, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.60 resolve-url 0.2.1
### 1.60.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.61 jackson-databind 2.10.5.1

#### 1.61.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

# 1.62 netty-transport-native-unix-common

## 4.1.59.Final

### 1.62.1 Available under license:

No license file was found, but licenses were detected in source scan.

```java
/*
 * Copyright 2015 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 *   https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

Found in path(s):

* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/io/netty/channel/unix/NativeInetAddress.java
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/io/netty/channel/unix/Errors.java
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/io/netty/channel/unix/UnixChannel.java
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/io/netty/channel/unix/UnixChannel.java
/*
 * Copyright 2017 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 *   https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 *   https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

No license file was found, but licenses were detected in source scan.
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
*   https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native- unix-common-4-1-59-final-sources-
  jar/netty_unix_buffer.h
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native- unix-common-4-1-59-final-sources-
  jar/netty_unix_buffer.c
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native- unix-common-4-1-59-final-sources-
  jar/io/netty/channel/unix/PreferredDirectByteBufAllocator.java
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native- unix-common-4-1-59-final-sources-
  jar/io/netty/channel/unix/Buffer.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
*   https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native- unix-common-4-1-59-final-sources-
  jar/io/netty/channel/unix/LimitsStaticallyReferencedJniMethods.java
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native- unix-common-4-1-59-final-sources-
  jar/io/netty/channel/unix/ErrorsStaticallyReferencedJniMethods.java
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native- unix-common-4-1-59-final-sources-
  jar/netty_unix_util.h
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native- unix-common-4-1-59-final-sources-
  jar/netty_unix_limits.h
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/netty_unix_util.c
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/netty_unix_limits.c
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/io/netty/channel/unix/PeerCredentials.java
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/io/netty/channel/unix/Limits.java

No license file was found, but licenses were detected in source scan.

/
* Copyright 2020 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/netty_unix.h
* /opt/cola/permits/1138680848_1614312229.62/0/netty-transport-native-unix-common-4-1-59-final-sources-jar/netty_unix.c

No license file was found, but licenses were detected in source scan.

~ Copyright 2016 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
1.63 fd-slicer 1.1.0

1.63.1 Available under license:

Copyright (c) 2014 Andrew Kelley

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.64 prepend-http 2.0.0

1.64.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.65 expiringmap 0.5.8

1.66 diff-match-patch 1.0.5

1.66.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"...
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.67 ansi-escapes 1.4.0

1.67.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.68 org.jacoco.agent.rt 0.7.9

1.68.1 Available under license:

License
=======

Copyright (c) 2009, 2017 Mountainminds GmbH & Co. KG and Contributors

The JaCoCo Java Code Coverage Library and all included documentation is made available by Mountainminds GmbH & Co. KG, Munich. Except indicated below, the Content is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is available at [http://www.eclipse.org/legal/epl-v10.html](http://www.eclipse.org/legal/epl-v10.html).

1.69 optionator 0.8.3

1.69.1 Available under license:
Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.70 checker-qual 3.5.0

1.70.1 Available under license:
Checker Framework qualifiers
Copyright 2004-present by the Checker Framework developers

MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.71 libevent 1.4.13

1.71.1 Available under license:

CMake - Cross Platform Makefile Generator
Copyright 2000-2013 Kitware, Inc.
Copyright 2000-2011 Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the names of Kitware, Inc., the Insight Software Consortium, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The above copyright and license notice applies to distributions of
CMake in source and binary form. Some source files contain additional notices of original copyright by their contributors; see each source for details. Third-party software packages supplied with CMake under compatible licenses provide their own copyright notices documented in corresponding subdirectories.

------------------------------------------------------------------------------

CMake was initially developed by Kitware with the following sponsorship:

* National Library of Medicine at the National Institutes of Health as part of the Insight Segmentation and Registration Toolkit (ITK).

* US National Labs (Los Alamos, Livermore, Sandia) ASC Parallel Visualization Initiative.

* National Alliance for Medical Image Computing (NAMIC) is funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149.

* Kitware, Inc.

Libevent is available for use under the following license, commonly known as the 3-clause (or "modified") BSD license:

===============================================
Copyright (c) 2000-2007 Niels Provos <provos@citi.umich.edu>
Copyright (c) 2007-2012 Niels Provos and Nick Mathewson

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==============================================

Portions of Libevent are based on works by others, also made available by them under the three-clause BSD license above. The copyright notices are available in the corresponding source files; the license is as above. Here's a list:

log.c:
  Copyright (c) 2000 Dug Song <dugsong@monkey.org>
  Copyright (c) 1993 The Regents of the University of California.

strlcpy.c:
  Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

win32select.c:
  Copyright (c) 2003 Michael A. Davis <mike@datanerds.net>

evport.c:
  Copyright (c) 2007 Sun Microsystems

ht-internal.h:
  Copyright (c) 2002 Christopher Clark

minheap-internal.h:
  Copyright (c) 2006 Maxim Yegorushkin <maxim.yegorushkin@gmail.com>

==============================================

The arc4module is available under the following, sometimes called the "OpenBSD" license:

Copyright (c) 1996, David Mazieres <dm@uun.org>
Copyright (c) 2008, Damien Miller <djm@openbsd.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

==============================================
The Windows timer code is based on code from libutp, which is distributed under this license, sometimes called the "MIT" license.

Copyright (c) 2010 BitTorrent, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE POI SUBCOMPONENTS:

Apache POI includes subcomponents with separate copyright notices and
license terms. Your use of these subcomponents is subject to the terms
and conditions of the following licenses:

Office Open XML schemas (poi-ooxml-full-* .jar)

The Office Open XML schema definitions used by Apache POI are
a part of the Office Open XML ECMA Specification (ECMA-376, [1]).
As defined in section 9.4 of the ECMA bylaws [2], this specification
is available to all interested parties without restriction:

9.4 All documents when approved shall be made available to
all interested parties without restriction.

Furthermore, both Microsoft and Adobe have granted patent licenses
to this work [3,4,5].


Bouncy Castle library (bcprov-* .jar, bcpg-* .jar, bcpkix-* .jar)

Copyright (c) 2000 - 2021 The Legion of the Bouncy Castle Inc.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CurvesAPI / Curve API

BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SLF4J library (slf4j-api-*-jar)

Copyright (c) 2004-2017 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

inbot-utils (https://github.com/Inbot/inbot-utils)

The MIT License (MIT)

Copyright (c) 2015 Inbot

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache POI
Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

This product contains parts that were originally based on software from BEA.
Copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/> (dead link),
which was acquired by Oracle Corporation in 2008.
<https://en.wikipedia.org/wiki/BEA_Systems>

This product contains W3C XML Schema documents. Copyright 2001-2003 (c)
World Wide Web Consortium (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University)

This product contains the chunks_parse_cmds.tbl file from the vsdump program.
Copyright (C) 2006-2007 Valek Filippov (frob@df.ru)

This product contains parts of the eID Applet project
Copyright (c) 2009-2018
FedICT (federal ICT department of Belgium), e-Contract.be BVBA (https://www.e-contract.be),
Bart Hanssens from FedICT

1.73 protobuf-java 2.5.0

1.74 jul-to-slf4j 1.7.26

1.75 mockito 2.23.4
1.75.1 Available under license:
The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.76 babel-plugin-syntax-trailing-function-commas 6.22.0
1.76.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.77 file-type 11.1.0

1.77.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.78 jetty-security 9.4.34.v20201102

1.78.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient’s responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such
Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.
5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and
* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------
Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------
OW2
The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------
Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jettyschemas
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.79 jffi 1.2.16

1.79.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that

Open Source Used In DNAC Platform Antelope 332
remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.80 jmockit 1.47

1.80.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2006 JMockit developers
 * This file is subject to the terms of the MIT license (see LICENSE.txt).
 */
Found in path(s):
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/util/Utilities.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/expectations/mocking/ParameterTypeRedefinitions.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/expectations/mocking/MockedType.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/TestedParameters.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/expectations/BaseVerificationPhase.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/full/TestConversation.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/classGeneration/ImplementationClass.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/integration/junit4/JUnit4TestRunnerDecorator.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/TestRun.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/CoveragePercentage.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/Mocked.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/integration/testng/package-info.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/util/StackTrace.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/dataItems/InstanceFieldData.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/dataItems/StaticFieldData.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/asm/metadata/ClassMetadataReader.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/reporting/lineCoverage/LineSegmentsFormatter.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/Capturing.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/full/FullInjection.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/modification/ClassSelection.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/invocation/ArgumentValuesAndMatchersWithVarargs.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/modification/ClassSelection.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/invocation/MultiValuedConversion.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/lineCoverage/LineSegmentsFormatter.java
Open Source Used In DNAC Platform Antelope 342

jar/mockit/internal/expectations/invocation/ExpectationError.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/BeanExporter.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/expectations/argumentMatching/PatternMatcher.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/expectations/invocation/InvocationArguments.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/field/FieldToInject.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/InjectionPoint.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/MockingFilters.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/package-info.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/reporting/sourceFiles/InputFile.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/expectations/transformation/InvocationBlockModifier.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/expectations/argumentMatching/SubstringMatcher.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/expectations/mocking/CaptureOfNewInstances.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/Injector.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/reflection/MethodReflection.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/integration/testng/TestNGRunnerDecorator.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/reporting/parsing/FileParser.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/coverage/reporting/lineCoverage/LineCoverageOutput.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/reflection/RealMethodOrConstructor.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/TestedObjectCreation.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/TestedObjectInstantiations.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/injection/TestedClassInstantiations.java
* /opt/cola/permits/1202446415_1631568333.62/0/jmockit-1-47-sources-jar/mockit/internal/classGeneration/MockedTypeInfo.java
1.81 isemail 1.2.0

1.81.1 Available under license:

Copyright 2008-2011, Dominic Sayers
Copyright 2013-2014, GlobeSherpa
Copyright 2014-2015, Eli Skeggs

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Dominic Sayers nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.82 jetty-io 9.3.27.v20190418

1.82.1 Available under license:

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular
Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself
or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are
separate modules of software distributed in conjunction with the Program under their own license agreement, and
(ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or
sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide,
royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform,
distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and
object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide,
royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the
Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the
combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such
addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall
not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no
assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual
property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any
other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights
and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual
property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute
the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to
grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided
that:
a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's
responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program
(including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.83 glob 7.1.3
1.83.1 Available under license:
MIT License

Copyright (c) Microsoft Corporation.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.84 pluralize 1.2.1

1.84.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.85 jackson 2.11.4

1.85.1 Available under license:

===========================================================================
== NOTICE file corresponding to the section 4 d of                      ==
== the Apache License, Version 2.0,                                   ==
== in this case for the Apache Camel distribution.                     ==
===========================================================================

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Please read the different LICENSE files present in the licenses directory of this distribution.

Camel :: Jackson
Copyright 2007-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.86 akka 2.5.31
1.86.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.
We call this license the "Lesser" General Public License because it does less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License").

Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses
the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, 5th Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the distributors holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this
work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The Netty Project
=================

Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2011 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

---------------------------------------------------------------------------
This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified version of 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD Style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product contains a modified version of 'Webbit', a Java event based WebSocket and HTTP server:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'SLF4J', a simple logging facade for Java, which can be obtained at:
This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'JBoss Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://anonsvn.jboss.org/repos/common/common-logging-spi/

This product optionally depends on 'Apache Felix', an open source OSGi framework implementation, which can be obtained at:

* LICENSE:
  * license/LICENSE.felix.txt (Apache License 2.0)
* HOMEPAGE:
  * http://felix.apache.org/

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCraft,
INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(BSD License: http://www.opensource.org/licenses/bsd-license)
Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the
following conditions are met:

* Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other
materials provided with the distribution.

* Neither the name of the Webbit nor the names of
  its contributors may be used to endorse or promote products
  derived from this software without specific prior written
  permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
/*
 * Copyright (c) 2004-2007 QOS.ch
 * All rights reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.87 moment-timezone 0.5.21 0.5.21
1.87.1 Available under license:
The MIT License (MIT)

Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.88 babel-preset-stage-2 6.24.1
1.88.1 Available under license:

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

MIT License

Copyright (c) 2014-2017 Sebastian McKenzie <sebmck@gmail.com>
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2015 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

1.89 type-is 1.6.16 1.6.16

1.89.1 Available under license :
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

1.90 jackson-dataformat-cbor 2.6.7
1.90.1 Available under license:
This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.91 dayjs 1.9.3
1.91.1 Available under license:
MIT License

Copyright (c) 2018-present, iamkun

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.92 @babel/parser 7.13.13
1.92.1 Available under license:
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.93 listenercount 1.0.1
1.93.1 Available under license :
Copyright (c) MMXV jden <jason@denizac.org>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.94 is-arrayish 0.2.1
1.94.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 JD Ballard

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.95 socket.io-client 2.3.0
1.95.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Guillermo Rauch

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.96 amcharts3-animate 1.1.2 1.1.2
1.96.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.97 hadoop-hdfs-client 3.2.2

1.97.1 Available under license:

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of Gson 2.2.4,
which has the following notices:

The Netty Project
=================
Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

---

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and
decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSEprotobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/
This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
  * http://logging.apache.org/log4j/
This product optionally depends on ‘Aalto XML’, an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of ‘HPACK’, a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of ‘Apache Commons Lang’, a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of ‘JDOM 1.1’, which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors. Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:

* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())
The binary distribution of this product bundles binaries of
Apache Log4j 1.2.17,
which has the following notices:
* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of
"Java Concurrency in Practice" book annotations 1.0,
which has the following notices:
* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative
  Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
  Official home: http://www.jcip.net Any republication or derived work
distributed in source code form must include this copyright and license
  notice.

The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,
which has the following notices:
* ==============================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
  ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License
Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html
org.ow2.asm:asm-commons
org.ow2.asm:asm

------
Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida,
Open Source Used In DNAC Platform Antelope  402
modified April 2001  by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt
for non-commercial or commercial purposes and without fee is
granted provided that the copyright notice appears in all copies./

The binary distribution of this product bundles binaries of
Snappy for Java 1.0.4.1,
which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
(Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by
"GCC RUNtime Library Exception"

== Contributors ==
  * Tatu Saloranta
    * Providing benchmark suite
  * Alec Wysoker
    * Performance and memory usage improvement

The binary distribution of this product bundles binaries of
Xerces2 Java Parser 2.9.1,
which has the following notices:
* =========================================================================
  ==  NOTICE file corresponding to section 4(d) of the Apache License,   ==
  ==  Version 2.0, in this case for the Apache Xerces Java distribution. ==
  =========================================================================

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
  - voluntary contributions made by Paul Eng on behalf of the
    Apache Software Foundation that were originally developed at iClick, Inc.,
    software copyright (c) 1999.

The binary distribution of this product bundles binaries of
Logback Classic Module 1.1.2.
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0
--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:
--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.
Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of
Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library
distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5
(https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schreoder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-
ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-
The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1,
which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9
which has the following notices:
* =----------------------------------------------------------------------------------------------------------------------------------
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=----------------------------------------------------------------------------------------------------------------------------------
This product includes software developed by
Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 * Copyright (c) 2005, European Commission project OneLab under contract 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 * - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*4++
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 * Abstract: The main routine
 * -- */

Open Source Used In DNAC Platform Antelope 411
LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.

THE SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :
- LZ4 source repository : http://code.google.com/p/lz4/
- LZ4 public forum : https://groups.google.com/forum/#!forum/lz4c

/*

For hadoop-common-project/hadoop-common/src/main/native/gtest

---------------------------------------------------------------------
Copyright 2008, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

Open Source Used In DNAC Platform Antelope 412
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1

(OpenBSD license)
This is a work derived from Russ Cox's RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

* Neither the name of FuseSource Corp. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from
the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

=============

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

--------------------------------------------------------------------------------

Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

--------------------------------------------------------------------------------
Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
em-helpers v0.5.13
em-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js
--------------------------------------------------------------------------------
The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js
--------------------------------------------------------------------------------
The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery
- Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
- Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- oj! Algorithms - version 43.0


This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js

_____________________________________________________________________________

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

D3 is available under a 3-clause BSD license. For details, see:

hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:

HSQLDB Database 2.3.4

(HSQL License)

"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQldb.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.

The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. **Original Software** means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. **Patent Claims** means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. **Source Code** means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. **You** (or **Your** means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The **Initial Developer Grant**. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.
2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your
Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in
Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a
Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to
make it enforceable. This License shall be governed by the law of the
d Jurisdiction specified in a notice contained within the Original Software
(except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
grizzly-http 2.2.21
grizzly-http-server 2.2.21
grizzly-http-servlet 2.2.21
grizzly-rcm 2.2.21

( CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE ( CDDL) Version 1.1

1. Definitions.
1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to
third party intellectual property claims, the Initial Developer hereby grants
You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual
property rights (other than patent or trademark) Licensable by Initial
Developer, to use, reproduce, modify, display, perform, sublicense and
distribute the Original Software (or portions thereof), with or without
Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims
infringed by the making, using or selling of Original Software, to make, have
made, use, practice, sell, and offer for sale, and/or otherwise dispose of the
Original Software (or portions thereof).
(c) The licenses granted in Sections
2.1(a) and (b) are effective on the date Initial Developer first distributes or
otherwise makes the Original Software available to a third party under the terms
of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is
granted: (1) for code that You delete from the Original Software, or (2) for
infringements caused by: (i) the modification of the Original Software, or (ii)
the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and
subject to third party intellectual property claims, each Contributor hereby
grants You a world-wide, royalty-free, non-exclusive license:
(a) under
intellectual property rights (other than patent or trademark) Licensable by
Contributor to use, reproduce, modify, display, perform, sublicense and
distribute the Modifications created by such Contributor (or portions thereof),
either on an unmodified basis, with other Modifications, as Covered Software
and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the
making, using, or selling of Modifications made by that Contributor either alone
and/or in combination with its Contributor Version (or portions of such
combination), to make, use, sell, offer for sale, have made, and/or otherwise
dispose of: (1) Modifications made by that Contributor (or portions thereof);
and (2) the combination of Modifications made by that Contributor with its
Contributor Version (or portions of such combination).
(c) The licenses granted
in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications available to a third
party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2)
for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with
other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every
Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.
6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR
8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

--------------------------------------------------------------------------------
By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1
--------------------------------------------------------------------------------
(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.
"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and
fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for
damages, including direct, indirect, special, incidental and consequential
damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by
that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor,
and informs licensees how to obtain it in a reasonable manner on or through a
medium customarily used for software exchange.
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the
Program.

Each Contributor must identify itself as the originator of its Contribution, if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore, if
a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses, damages
and costs (collectively "Losses") arising from claims, lawsuits and other legal
actions brought by a third party against the Indemnified Contributor to the
extent caused by the acts or omissions of such Commercial Contributor in
connection with its distribution of the Program in a commercial product
offering. The obligations in this section do not apply to any claims or Losses
relating to any actual or alleged intellectual property infringement. In order
to qualify, an Indemnified Contributor must: a) promptly notify the Commercial
Contributor in writing of such claim, and b) allow the Commercial Contributor to
control, and cooperate with the Commercial Contributor in, the defense and any
related settlement negotiations. The Indemnified Contributor may participate in
any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product
offering, Product X. That Contributor is then a Commercial Contributor. If that
Commercial Contributor then makes performance claims, or offers warranties
related to Product X, those performance claims and warranties are such
Commercial Contributor's responsibility alone. Under this section, the
Commercial Contributor would have to defend claims against the other
Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

--------------------------------------------------------------------------------
(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

(2-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR
OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and
perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.
For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats.
All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by
clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license
terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

jamon-runtime 2.4.1

--------------------------------------------------------------------------------
(MPL 2.0)
Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.
1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0,
or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form
how they can obtain a copy of such Source Code Form by reasonable means in a
timely manner, at a charge no more than the cost of distribution to the
recipient; and

You may distribute such Executable Form under the terms of this License, or
sublicense it under different terms, provided that the license for the
Executable Form does not attempt to limit or alter the recipients rights in
the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided
that You also comply with the requirements of this License for the Covered
Software. If the Larger Work is a combination of Covered Software with a work
governed by one or more Secondary Licenses, and the Covered Software is not
Incompatible With Secondary Licenses, this License permits You to additionally
distribute such Covered Software under the terms of such Secondary License(s),
so that the recipient of the Larger Work may, at their option, further
distribute the Covered Software under the terms of either this License or such
Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including
copyright notices, patent notices, disclaimers of warranty, or limitations of
liability) contained within the Source Code Form of the Covered Software, except
that You may alter any license notices to the extent required to remedy known
factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity
or liability obligations to one or more recipients of Covered Software. However,
You may do so only on Your own behalf, and not on behalf of any Contributor. You
must make it absolutely clear that any such warranty, support, indemnity, or
liability obligation is offered by You alone, and You hereby agree to indemnify
every Contributor for any liability incurred by such Contributor as a result of
warranty, support, indemnity or liability terms You offer. You may include
additional disclaimers of warranty and limitations of liability specific to any
jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with
respect to some or all of the Covered Software due to statute, judicial order,
or regulation then You must: (a) comply with the terms of this License to the
maximum extent possible; and (b) describe the limitations and the code they
affect. Such description must be placed in a text file included with all
distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any
direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this
10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:

JDOM 1.1

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN
License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.
2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made
subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution
in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will
survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under
1.98 redux-act 1.7.4 1.7.4

1.98.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.99 tweetnacl 0.14.5
1.99.1 Available under license:

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.
In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <http://unlicense.org>

1.100 dtrace-provider 0.6.0

1.100.1 Available under license:

Copyright 2011 Chris Andrews. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY CHRIS ANDREWS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CHRIS ANDREWS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2012 Chris Andrews. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of
   conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list
   of conditions and the following disclaimer in the documentation and/or other materials
   provided with the distribution.

THIS SOFTWARE IS PROVIDED BY CHRIS ANDREWS ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CHRIS ANDREWS
OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.101 doctrine 2.1.0
1.101.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

---

Open Source Used In DNAC Platform Antelope 472
as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
1.102 string_decoder 1.0.3 1.0.3

1.102.1 Available under license:

Node.js is licensed for use as follows:

```text
---

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
---

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

```text
---

Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
1.103 httpcomponents-core 4.4.13

1.103.1 Available under license:

Apache HttpCore
Copyright 2005-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and eachContributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.104 proxy-addr 2.0.6

1.104.1 Available under license:

(The MIT License)

Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.105 @babel/plugin-transform-arrow-functions 7.13.0

1.105.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.106 require-uncached 1.0.3

1.106.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.107 require-directory 2.1.1 2.1.1
1.107.1 Available under license:
The MIT License (MIT)

Copyright (c) 2011 Troy Goode <troygoode@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.108 intl-messageformat-parser 1.4.0 1.4.0
1.108.1 Available under license:

Copyright 2014 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

--------------------------------------------------------------------------------

Inspired by and derived from:
messageformat.js https://github.com/SlexAxton/messageformat.js
Copyright 2014 Alex Sexton
Apache License, Version 2.0

1.109 inquirer 6.2.2
1.109.1 Available under license :

No license file was found, but licenses were detected in source scan.

Licensed under the MIT license.

Found in path(s):
* /opt/cola/permits/1217889055_1634859870.64/0/inquirer-6-2-2-2-tgz/package/README.md

1.110 vary 1.0.1 1.0.1
1.110.1 Available under license :

(The MIT License)

Copyright (c) 2014-2015 Douglas Christopher Wilson
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.111 jetty-util-ajax 9.4.41.v20210516

1.111.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates'
from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
   a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
   b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
   c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
   d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright
3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial
product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the
parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.112 regenerator-runtime 0.13.7 0.13.7
1.112.1 Available under license:
MIT License

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
1.113 hbase-metrics-api 2.4.2
1.113.1 Available under license:

Apache HBase - Metrics API
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.114 lucene 7.4.0
1.114.1 Available under license :

# JTS Topology Suite Licensing

May 30th, 2017

## Project Licenses

The Eclipse Foundation makes available all content in this project ("Content"). Unless otherwise indicated below, the Content is provided to you under the terms and conditions of either the [Eclipse Public License 1.0](https://www.eclipse.org/legal/epl-v10.html) ("EPL") or the [Eclipse Distribution License 1.0](http://www.eclipse.org/org/documents/edl-v10.php) (a BSD Style License). For purposes of the EPL, "Program" will mean the Content.

If you did not receive this Content directly from the Eclipse Foundation, the Content is being redistributed by another party ("Redistributor") and different terms and conditions may apply to your use of any object code in the Content. Check the Redistributor's license that was provided with the Content. If no such license exists, contact the Redistributor. Unless otherwise indicated below, the terms and conditions of the EPL still apply to any source code in the Content and such source code may be obtained at http://www.eclipse.org.
## Third Party Content

The Content includes items that have been sourced from third parties as set out below. If you did not receive this Content directly from the Eclipse Foundation, the following is provided for informational purposes only, and you should look to the Redistributor's license for terms and conditions of use.

### Jdom-contrib version 1.1.3

The JTS tests module contains source from JDOM Contrib 1.1.3 which is available under the Jdom License [1]. The two files are LineNumberElement.java and LineNumberSAXHandler.java.

Typical users of this library will not need to depend on this module.


### GeoTools

JTS includes some code from the GeoTools project. This code has been licensed to the JTS project under the OSGeo BSD License [2] by the GeoTools PSC [3,4].


```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Eclipse.org Software User Agreement</title>
</head><body lang="EN-US" link="blue" vlink="purple">
<h2>Eclipse Foundation Software User Agreement</h2>
<p>March 17, 2005</p>
<h3>Usage Of Content</h3>
<p>THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY NOT USE THE CONTENT.</p>
</body></html>
```
<h3>Applicable Licenses</h3>

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is also available at <a href="http://www.eclipse.org/legal/epl-v10.html">http://www.eclipse.org/legal/epl-v10.html</a>. For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the Eclipse.org CVS repository ("Repository") in CVS modules ("Modules") and made available as downloadable archives ("Downloads").

Content may be structured and packaged into modules to facilitate delivering, extending, and upgrading the Content. Typical modules may include plug-ins ("Plug-ins"), plug-in fragments ("Fragments"), and features ("Features"). Each Plug-in or Fragment may be packaged as a sub-directory or JAR (Java® ARchive) in a directory named "plugins".

A Feature is a bundle of one or more Plug-ins and/or Fragments and associated material. Each Feature may be packaged as a sub-directory in a directory named "features". Within a Feature, files named "feature.xml" may contain a list of the names and version numbers of the Plug-ins and/or Fragments associated with that Feature.

Features may also include other Features ("Included Features"). Within a Feature, files named "feature.xml" may contain a list of the names and version numbers of Included Features.

The terms and conditions governing Plug-ins and Fragments should be contained in files named "about.html" ("Abouts"). The terms and conditions governing Features and Included Features should be contained in files named "license.html" ("Feature Licenses"). Abouts and Feature Licenses may be located in any directory of a Download or Module including, but not limited to the following locations:

The top-level (root) directory
Plug-in and Fragment directories
Inside Plug-ins and Fragments packaged as JARs
Sub-directories of the directory named "src" of certain Plug-ins
Feature directories
Note: if a Feature made available by the Eclipse Foundation is installed using the Eclipse Update Manager, you must agree to a license ("Feature Update License") during the installation process. If the Feature contains Included Features, the Feature Update License should either provide you with the terms and conditions governing the Included Features or inform you where you can locate them. Feature Update Licenses may be found in the "license" property of files named "feature.properties" found within a Feature. Such Abouts, Feature Licenses, and Feature Update Licenses contain the terms and conditions (or references to such terms and conditions) that govern your use of the associated Content in that directory.

THE ABOUTS, FEATURE LICENSES, AND FEATURE UPDATE LICENSES MAY REFER TO THE EPL OR OTHER LICENSE AGREEMENTS, NOTICES OR TERMS AND CONDITIONS. SOME OF THESE OTHER LICENSE AGREEMENTS MAY INCLUDE (BUT ARE NOT LIMITED TO):</p>

<ul>
<li>Apache Software License 1.1 (available at http://www.apache.org/licenses/LICENSE)</li>
<li>Apache Software License 2.0 (available at http://www.apache.org/licenses/LICENSE-2.0)</li>
<li>IBM Public License 1.0 (available at http://oss.software.ibm.com/developerworks.opensource/license10.html)</li>
<li>Metro Link Public License 1.00 (available at http://www.opengroup.org/openmotif/supporters/metrolink/license.html)</li>
</ul>

IT IS YOUR OBLIGATION TO READ AND ACCEPT ALL SUCH TERMS AND CONDITIONS PRIOR TO USE OF THE CONTENT. If no About, Feature License, or Feature Update License is provided, please contact the Eclipse Foundation to determine what terms and conditions govern that particular Content.

<h3>Cryptography</h3>

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use,
and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.</p>

<small>Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.</small>

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2018 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU
libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies of
the Software and that both the above copyright notice(s) and this
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale, use
or other dealings in this Software without prior written authorization
of the copyright holder.

All trademarks and registered trademarks mentioned herein are the
property of their respective owners.


# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
# 
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
# *
# * Libtabe (Chinese)
#   - https://sourceforge.net/project/?group_id=1519
#   - Its license terms and conditions are shown below.
# *
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#   - Its license terms and conditions are shown below.
# *
# ---------COPYING.libtabe ---- BEGIN---------------------
# /*
# * Copyright (c) 1999 TaBE Project.
# * Copyright (c) 1999 Pai-Hsiang Hsiao.
# * All rights reserved.

# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# *. Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# *. Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# *. Neither the name of the TaBE Project nor the names of its
# * contributors may be used to endorse or promote products derived
# * from this software without specific prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# /*
# * Copyright (c) 1999 Computer Systems and Communication Lab,
# * Institute of Information Science, Academia
# * Sinica. All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# *. Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# *. Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# *. Neither the name of the Computer Systems and Communication Lab
# * nor the names of its contributors may be used to endorse or
# * promote products derived from this software without specific
# * prior written permission.
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */

# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-tsei4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsei4
#
# -------------COPYING.libtabe-----END---------------------------
#
# -------------COPYING.ipadic-----BEGIN------------------------
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.
3. Lao Word Break Dictionary Data (laodict.txt)

# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
#         (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# ---------------------------------------------------------------
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
# ---------------------------------------------------------------

4. Burmese Word Break Dictionary Data (burmesedict.txt)

# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
# github.com/kanyawtech/myanmar-karen-word-lists
#
# .............................................................................................
# Copyright (c) 2013, LeRoy Benjamin Sharon
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
# disclaimer. Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided
# with the distribution.
#
# Neither the name Myanmar Karen Word Lists, nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
# BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
# TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
# TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
# THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
# SUCH DAMAGE.
# .............................................................................................

5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

# 7. Database Ownership
#
# The TZ database itself is not an IETF Contribution or an IETF document. Rather it is a pre-existing and regularly updated work that is in the public domain, and is intended to remain in the public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do
# not apply to the TZ Database or contributions that individuals make to it. Should any claims be made and substantiated against the TZ Database, the organization that is providing the IANA Considerations defined in this RFC, under the memorandum of understanding with the IETF, currently ICANN, may act in accordance with all competent court orders. No ownership claims will be made by ICANN or the IETF Trust on the database or the code. Any person making a contribution to the database or code waives all rights to future claims in that contribution or in the TZ Database.

6. Google double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

UIMA Annotator: Tagger
Copyright 2006-2010 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

/*
 * Apache License
 * Version 2.0, January 2004
 * http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner.
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

W3C SOFTWARE NOTICE AND LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other
related items) is being provided by the copyright holders under the following
license. By obtaining, using and/or copying this work, you (the licensee) agree
that you have read, understood, and will comply with the following terms and
conditions.

Permission to copy, modify, and distribute this software and its documentation,
with or without modification, for any purpose and without fee or royalty is
hereby granted, provided that you include the following on ALL copies of the
software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the
   redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms
   and conditions. If none exist, the W3C Software Short Notice should be
   included (hypertext is preferred, text is permitted) within the body
   of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date
   changes were made. (We recommend you provide URIs to the location from
   which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE
NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT
THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY
PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or
publicity pertaining to the software without specific, written prior permission.
Title to copyright in this software and any associated documentation will at
all times remain with copyright holders.
This formulation of W3C's notice and license became active on December 31, 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered under http://saxproject.org/?selected=pd

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2016 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ICU4J, (under lucene/analysis/icu) is licensed under an MIT style license (modules/analysis/icu/lib/icu4j-LICENSE-BSD_LIKE.txt) and Copyright (c) 1995-2012 International Business Machines Corporation and others

[The "BSD license"]

Copyright (c) 2015 Terence Parr, Sam Harwell
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which:
   i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and
   ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no
assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a)
promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's
patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2012 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

# about.md file

## About This Content

May 22, 2015

### License

The Eclipse Foundation makes available all content in this plug-in ("Content"). Unless otherwise indicated below, the Content is provided to you under the terms and conditions of the Apache License, Version 2.0. A copy of the Apache License, Version 2.0 is available at [http://www.apache.org/licenses/LICENSE-2.0.txt](http://www.apache.org/licenses/LICENSE-2.0.txt)

If you did not receive this Content directly from the Eclipse Foundation, the Content is being redistributed by another party ("Redistributor") and different terms and conditions may apply to your use of any object code in the Content. Check the Redistributors license that was provided with the Content. If no such license exists, contact the Redistributor. Unless otherwise indicated below, the terms and conditions of the Apache License, Version 2.0 still apply to any source code in the Content and such source code may be obtained at [http://www.eclipse.org](http://www.eclipse.org).

# notice.md file

Note: the below Eclipse user agreement is standard. It says "Unless otherwise indicated, ... before referring to the EPL. We indicate above that all content is licensed under the ASLv2 license. -- David Smiley

## Eclipse Foundation Software User Agreement

April 9, 2014

### Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION
AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY NOT USE THE CONTENT.

### Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is also available at [http://www.eclipse.org/legal/epl-v10.html](http://www.eclipse.org/legal/epl-v10.html). For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the Eclipse Foundation source code repository ("Repository") in software modules ("Modules") and made available as downloadable archives ("Downloads").

* Content may be structured and packaged into modules to facilitate delivering, extending, and upgrading the Content. Typical modules may include plug-ins ("Plug-ins"), plug-in fragments ("Fragments"), and features ("Features").
  * Each Plug-in or Fragment may be packaged as a sub-directory or JAR (Java ARchive) in a directory named "plugins".
  * A Feature is a bundle of one or more Plug-ins and/or Fragments and associated material. Each Feature may be packaged as a sub-directory in a directory named "features". Within a Feature, files named "feature.xml" may contain a list of the names and version numbers of the Plug-ins and/or Fragments associated with that Feature.
  * Features may also include other Features ("Included Features"). Within a Feature, files named "feature.xml" may contain a list of the names and version numbers of Included Features.

The terms and conditions governing Plug-ins and Fragments should be contained in files named "about.html" ("Abouts").

The terms and conditions governing Features and Included Features should be contained in files named "license.html" ("Feature Licenses"). Abouts and Feature Licenses may be located in any directory of a Download or Module including, but not limited to the following locations:

* The top-level (root) directory
* Plug-in and Fragment directories
* Inside Plug-ins and Fragments packaged as JARs
* Sub-directories of the directory named "src" of certain Plug-ins
* Feature directories

Note: if a Feature made available by the Eclipse Foundation is installed using the Provisioning Technology (as defined below), you must agree to a license ("Feature Update License") during the installation process. If the Feature contains Included Features, the Feature Update License should either provide you with the terms and conditions governing the Included Features or inform you where you can locate them. Feature Update Licenses may be found in the "license" property of files named "feature.properties" found within a Feature. Such Abouts, Feature Licenses, and Feature Update Licenses contain the terms and conditions (or references to such terms and conditions) that govern your use of the associated Content in that directory.

THE ABOUTS, FEATURE LICENSES, AND FEATURE UPDATE LICENSES MAY REFER TO THE EPL OR OTHER LICENSE AGREEMENTS, NOTICES OR TERMS AND CONDITIONS. SOME OF THESE OTHER LICENSE AGREEMENTS MAY INCLUDE (BUT ARE NOT LIMITED TO):

* Eclipse Distribution License Version 1.0 (available at [http://www.eclipse.org/licenses/edl-v1.0.html](http://www.eclipse.org/licenses/edl-v1.0.html))
* Apache Software License 1.1 (available at [http://www.apache.org/licenses/LICENSE](http://www.apache.org/licenses/LICENSE))
* Apache Software License 2.0 (available at [http://www.apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0))

IT IS YOUR OBLIGATION TO READ AND ACCEPT ALL SUCH TERMS AND CONDITIONS PRIOR TO USE OF THE CONTENT. If no About, Feature License, or Feature Update License is provided, please contact the Eclipse Foundation to determine what terms and conditions govern that particular Content.

### Use of Provisioning Technology

The Eclipse Foundation makes available provisioning software, examples of which include, but are not limited to, p2 and the Eclipse Update Manager ("Provisioning Technology") for the purpose of allowing users to install software, documentation, information and/or other materials (collectively "Installable Software"). This capability is provided with the intent of allowing such users to install, extend and update Eclipse-based products. Information about packaging Installable Software is available at [http://eclipse.org/equinox/p2/repository_packaging.html](http://eclipse.org/equinox/p2/repository_packaging.html)
You may use Provisioning Technology to allow other parties to install Installable Software. You shall be responsible for enabling the applicable license agreements relating to the Installable Software to be presented to, and accepted by, the users of the Provisioning Technology in accordance with the Specification. By using Provisioning Technology in such a manner and making it available in accordance with the Specification, you further acknowledge your agreement to, and the acquisition of all necessary rights to permit the following:

1. A series of actions may occur ("Provisioning Process") in which a user may execute the Provisioning Technology on a machine ("Target Machine") with the intent of installing, extending or updating the functionality of an Eclipse-based product.
2. During the Provisioning Process, the Provisioning Technology may cause third party Installable Software or a portion thereof to be accessed and copied to the Target Machine.
3. Pursuant to the Specification, you will provide to the user the terms and conditions that govern the use of the Installable Software ("Installable Software Agreement") and such Installable Software Agreement shall be accessed from the Target Machine in accordance with the Specification. Such Installable Software Agreement must inform the user of the terms and conditions that govern the Installable Software and must solicit acceptance by the end user in the manner prescribed in such Installable Software Agreement. Upon such indication of agreement by the user, the provisioning Technology will complete installation of the Installable Software.

### Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

<small>Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries, or both.</small>

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or ,Your,) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, ,You, includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, ,control, means (a) the power, direct or indirect, to cause
the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.
3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients’ rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.
4.1. New Versions.
Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.
6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software, (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and commercial computer software documentation, as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising,
directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works” shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution” shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor” shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Open Source Used In DNAC Platform Antelope 552
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

/*
 * Copyright 2001-2004 Unicode, Inc.
 *
 * Disclaimer
 *
 * This source code is provided as is by Unicode, Inc. No claims are made as to fitness for any particular purpose. No warranties of any kind are expressed or implied. The recipient agrees to determine applicability of information provided. If this file has been purchased on magnetic or optical media from Unicode, Inc., the sole remedy for any claim will be exchange of defective media within 90 days of receipt.
 *
 * Limitations on Rights to Redistribute This Code
 *
 * Unicode, Inc. hereby grants the right to freely use the information supplied in this file in the creation of products supporting the Unicode Standard, and to make copies of this file in any form for internal or external distribution as long as this notice remains attached.
 */

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/
Some code in core/src/java/org/apache/lucene/util/automaton was derived from Brics automaton sources available at www.brics.dk/automaton/. Here is the copyright from those sources:

/*
 * Copyright (c) 2001-2009 Anders Moeller
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 *    derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton were automatically generated with the moman/finenight FSA package. Here is the copyright for those sources:

# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
 * Copyright (C) 1999-2010, International Business Machines
 * Corporation and others. All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, and/or sell copies of the
 * Software, and to permit persons to whom the Software is furnished to do so,
 * provided that the above copyright notice(s) and this permission notice appear
 * in all copies of the Software and that both the above copyright notice(s) and
 * this permission notice appear in supporting documentation.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
 * IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
 * LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
 * ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
 * IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
 * OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * Except as contained in this notice, the name of a copyright holder shall not
 * be used in advertising or otherwise to promote the sale, use or other
 * dealings in this Software without prior written authorization of the
 * copyright holder.
 */

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and
are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

---

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Salon, Wodzimierz Gruszczyski, Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Morfologik

VERSION: 2.1 PoliMorf
BUILD: 2016-02-13 19:37:50+01:00
GIT: 6e63b53

Copyright (c) 2016, Marcin Mikowski
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Ivy (TM)
Copyright 2007-2013 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Portions of Ivy were originally developed by
Jayasoft SARL (http://www.jayasoft.fr/)
and are licensed to the Apache Software Foundation under the "Software Grant License Agreement"

SSH and SFTP support is provided by the JCraft JSch package,
which is open source software, available under the terms of a BSD style license.
The original software and related information is available

========================================================================
==  NOTICE file corresponding to section 4(d) of the Apache License,   ==
==  Version 2.0, in this case for the Apache Xerces Java distribution. ==
========================================================================

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

UIMA Base: uimaj-core
Copyright 2006-2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of Apache UIMA were originally developed by
International Business Machines Corporation and are
licensed to the Apache Software Foundation under the
"Software Grant License Agreement", informally known as the
"IBM UIMA License Agreement".
Copyright (c) 2003, 2006 IBM Corporation.
javax.servlet-*.jar is under the CDDL license, the original source
code for this can be found at http://www.eclipse.org/jetty/downloads.php
Apache Commons Compress
Copyright 2002-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
morfologik-ukrainian-search is a POS tag dictionary in morfologik format adjusted for searching.
It’s part of dict_uk project (https://github.com/brown-uk/dict_uk)

Note: to better fit into full-text search model this dictionary has all word forms in lower case but keeps lemmas for
proper nouns in upper case.

Licensed under Apache License 2.0.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all...
other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
BSD-licensed dictionary of Polish (Morfologik)

VERSION: 2.1 PoliMorf
BUILD: 2016-02-13 19:37:50+01:00
GIT: 6e63b53

Copyright (c) 2016, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2004-2008 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Commons Codec
Copyright 2002-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

---------------------------------------------
src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains
test data from http://aspell.sourceforge.net/test/batch0.tab.

Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org). Verbatim copying
and distribution of this entire article is permitted in any medium,
provided this notice is preserved.
---------------------------------------------

Apache Commons Logging
Copyright 2003-2013 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Morfologik

VERSION: 2.1 PoliMorf
BUILD: 2016-02-13 19:37:51+01:00
GIT: 6e63b53

Copyright (c) 2016, Marcin Mikowski
Wszelkie prawa zastrzezone
Redystrybucja i uywanie, czy to w formie kodu rdowego, czy w formie kodu wykonawczego, s dozwolone pod warunkiem spenienia ponizszych warunkw:

1. Redystrybucja kodu rdowego musi zawiera powysz not copyrightow, niniejsz list warunkw oraz ponsie owiadczenie o wyczeniu odpowiedzialnoci.
2. Redystrybucja kodu wykonawczego musi zawiera powysz not copyrightow, niniejsz list warunkw oraz ponsie owiadczenie o wyczeniu odpowiedzialnoci w dokumentacji i/ lub w innych materiaach dostarczanych wraz z kopi oprogramowania.

TO OPROGRAMOWANIE JEST DOSTARCZONE PRZEZ <POSIADACZA PRAW AUTORSKICH> TAKIM, JAKIE JEST. KADA, DOROZUMIANA LUB BEZPOREDNIO WYRAONA GWARANCJA, NIE WYCZAJC DOROZUMIANEJ GWARANCJI PRZYDATNOCI HANDLOWEJ I PRZYDATNOCI DO OKRELONEGO ZASTOSOWANIA, JEST WYCZONA. W ADNYM WYPADKU <POSIADACZE PRAW AUTORSKICH> NIE MOG BY ODPOWIEDZIALNI ZA JAKIEKOLWIEK BEZPOREDNIE, INCidentalNE, SPECJALNE, UBoczNE I WTRNE SZKODY (NIE WYCZAJC OBOWIZKU DOSTARCZENIA PRODUKTU ZASTPCZEGO LUB SERWISU, ODPOWIEDZIALNOCI Z TYTUU UTRATY WALORW UYTKOWYCH, UTRATY DANYCH LUB KORZYCI, A TAKE PRZERW W PRACY PRZEDSIBIORSTWA) SPOWODOWANE W JAKIKOLWIEK SPOSB I NA PODSTAWIE ISTniejCEJ W TEORII ODPOWIEDZIALNOCI KONTRAKTOWEJ, CAKOWITEJ LUB DELIKTOWEJ (WYNIKEJ ZARWNO Z NIEDBALSTWA JAK INNYCH POSTACI WINY), POWSTAE W JAKIKOLWIEK SPOSB W WYNIKU UYWANIA LUB MAJCE ZWIZEK Z UYWANIEM OPROGRAMOWANIA, NAWET JELI O MALOWICI POWSTANIA TAKICH SZKD OSTRZEONO.

== NOTICE file corresponding to the section 4 d of ==
== the Apache License, Version 2.0, ==
== in this case for the Apache Ant distribution. ==

Apache Ant
Copyright 1999-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

This product includes also software developed by:
- the W3C consortium (http://www.w3c.org).
- the SAX project (http://www.saxproject.org)

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

Apache OpenNLP Maxent
Copyright 2013 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

UIMA Annotator: WhitespaceTokenizer
Copyright 2006-2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache OpenNLP Tools
Copyright 2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Miller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
   analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
   analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
   http://snowball.tartarus.org/

The KStem stemmer in
   analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
   analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
   analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
The Stempel analyzer (stempel) includes BSD-licensed software developed by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil, and Edmond Nolan.

The Polish analyzer (stempel) comes with a default stopword list that is BSD-licensed created by the Carrot2 project. The file resides in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt. See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/) is derived from Unicode data such as the Unicode Character Database. See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary (http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP) (http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===========================================================================
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz
http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

mecab-ipadic-2.7.0-20070801 Notice

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

===========================================================================
Nori Korean Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from

mecab-ko-dic-2.0.3-20170922

which can be obtained from

https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.0.3-20170922.tar.gz

3Y
Z[;]
(^
(\`a
b
cедф
gh
ij)
kl-m.n/o0p1q2r3s4t5u6v7wxSETUID
PERMISSIONInnerClassesHL.org/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;
SETGUISTICKYUSER_READ
USER_WRITEUSER_EXEC
GROUP_READGROUP_WRITE
GROUP_EXEC
This product includes data from BSD-licensed dictionary of Polish (Morfologik, PoliMorf)
(http://morfologik.blogspot.com/)
CyberNeko HTML Parser
(C) Copyright 2002-2009, Andy Clark, Marc Guillemot. All rights reserved.

NekoHTML is a simple HTML scanner and tag balancer that enables application programmers to parse HTML documents and access the information using standard XML interfaces. The parser can scan HTML files and "fix up" many common mistakes that human (and computer) authors make in writing HTML documents. NekoHTML adds missing parent elements; automatically closes elements with optional end tags; and can handle mismatched inline element tags.

NekoHTML is written using the Xerces Native Interface (XNI) that is the foundation of the Xerces2 implementation. This enables you to use the NekoHTML parser with existing XNI tools without modification or rewriting code.

Copyright (c) 2012 France Tlcom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions
Open Source Used In DNAC Platform Antelope  573

are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

" Licensor " shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

" Legal Entity " shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

" You " (or " Your ") shall mean an individual or Legal Entity exercising permissions granted by this License.

" Source " form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

" Object " form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

" Work " shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

" Derivative Works " shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

License for JCraft JSch package

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT,
INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for jQuery
Copyright (c) 2007 John Resig, http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

========================================================================
==  SLF4J Notice -- http://www.slf4j.org/license.html                  ==
========================================================================

Copyright (c) 2004-2008 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache HttpComponents Client
Copyright 1999-2011 The Apache Software Foundation
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net
This product includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski
(http://morfologik.blogspot.com/).
JUnit (under lib/junit-4.10.jar) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

RandomizedRunner, a JUnit @Runner for randomized tests (and more)
Copyright 2011-2012 Carrot Search s.c.
http://labs.carrotsearch.com/randomizedtesting.html

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes asm (asmlib), BSD license
This product includes Google Guava, ASL license
This product includes simple-xml, ASL license
This product includes Google GSON, ASL license

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a ”NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.115 jersey-server 2.25.1
1.115.1 Available under license:
   CDDL+GPL License
   http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.116 readable-stream 2.3.6 2.3.6
1.116.1 Available under license:
   Node.js is licensed for use as follows:

   ---

   Copyright Node.js contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.117 caffeine 2.7.0
1.118 jnr-posix 1.1.4
1.118.1 Available under license:

jnr-posix is released under a tri CPL/GPL/LGPL license. You can use it, redistribute it and/or modify it under the terms of the license of your choice.

=== Common Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.
Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively “Losses”) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL
If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

=== GPL 2.0

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you

Open Source Used In DNAC Platform Antelope 593
may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author...
to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

== GNU LGPL 2.1

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.
Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY
EXPRESSION IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

1.119 global-dirs 2.1.0
1.119.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.120 htrace-core 3.2.0-incubating

1.120.1 Available under license:

htrace-core
Copyright 2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.121 psl 1.8.0 1.8.0

1.121.1 Available under license:

The MIT License (MIT)

Copyright (c) 2017 Lupo Montero lupomontero@gmail.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.122 node-fetch 2.6.1 2.6.1

1.122.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 David Frank

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.123 commons-email 1.5

1.123.1 Available under license :

Apache Commons Email
Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.124 wayland 1.12.0-1~ubuntu16.04.3

1.124.1 Available under license:
Copyright 2008-2012 Kristian Hgsberg
Copyright 2010-2012 Intel Corporation
Copyright 2011 Benjamin Franzke
Copyright 2012 Collabora, Ltd.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next
paragraph) shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

---

The above is the version of the MIT "Expat" License used by X.org:

http://cgit.freedesktop.org/xorg/xserver/tree/COPYING

1.125 expand-brackets 0.1.5
1.125.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.126 odata-parser 1.4.1

1.126.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Auth0, Inc. <support@auth0.com> (http://auth0.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
1.127 string-replace-async 2.0.0
1.127.1 Available under license :

The MIT License (MIT)

Copyright (c) Dmitrii Sobolev <disobolev@icloud.com> (github.com/dsblv)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.128 finalhandler 0.4.1 0.4.1
1.128.1 Available under license :

(The MIT License)

Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingsdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
1.129 composite-error 0.1.1 0.1.1

1.129.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Nick Uraltsev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.130 hadoop-core 1.2.1

1.130.1 Available under license:
JZlib 0.0.* were released under the GNU LGPL license. Later, we have switched over to a BSD-style license.

------------------------------------------------------------------------------
Copyright (c) 2000,2001,2002,2003 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

________________________________________________________________________

Copyright (c) 2000,2001,2002,2003 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

${project.name} is an effort undergoing incubation at the Apache Software Foundation (ASF), sponsored by the Apache Incubator PMC.

Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects.

While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.
Copyright (c) 2000-2005 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2003-2008, Objenesis Team and all contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
OPEN SOURCE USED IN DNAC PLATFORM ANTELOPE

/* Copyright (c) 1995-2000, The Hypersonic SQL Group. */
/* All rights reserved. */
/* Redistribution and use in source and binary forms, with or without */
/* modification, are permitted provided that the following conditions are met: */
/* */
/* Redistributions of source code must retain the above copyright notice, this */
/* list of conditions and the following disclaimer. */
/* */
/* Redistributions in binary form must reproduce the above copyright notice, */
/* this list of conditions and the following disclaimer in the documentation */
/* and/or other materials provided with the distribution. */
/* */
/* Neither the name of the Hypersonic SQL Group nor the names of its */
/* contributors may be used to endorse or promote products derived from this */
/* software without specific prior written permission. */
/* */
/* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" */
/* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE */
/* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE */
/* ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, */
/* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, */
/* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, */
/* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; */
/* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND */
/* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT */
/* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS */
/* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. */
/* */
/* This software consists of voluntary contributions made by many individuals */
/* on behalf of the Hypersonic SQL Group. */
/* */
/* */
/* For work added by the HSQL Development Group: */
/* */
/* Copyright (c) 2001-2004, The HSQL Development Group */
/* All rights reserved. */
/* */
/* Redistribution and use in source and binary forms, with or without */
/* modification, are permitted provided that the following conditions are met: */
/* */
/* Redistributions of source code must retain the above copyright notice, this */
/* list of conditions and the following disclaimer. */
/* */
/* Redistributions in binary form must reproduce the above copyright notice, */
/* this list of conditions and the following disclaimer in the documentation */
/* and/or other materials provided with the distribution. */
/* */
* Neither the name of the HSQL Development Group nor the names of its
* contributors may be used to endorse or promote products derived from this
* software without specific prior written permission.
* 
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG,
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a ”NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the distribution.
* - Neither the name of the University Catholique de Louvain - UCL
* nor the names of its contributors may be used to endorse or
* promote products derived from this software without specific prior
* written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
* ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Copyright (c) 2000-2007, jMock.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of jMock nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
Mockito license - MIT.

Libraries used:

Cglib - Apache License 2.0
ASM - BSD license

Mockito all distribution:

Objenesis - MIT license
Hamcrest - BSD license
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form,
including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and

2. You must cause any modified files to carry prominent notices stating that You changed the files; and

3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent,
trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within the NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

------------------------------------------------------------------------------
License for JCraft JSch package
------------------------------------------------------------------------------
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

------------------------------------------------------------------------------
License for jQuery
------------------------------------------------------------------------------
Copyright (c) 2007 John Resig, http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Thrift Software License
Copyright (c) 2006- Facebook, Inc.

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2004-2007 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, 
distribute, and/or sell copies of the Software, and to permit persons 
to whom the Software is furnished to do so, provided that the above 
copyright notice(s) and this permission notice appear in all copies of 
the Software and that both the above copyright notice(s) and this 
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT 
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR 
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY 
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER 
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF 
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN 
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder 
shall not be used in advertising or otherwise to promote the sale, use 
or other dealings in this Software without prior written authorization 
of the copyright holder.

Apache Ivy 
Copyright 2007-2009 The Apache Software Foundation

This product includes software developed by 
The Apache Software Foundation (http://www.apache.org/).

Portions of Ivy were originally developed by 
Jayasoft SARL (http://www.jayasoft.fr/) 
and are licensed to the Apache Software Foundation under the 
"Software Grant License Agreement"

SSH and SFTP support is provided by the JCraft JSch package, 
which is open source software, available under 
the terms of a BSD style license. 
The original software and related information is available 
This product includes software developed by The Apache Software 
Foundation (http://www.apache.org/).
OGNL is the creation of Luke Blanshard and Drew Davidson.

Copyright 1997-2003, Drew Davidson and Luke Blanshard. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the 
following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following 
disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the Drew Davidson nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

BSD License
Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all ...)
other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.131 batik-squiggle 1.13

1.131.1 Available under license :

Apache License

Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to
You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: HOW TO APPLY THE APACHE LICENSE TO YOUR WORK

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.132 rimraf 3.0.2
1.132.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.133 drill-shaded-guava 28.2-jre

1.134 lodash-difference 4.5.0
1.134.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.135 strip-ansi 3.0.1

1.135.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
1.136 @babel/helper-builder-binary-assignment-operator-visitor 7.12.13

1.136.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.137 batik-awt-util 1.13

1.137.1 Available under license:
Apache Batik

Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for Standardization for the definition of character entities used in the software's
This product includes images from the Tango Desktop Project (http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme (http://www.jesusda.com/projects/pasodoble).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.138 jsreport-browser-client-dist 1.3.0
1.138.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.139 require_optional 1.0.1
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.140 stax 4.2
1.140.1 Available under license:
This copy of Stax2 API is licensed under the
Simplified BSD License (also known as "2-clause BSD", or "FreeBSD License")
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.opensource.org/licenses/bsd-license.php

with details of:

<COPYRIGHT HOLDER> = FasterXML.com
<YEAR> = 2010-

1.141 micromatch 2.3.11
1.141.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.142 flux 3.1.3 3.1.3

1.142.1 Available under license :
BSD License

For Flux software
Copyright (c) 2014-present, Facebook, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name Facebook nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.143 amqplib-tutorials 0.0.1
1.143.1 Available under license:

amqplib copyright (c) 2013, 2014 Michael Bridgen <mikeb@squaremobius.net>

This package, "amqplib", is licensed under the MIT License. A copy may be found in the file LICENSE-MIT in this directory, or downloaded from http://opensource.org/licenses/MIT.
The MIT License (MIT)

Copyright (c) 2013, 2014 Michael Bridgen <mikeb@squaremobius.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.144 engine.io-client 1.8.5 1.8.5

1.144.1 Available under license:

(The MIT License)

Copyright (c) 2014-2015 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.145 regenerator-transform 0.10.1 0.10.1

1.145.1 Available under license:

```
/**
 * Copyright (c) 2014, Facebook, Inc.
 * All rights reserved.
 *
 * This source code is licensed under the BSD-style license found in the
 * https://raw.github.com/facebook/regenerator/master/LICENSE file. An
 * additional grant of patent rights can be found in the PATENTS file in
 * the same directory.
 */

MIT License

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

1.146 asm-util 7.3.1
1.146.1 Available under license:

No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/CheckMethodAdapter.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/TraceFieldVisitor.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/Printer.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/CheckClassAdapter.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/CheckSignatureAdapter.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/CheckModuleAdapter.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/TraceModuleVisitor.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/TraceAnnotationVisitor.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/CheckAnnotationAdapter.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/Textifier.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/TraceRecordComponentVisitor.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/TraceMethodVisitor.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/TraceAnnotationVisitor.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/CheckFieldAdapter.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/TraceClassVisitor.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
  jar/org/objectweb/asm/util/CheckRecordComponentAdapter.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
2011 INRIA, France Telecom
* All rights reserved.

* Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
* 1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
jar/org/objectweb/asm/util/package.html
No license file was found, but licenses were detected in source scan.

/**
* ASM: a very small and fast Java bytecode manipulation framework Copyright (c) 2000-2011 INRIA,
* France Telecom All rights reserved.
*
* <p>Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: 1. Redistributions of source code must retain the
above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions
in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its contributors may be used to
endorse or promote products derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND
* FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY,
* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY
* WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
jar/org/objectweb/asm/util/ASMifierSupport.java
* /opt/cola/permits/1143533479_1615689073.97/0/asm-util-7-3-1-sources-1-
jar/org/objectweb/asm/util/TextifierSupport.java

1.147 @babel/highlight 7.13.10

1.147.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.148 core-js 2.6.11

1.148.1 Available under license:

Copyright (c) 2014-2019 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.149 statuses 1.2.1

1.149.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.150 j2objc-annotations 1.1
1.150.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications
or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions
stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for
inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any
additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of
any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product
names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and
reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work
(and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR
PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and
assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or
otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing,
shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or
consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of
such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may
choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.151 bytes 0.2.1

1.151.1 Available under license :

No license file was found, but licenses were detected in source scan.

(The MIT License)
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
1.152 lodash.escaperegexp 4.1.2

1.152.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/
Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.153 curator-x-discovery 5.1.0
1.153.1 Available under license:

Curator Service Discovery
Copyright 2011-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.154 jersey-media-multipart 2.5.1
1.154.1 Available under license :
CDDL+GPL License
http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.155 source-map-support 0.5.19
1.155.1 Available under license :
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.156 strip-bom-buf 2.0.0

1.156.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.157 object-keys 0.4.0

1.157.1 Available under license :
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.158 utils-merge 1.0.1
1.158.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013-2017 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.159 babel-generator 6.26.1 6.26.1
1.159.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
1.160 babel-plugin-transform-strict-mode

6.24.1 6.24.1

1.160.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.161 okhttp 4.8.0

1.161.1 Available under license:
Note that publicsuffixes.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
https://mozilla.org/MPL/2.0/
1.162 jsesc 2.5.2

1.162.1 Available under license:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.163 dashify 0.2.2 0.2.2

1.163.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.644 apache-log4j 2.14.0
1.644.1 Available under license:
Apache Log4j Core
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.165 xmlbuilder 10.1.1
1.165.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013 Ozgur Ozcitak

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.166 @babel/plugin-syntax-dynamic-import

7.8.3

1.166.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.167 okhttp 4.4.0

1.167.1 Available under license:

Note that publicsuffixes.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
https://mozilla.org/MPL/2.0/

1.168 parquet-jackson 1.10.0
1.168.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This product depends on Jackson and includes it in this binary artifact. Jackson is a high-performance JSON processor.

Copyright: 2007-2015 Tatu Saloranta and other contributors
Home page: http://jackson.codehaus.org/
License: http://www.apache.org/licenses/LICENSE-2.0.txt

1.169 node-signpdf 1.3.2
1.169.1 Available under license:

**MIT License**

Copyright (c) 2014 Devon Govett

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2019 The Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.170 buffer-alloc-unsafe 1.1.0

1.170.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.171 ansi-styles 1.0.0 1.0.0

1.171.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.172 verror 1.10.0 1.10.0
1.172.1 Available under license:
Copyright (c) 2016, Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

1.173 groovy-groovydoc 3.0.8
1.173.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1218304752_1634925093.2/0/groovy-groovydoc-3-0-8-sources-
jar/org/codehaus/groovy/groovydoc/GroovyTag.java
* /opt/cola/permits/1218304752_1634925093.2/0/groovy-groovydoc-3-0-8-sources-
jar/org/codehaus/groovy/tools/groovydoc/ClasspathResourceManager.java
* /opt/cola/permits/1218304752_1634925093.2/0/groovy-groovydoc-3-0-8-sources-
jar/org/codehaus/groovy/tools/groovydoc/FileSystemResourceManager.java
* /opt/cola/permits/1218304752_1634925093.2/0/groovy-groovydoc-3-0-8-sources-
jar/org/codehaus/groovy/tools/groovydoc/SimpleGroovyClassDocAssembler.java
No license file was found, but licenses were detected in source scan.

/*! normalize.css v2.1.0 | MIT License | git.io/normalize */

Found in path(s):
* /opt/cola/permits/1218304752_1634925093.2/0/groovy-groovydoc-3-0-8-sources-
jar/org/codehaus/groovy/tools/groovydoc/gstringTemplates/topLevel/stylesheet.css
No license file was found, but licenses were detected in source scan.

# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# software distributed under the License is distributed on an

Found in path(s):
* /opt/cola/permits/1218304752_1634925093.2/0/groovy-groovydoc-3-0-8-sources-
jar/org/codehaus/groovy/tools/groovydoc/Main.properties
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.

Found in path(s):
* /opt/cola/permits/1218304752_1634925093.2/0/groovy-groovydoc-3-0-8-sources-
jar/org/codehaus/groovy/tools/groovydoc/gstringTemplates/topLevel/allclasses-frame.html
1.174 kerby-util 1.0.0

1.174.1 Available under license:

Kerby Util
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
1.175 mongodb 3.6.6 3.6.6

1.175.1 Available under license :
MIT License

Copyright (c) Microsoft Corporation.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

1.176 assertion-error 1.1.0

1.176.1 Available under license :
No license file was found, but licenses were detected in source scan.

# AssertionError ![Build Status](https://travis-ci.org/chaijs/assertion-error.png?branch=master)

> Error constructor for test and validation frameworks that implements standardized AssertionError specification.

## Installation

### Node.js

`assertion-error` is available on [npm](http://npmjs.org).

```
$ npm install assertion-error
```

### Component
`assertion-error` is available as a [component](https://github.com/component/component).

```bash
$ component install chaijs/assertion-error
```

## License

(The MIT License)

Copyright (c) 2013 Jake Luer <jake@qualiancy.com> (http://qualiancy.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):
* /opt/cola/permits/1112206225_1607050024.11/0/chaijs-assertion-error-1-1-0-0-gfaa3f8c-1-tar-gz/chaijs-assertion-error-faa3f8c/README.md

No license file was found, but licenses were detected in source scan.

```javascript
/*! 
 * assertion-error
 * Copyright(c) 2013 Jake Luer <jake@qualiancy.com>
 * MIT Licensed
 */

Found in path(s):
* /opt/cola/permits/1112206225_1607050024.11/0/chaijs-assertion-error-1-1-0-0-gfaa3f8c-1-tar-gz/chaijs-assertion-error-faa3f8c/index.js

1.177 jetty-security 9.3.27.v20190418
1.178.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.179 drill-yarn 1.19.0
1.179.1 Available under license :

Drill : On-YARN
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.180 seamless-immutable 7.1.3 7.1.3

1.180.1 Available under license:

Copyright (c) 2016, Richard Feldman
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of seamless-immutable nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.


1.181.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.182 hadoop-yarn-client 3.2.2

1.182.1 Available under license:

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of Gson 2.2.4,
which has the following notices:

The Netty Project

Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

```
* LICENSE:
* license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java
```

This product contains a modified portion of `jbzip2`, a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

```
* LICENSE:
* license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/
```

This product contains a modified portion of `libdivsufsort`, a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

```
* LICENSE:
* license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/
```

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

```
* LICENSE:
* license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools
```

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

```
* LICENSE:
* license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/
```

This product optionally depends on 'Compress-LZF', a Java library for encoding and
decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression
and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression
and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression
and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data
interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate
a temporary self-signed X.509 certificate when the JVM does not provide the
equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/
This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:  
  * license/LICENSE.snappy.txt (New BSD License)  
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:  
  * license/COPYING (Apache License 2.0)  
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:  
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)  
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:  
  * license/LICENSE.caliper.txt (Apache License 2.0)  
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:  
  * license/LICENSE.commons-logging.txt (Apache License 2.0)  
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:  
  * license/LICENSE.log4j.txt (Apache License 2.0)  
  * http://logging.apache.org/log4j/
This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===============================================================================
The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.
Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:

* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())
The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices:

* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell

Dumbster SMTP test server
Copyright 2004 Jason Paul Kitchen

TypeUtil.java
Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of "Java Concurrency in Practice" book annotations 1.0, which has the following notices:

* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)

Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.


which has the following notices:

* ==============================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
  ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License
Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLx2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html
org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida,
modified April 2001 by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt
for non-commercial or commercial purposes and without fee is
granted provided that the copyright notice appears in all copies.

The binary distribution of this product bundles binaries of
Snappy for Java 1.0.4.1,
which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
(Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by
"GCC Runtime Library Exception"

== Contributors ==
* Tatu Saloranta
  * Providing benchmark suite
* Alec Wysoker
  * Performance and memory usage improvement

The binary distribution of this product bundles binaries of
Xerces2 Java Parser 2.9.1,
which has the following notices:
* ===========================================================================
  NOTICE file corresponding to section 4(d) of the Apache License, Version 2.0,
  in this case for the Apache Xerces Java distribution. ==
  ===========================================================================

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

The binary distribution of this product bundles binaries of
Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0

--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting
it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was
relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.
Open Source Used In DNAC Platform Antelope

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of
Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library
distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5
(https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-
commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1,
which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9
which has the following notices:
* =============================================================================
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
================================================================================
This product includes software developed by
Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in
 *   the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL
 *   nor the names of its contributors may be used to endorse or
 *   promote products derived from this software without specific prior
 *   written permission.
 * *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
 * ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/ * + +
 * *
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 * *
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 * *
 * Abstract: The main routine
 * *
 * - - */
LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :
- LZ4 source repository : http://code.google.com/p/lz4/
- LZ4 public forum : https://groups.google.com/forum/#!forum/lz4c
*/

For hadoop-common-project/hadoop-common/src/main/native/gtest

--------------------------------------------------------------------------------
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1

ENO LICENSE

This is a work derived from Russ Cox's RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni (https://github.com/fusesource/leveldbjni), which is available under the following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of FuseSource Corp. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

=================

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/
--------------------------------------------------------------------------------

Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js
--------------------------------------------------------------------------------
Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
em-helpers v0.5.13
em-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License
Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery
- Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
- Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- oj! Algorithms - version 43.0

jQuery Foundation and other contributors, https://jquery.org/

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

D3 is available under a 3-clause BSD license. For details, see:
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
- HSQLDB Database 2.3.4

(HSQL License)

"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ""AS IS"" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.”

The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.
2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your
Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in
Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a
Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to
make it enforceable. This License shall be governed by the law of the
jurisdiction specified in a notice contained within the Original Software
(except to the extent applicable law, if any, provides otherwise), excluding
such jurisdictions conflict-of-law provisions. Any litigation relating to this
License shall be subject to the jurisdiction of the courts located in the
jurisdiction and venue specified in a notice contained within the Original
Software, with the losing party responsible for costs, including, without
limitation, court costs and reasonable attorneys fees and expenses. The
application of the United Nations Convention on Contracts for the International
Sale of Goods is expressly excluded. Any law or regulation which provides that
the language of a contract shall be construed against the drafter shall not
apply to this License. You agree that You alone are responsible for compliance
with the United States export administration regulations (and the export control
laws and regulation of any other countries) when You use, distribute or
otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is
responsible for claims and damages arising, directly or indirectly, out of its
utilization of rights under this License and You agree to work with Initial
Developer and Contributors to distribute such responsibility on an equitable
basis. Nothing herein is intended or shall be deemed to constitute any admission
of liability.

The binary distribution of this product bundles these dependencies under the
following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
grizzly-http 2.2.21
grizzly-http-server 2.2.21
grizzly-http-servlet 2.2.21
grizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.
1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to
third party intellectual property claims, the Initial Developer hereby grants
You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual
property rights (other than patent or trademark) Licensable by Initial
Developer, to use, reproduce, modify, display, perform, sublicense and
distribute the Original Software (or portions thereof), with or without
Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims
infringed by the making, using or selling of Original Software, to make, have
made, use, practice, sell, and offer for sale, and/or otherwise dispose of the
Original Software (or portions thereof).
(c) The licenses granted in Sections
2.1(a) and (b) are effective on the date Initial Developer first distributes or
otherwise makes the Original Software available to a third party under the terms
of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is
granted: (1) for code that You delete from the Original Software, or (2) for
infringements caused by: (i) the modification of the Original Software, or (ii)
the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and
subject to third party intellectual property claims, each Contributor hereby
grants You a world-wide, royalty-free, non-exclusive license:
(a) under
intellectual property rights (other than patent or trademark) Licensable by
Contributor to use, reproduce, modify, display, perform, sublicense and
distribute the Modifications created by such Contributor (or portions thereof),
either on an unmodified basis, with other Modifications, as Covered Software
and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the
making, using, or selling of Modifications made by that Contributor either alone
and/or in combination with its Contributor Version (or portions of such
combination), to make, use, sell, offer for sale, have made, and/or otherwise
dispose of: (1) Modifications made by that Contributor (or portions thereof); and
(2) the combination of Modifications made by that Contributor with its
Contributor Version (or portions of such combination).
(c) The licenses granted
in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications available to a third
party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2)
for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with
other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every
Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.
6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR
LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
    copyright notice, this list of conditions and the following disclaimer
    in the documentation and/or other materials provided with the
    distribution.
  * Neither the name of Google Inc. nor the names of its
    contributors may be used to endorse or promote products derived from
    this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

---------------------------------------------------------------------
By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and
fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for
damages, including direct, indirect, special, incidental and consequential
damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by
that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor,
and informs licensees how to obtain it in a reasonable manner on or through a
medium customarily used for software exchange.
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the
Program.

Each Contributor must identify itself as the originator of its Contribution, if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore, if
a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses, damages
and costs (collectively "Losses") arising from claims, lawsuits and other legal
actions brought by a third party against the Indemnified Contributor to the
extent caused by the acts or omissions of such Commercial Contributor in
connection with its distribution of the Program in a commercial product
offering. The obligations in this section do not apply to any claims or Losses
relating to any actual or alleged intellectual property infringement. In order
to qualify, an Indemnified Contributor must: a) promptly notify the Commercial
Contributor in writing of such claim, and b) allow the Commercial Contributor to
control, and cooperate with the Commercial Contributor in, the defense and any
related settlement negotiations. The Indemnified Contributor may participate in
any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product
offering, Product X. That Contributor is then a Commercial Contributor. If that
Commercial Contributor then makes performance claims, or offers warranties
related to Product X, those performance claims and warranties are such
Commercial Contributor's responsibility alone. Under this section, the
Commercial Contributor would have to defend claims against the other
Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:

JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the
following license:

FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.
--------------------------------------------------------------------------------
(2-clause BSD)
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those
of the authors and should not be interpreted as representing official policies,
either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the
following license:

"Java Concurrency in Practice" book annotations 1.0
--------------------------------------------------------------------------------
(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS
PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR
OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and
perform publicly by means of a digital audio transmission the Work including as
incorporated in Collective Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and
perform publicly by means of a digital audio transmission Derivative Works.
For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive
right to collect, whether individually or via a performance rights society (e.g.
ASCAP, BMI, SESAC), royalties for the public performance or public digital
performance (e.g. webcast) of the Work.
Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right
to collect, whether individually or via a music rights agency or designated
agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the
Work ("cover version") and distribute, subject to the compulsory license created
by 17 USC Section 115 of the US Copyright Act (or the equivalent in other
jurisdictions).
Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the
Work is a sound recording, Licensor waives the exclusive right to collect,
whether individually or via a performance-rights society (e.g. SoundExchange),
royalties for the public digital performance (e.g. webcast) of the Work, subject
to the compulsory license created by 17 USC Section 114 of the US Copyright Act
(or the equivalent in other jurisdictions).
The above rights may be exercised in all media and formats whether now known or
hereafter devised. The above rights include the right to make such modifications
as are technically necessary to exercise the rights in other media and formats.
All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject
to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally
perform the Work only under the terms of this License, and You must include a
copy of, or the Uniform Resource Identifier for, this License with every copy or
phonorecord of the Work You distribute, publicly display, publicly perform, or
publicly digitally perform. You may not offer or impose any terms on the Work
that alter or restrict the terms of this License or the recipients' exercise of
the rights granted hereunder. You may not sublicense the Work. You must keep
intact all notices that refer to this License and to the disclaimer of
warranties. You may not distribute, publicly display, publicly perform, or
publicly digitally perform the Work with any technological measures that control
access or use of the Work in a manner inconsistent with the terms of this
License Agreement. The above applies to the Work as incorporated in a Collective
Work, but this does not require the Collective Work apart from the Work itself
to be made subject to the terms of this License. If You create a Collective
Work, upon notice from any Licensor You must, to the extent practicable, remove
from the Collective Work any credit as required by clause 4(b), as requested. If
You create a Derivative Work, upon notice from any Licensor You must, to the
extent practicable, remove from the Derivative Work any credit as required by
clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license
terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

jamon-runtime 2.4.1

(MPL 2.0)

Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributors Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.
1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice
in Exhibit A, the Executable Form of such Source Code Form, and Modifications of
such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to
the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or
earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate
file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the
time of the initial grant or subsequently, any and all of the rights conveyed by
this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or
modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and
apparatus claims, in any patent Licensable by such Contributor that would be
infringed, but for the grant of the License, by the making, using, selling,
offering for sale, having made, import, or transfer of either its Contributions
or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General
Public License, Version 2.1, the GNU Affero General Public License, Version 3.0,
or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For
legal entities, You includes any entity that controls, is controlled by,
or is under common control with You. For purposes of this definition,
control means (a) the power, direct or indirect, to cause the direction or
management of such entity, whether by contract or otherwise, or (b) ownership of
more than fifty percent (50%) of the outstanding shares or beneficial ownership
of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive
license:

under intellectual property rights (other than patent or trademark) Licensable
by such Contributor to use, reproduce, make available, modify, display, perform,
distribute, and otherwise exploit its Contributions, either on an unmodified
basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have
made, import, and otherwise transfer either its Contributions or its Contributor
Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become
effective for each Contribution on the date the Contributor first distributes
such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this
License. No additional rights or licenses will be implied from the distribution
or licensing of Covered Software under this License. Notwithstanding Section
2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third partys
modifications of Covered Software, or (ii) the combination of its Contributions
with other software (except as part of its Contributor Version); or
under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form
how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all
distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any
direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party’s ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this
10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

/*--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

/*

The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN
License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.
2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made
subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution.
in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will
survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under
applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons” or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons’ then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.


Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.183 bufferutil 1.2.1
1.183.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*! 
 * bufferutil: WebSocket buffer utils
 * Copyright(c) 2015 Einar Otto Stangvik <einaros@gmail.com>
 * MIT Licensed
 */

Found in path(s):
* /opt/cola/permits/1127086007_1611783465.5/0/bufferutil-1-2-1-1-tgz/package/src/bufferutil.cc
* /opt/cola/permits/1127086007_1611783465.5/0/bufferutil-1-2-1-1-tgz/package/fallback.js

1.184 jline 2.14.3

1.185 bops 0.1.1
1.185.1 Available under license:
MIT

1.186 yauaa-logparser 5.20
1.186.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*! 
 * Yet Another UserAgent Analyzer
 * Copyright (C) 2013-2020 Niels Basjes
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * https://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
2020 Niels Basjes
~
~ Licensed under the Apache License, Version 2.0 (the "License");
~ you may not use this file except in compliance with the License.
~ You may obtain a copy of the License at
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS,
~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
~ See the License for the specific language governing permissions and
~ limitations under the License.

1.187 retrofit 2.8.1

1.187.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2017 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 Square, Inc.
 */

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/OptionalConverterFactory.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 Square, Inc.
 */

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/Body.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/Retrofit.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/RequestBuilder.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/Converter.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/Callback.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/KotlinExtensions.kt
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/internal/EverythingIsNonNull.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/Invocation.java

No license file was found, but licenses were detected in source scan.

(/
* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/CompletableFutureCallAdapterFactory.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/HeaderMap.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/HttpException.java

No license file was found, but licenses were detected in source scan.

(/
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/ServiceMethod.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/Url.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/BuiltInConverters.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/Response.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/OkHttpCall.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/DefaultCallAdapterFactory.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/Call.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/CallAdapter.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/HttpServiceMethod.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/RequestFactory.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/ParameterHandler.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/OPTIONS.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/FieldMap.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/PartMap.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/SkipCallbackExecutor.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/Tag.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/Multipart.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/Part.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2013 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/Path.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/POST.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/Multipart.java
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/http/Part.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1199767211_1630583092.85/0/retrofit-2-8-1-sources-1-jar/retrofit2/Utils.java

1.188 he 1.2.0
1.188.1 Available under license:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.189 graphlib 2.1.8 2.1.8
1.189.1 Available under license:
Copyright (c) 2012-2014 Chris Pettitt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.190 ms 0.7.2 0.7.2
1.190.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Zeit, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.191 etag 1.8.1

1.191.1 Available under license :

(The MIT License)

Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.192 merge-class-names 1.3.0 1.3.0

1.192.1 Available under license :

MIT License

Copyright (c) 2017-2019 Wojciech Maj
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.193 load-json-file 2.0.0
1.193.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.194 immediate 3.0.6
1.194.1 Available under license:

Copyright (c) 2012 Barnesandnoble.com, llc, Donavon West, Domenic Denicola, Brian Cavalier

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.195 jetty-io 9.4.17.v20190418

1.195.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution ‘originates’
from a Contributor if it was added to the Program by such Contributor
itself or anyone acting on such Contributor's behalf. Contributions do not
include additions to the Program which: (i) are separate modules of
software distributed in conjunction with the Program under their own
license agreement, and (ii) are not derivative works of the Program.

“Contributor” means any person or entity that distributes the Program.

“Licensed Patents” mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when
combined with the Program.

“Program” means the Contributions distributed in accordance with this
Agreement.

“Recipient” means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free copyright license to
reproduce, prepare derivative works of, publicly display, publicly
perform, distribute and sublicense the Contribution of such Contributor,
if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free patent license under
Licensed Patents to make, use, sell, offer to sell, import and otherwise
transfer the Contribution of such Contributor, if any, in source code and
object code form. This patent license shall apply to the combination of
the Contribution and the Program if, at the time the Contribution is
added by the Contributor, such addition of the Contribution causes such
combination to be covered by the Licensed Patents. The patent license
shall not apply to any other combinations which include the Contribution.
No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses
to its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other
intellectual property rights of any other entity. Each Contributor
disclaims any liability to Recipient for claims brought by any other
entity based on infringement of intellectual property rights or
otherwise. As a condition to exercising the rights and licenses granted
hereunder, each Recipient hereby assumes sole responsibility to secure
any other intellectual property rights needed, if any. For example, if a
third party patent license is required to allow Recipient to distribute
the Program, it is Recipient's responsibility to acquire that license.
before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient
copyright rights in its Contribution, if any, to grant the copyright
license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under
its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties
      and conditions, express and implied, including warranties or
      conditions of title and non-infringement, and implied warranties or
      conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for
       damages, including direct, indirect, special, incidental and
       consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are
       offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such
       Contributor, and informs licensees how to obtain it in a reasonable
       manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained
within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore,
if a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and
other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and

b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL
If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

================================================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
================================================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html
* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----
Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----
OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----
Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.196 @babel/helper-validator-option 7.12.17

1.196.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
1.197 object-assign 4.1.0

1.197.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.198 kafka-avro-serializer 6.1.1

1.198.1 Available under license:
Confluent Community License Agreement
Version 1.0

This Confluent Community License Agreement Version 1.0 (the Agreement) sets forth the terms on which Confluent, Inc. (Confluent) makes available certain software made available by Confluent under this Agreement (the Software). BY INSTALLING, DOWNLOADING, ACCESSING, USING OR DISTRIBUTING ANY OF THE SOFTWARE, YOU AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO
1. LICENSE GRANT AND CONDITIONS.

1.1 License. Subject to the terms and conditions of this Agreement, Confluent hereby grants to Licensee a non-exclusive, royalty-free, worldwide, non-transferable, non-sublicenseable license during the term of this Agreement to: (a) use the Software; (b) prepare modifications and derivative works of the Software; (c) distribute the Software (including without limitation in source code or object code form); and (d) reproduce copies of the Software (the License). Licensee is not granted the right to, and Licensee shall not, exercise the License for an Excluded Purpose. For purposes of this Agreement, Excluded Purpose means making available any software-as-a-service, platform-as-a-service, infrastructure-as-a-service or other similar online service that competes with Confluent products or services that provide the Software.

1.2 Conditions. In consideration of the License, Licensee’s distribution of the Software is subject to the following conditions:

(a) Licensee must cause any Software modified by Licensee to carry prominent notices stating that Licensee modified the Software.

(b) On each Software copy, Licensee shall reproduce and not remove or alter all Confluent or third party copyright or other proprietary notices contained in the Software, and Licensee must provide the notice below with each copy.

This software is made available by Confluent, Inc., under the terms of the Confluent Community License Agreement, Version 1.0 located at http://www.confluent.io/confluent-community-license. BY INSTALLING, DOWNLOADING, ACCESSING, USING OR DISTRIBUTING ANY OF THE SOFTWARE, YOU AGREE TO THE TERMS OF SUCH LICENSE AGREEMENT.

1.3 Licensee Modifications. Licensee may add its own copyright notices to modifications made by Licensee and may provide additional or different license terms and conditions for use, reproduction, or distribution of Licensee’s modifications. While redistributing the Software or modifications thereof, Licensee may choose to offer, for a fee or free of charge, support, warranty, indemnity, or other obligations. Licensee, and not Confluent, will be responsible for any such obligations.

1.4 No Sublicensing. The License does not include the right to sublicense the Software, however, each recipient to which Licensee
provides the Software may exercise the Licenses so long as such recipient agrees to the terms and conditions of this Agreement.

2. TERM AND TERMINATION. This Agreement will continue unless and until earlier terminated as set forth herein. If Licensee breaches any of its conditions or obligations under this Agreement, this Agreement will terminate automatically and the License will terminate automatically and permanently.

3. INTELLECTUAL PROPERTY. As between the parties, Confluent will retain all right, title, and interest in the Software, and all intellectual property rights therein. Confluent hereby reserves all rights not expressly granted to Licensee in this Agreement. Confluent hereby reserves all rights in its trademarks and service marks, and no licenses therein are granted in this Agreement.

4. DISCLAIMER. CONFLUENT HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE.

5. LIMITATION OF LIABILITY. CONFLUENT WILL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, LOST PROFITS OR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, INDIRECT, OR DIRECT DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING OUT OF THIS AGREEMENT. THE FOREGOING SHALL APPLY TO THE EXTENT PERMITTED BY APPLICABLE LAW.

6. GENERAL.

6.1 Governing Law. This Agreement will be governed by and interpreted in accordance with the laws of the state of California, without reference to its conflict of laws principles. If Licensee is located within the United States, all disputes arising out of this Agreement are subject to the exclusive jurisdiction of courts located in Santa Clara County, California, USA. If Licensee is located outside of the United States, any dispute, controversy or claim arising out of or relating to this Agreement will be referred to and finally determined by arbitration in accordance with the JAMS International Arbitration Rules. The tribunal will consist of one arbitrator. The place of arbitration will be Palo Alto, California. The language to be used in the arbitral proceedings will be English. Judgment upon the award rendered by the arbitrator may be entered in any court having jurisdiction thereof.

6.2 Assignment. Licensee is not authorized to assign its rights under this Agreement to any third party. Confluent may freely assign its rights under this Agreement to any third party.

6.3 Other. This Agreement is the entire agreement between the parties
regarding the subject matter hereof. No amendment or modification of
this Agreement will be valid or binding upon the parties unless made in
writing and signed by the duly authorized representatives of both
parties. In the event that any provision, including without limitation
any condition, of this Agreement is held to be unenforceable, this
Agreement and all licenses and rights granted hereunder will immediately
terminate. Waiver by Confluent of a breach of any provision of this
Agreement or the failure by Confluent to exercise any right hereunder
will not be construed as a waiver of any subsequent breach of that right
or as a waiver of any other right.

Apache Kafka
Copyright 2016 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This distribution has a binary dependency on jersey, which is available under the CDDL
License. The source code of jersey can be found at https://github.com/jersey/jersey/.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
Java ClassMate library was originally written by Tatu Saloranta (tatu.saloranta@iki.fi)

Other developers who have contributed code are:

* Brian Langel
This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.
You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="Home page of The Apache Software Foundation">
<link rel="apple-touch-icon" sizes="57x57" href="/favicons/apple-touch-icon-57x57.png">
<link rel="apple-touch-icon" sizes="60x60" href="/favicons/apple-touch-icon-60x60.png">
<link rel="apple-touch-icon" sizes="72x72" href="/favicons/apple-touch-icon-72x72.png">
<link rel="apple-touch-icon" sizes="76x76" href="/favicons/apple-touch-icon-76x76.png">
<link rel="apple-touch-icon" sizes="114x114" href="/favicons/apple-touch-icon-114x114.png">
<link rel="apple-touch-icon" sizes="120x120" href="/favicons/apple-touch-icon-120x120.png">
<link rel="apple-touch-icon" sizes="144x144" href="/favicons/apple-touch-icon-144x144.png">
<link rel="apple-touch-icon" sizes="152x152" href="/favicons/apple-touch-icon-152x152.png">
<link rel="apple-touch-icon" sizes="180x180" href="/favicons/apple-touch-icon-180x180.png">
<link rel="icon" type="image/png" href="/favicons/favicon-32x32.png" sizes="32x32">
<link rel="icon" type="image/png" href="/favicons/favicon-194x194.png" sizes="194x194">
<link rel="icon" type="image/png" href="/favicons/favicon-96x96.png" sizes="96x96">
<link rel="icon" type="image/png" href="/favicons/android-chrome-192x192.png" sizes="192x192">
<link rel="icon" type="image/png" href="/favicons/favicon-16x16.png" sizes="16x16">
<link rel="manifest" href="/favicons/manifest.json">
<link rel="shortcut icon" href="/favicons/favicon.ico">
<meta name="msapplication-TileColor" content="#603cba">
<meta name="msapplication-TileImage" content="/favicons/mstile-144x144.png">
<meta name="msapplication-config" content="/favicons/browserconfig.xml">
<meta name="theme-color" content="#303284">
<title>Apache License, Version 2.0</title>
<link href='https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700%7cDroid+Serif:400,700' rel="stylesheet" type="text/css">
<link href="/css/min.bootstrap.css" rel="stylesheet">
<link href="/css/styles.css" rel="stylesheet">

<!-- Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. -->
Apache License
Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You
distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or
redistributing the Work and assume any risks associated with Your exercise of permissions under this License.</p>
<p><strong>8. Limitation of Liability</strong>. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.</p>
<p><strong>9. Accepting Warranty or Additional Liability</strong>. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.</p>

END OF TERMS AND CONDITIONS

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Open Source Used in DNAC Platform Antelope 827
Open Source Used In DNAC Platform Antelope 828
See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head><title>301 Moved Permanently</title></head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://opensource.org/licenses/mit-license.php">here</a>.</p>
</body></html>
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: schema-registry
Source: https://github.com/confluentinc/schema-registry

Files: *
  Copyright: 2018 Confluent, Inc.
  License: Apache-2

Files: core/*
  Copyright: 2015 Confluent, Inc.
  License: Confluent Community License

License: Confluent Community License
Licensed under the Confluent Community License; you may not use this file
except in compliance with the License. You may obtain a copy of the License at

http://www.confluent.io/confluent-community-license
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OF ANY KIND, either express or implied. See the License for the
specific language governing permissions and limitations under the License.

License: Apache-2
Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License. You may obtain a copy of
the License at

http://www.apache.org/licenses/LICENSE-2.0

- Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under
the License.
- On Debian systems, the Apache 2.0 license can be found in
/usr/share/common-licenses/Apache-2.0.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This distribution has a binary dependency on jersey, which is available under the CDDL License as described below.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL - Version 1.1)

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification.
You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.
4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.
COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.
6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY
AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN
INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT
APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S
NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME
JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR
CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of
commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer
software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212
and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered
Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes,
any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under
this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is
held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This
License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software
(except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law
provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the
jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible
for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of
the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or
regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this
License. You agree that You alone are responsible for compliance with the United States export administration
regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise
make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising,
directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer
and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be
deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE
(CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-
law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the
Northern District of California and the state courts of the State of California, with venue lying in Santa Clara
County, California.

Apache Avro
Copyright 2009-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
The following commands were used to generate license and notice files. Replace <VERSION> with
the Schema Registry version, <SRC_PATH> with the path to the Schema Registry source directory,
and <LICENSE_TOOL_PATH> with the path of the license tool.

cd <SRC_PATH>
mvn package -DskipTests
mkdir /tmp/jars
mkdir /tmp/overrides
cd <LICENSE_TOOL_PATH>
./bin/run_license_job.bash -i /tmp/jars -l <SRC_PATH>/licenses -n <SRC_PATH>/notices -h <SRC_PATH>/licenses-and-notices.html -o /tmp/overrides
<!DOCTYPE HTML PUBLIC "://W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html lang="en-US">
<head>
<title>GlassFish Dual License Header and License Notice GPL v2 and CDDL</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<style type="text/css">
li {display: block;}
</style>
</head>

<body leftmargin="0" topmargin="0" marginheight="0" marginwidth="0" bgcolor="#ffffff">

<h3>COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0</h3>

<ul>

<li><p>1. Definitions.</p></li>

<ol>

<li>Contributor means each individual or entity that creates or contributes to the creation of Modifications.</li>

</ol>

</ul>

<p>1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the
Modifications made by that particular Contributor.
</p>
</li>

<li><p>
<b>1.3. &ldquo;Covered Software&rdquo;</b> means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
</p>
</li>

<li><p>
<b>1.4. &ldquo;Executable&rdquo;</b> means the Covered Software in any form other than Source Code.
</p>
</li>

<li><p>
<b>1.5. &ldquo;Initial Developer&rdquo;</b> means the individual or entity that first makes Original Software available under this License.
</p>
</li>

<li><p>
<b>1.6. &ldquo;Larger Work&rdquo;</b> means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
</p>
</li>

<li><p>
<b>1.7. &ldquo;License&rdquo;</b> means this document.
</p>
</li>

<li><p>
<b>1.8. &ldquo;Licensable&rdquo;</b> means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired,
any and all of the rights conveyed herein.
</p>
</li>

<li>
<p>
<b>1.9. “Modifications”</b> means the Source Code and Executable form of any of the following:
</p>
<ul>
<li>
<p>
<b>A.</b> Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
</p>
</li>
<li>
<p>
<b>B.</b> Any new file that contains any part of the Original Software or previous Modification; or
</p>
</li>
<li>
<p>
<b>C.</b> Any new file that is contributed or otherwise made available under the terms of this License.
</p>
</li>
</ul>
</li>

<li>
<p>
<b>1.10. “Original Software”</b> means the Source Code and Executable form of computer software code that is originally released under this License.
</p>
</li>

<li>
<p>
<b>1.11. “Patent Claims”</b> means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
</p>
</li>
1.12. “Source Code” means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. “You” (or “Your”) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, “You” includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform,
 sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
</p>
<li>
<p>
<b>(b)</b> under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
</p>
</li>
<li>
<p>
<b>(c)</b> The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
</p>
</li>
<li>
<p>
<b>(d)</b> Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.
</p>
</li>
<li>
<p>
2.2. Contributor Grant.</p>
</li>
<li>
<p>Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
</p>
</li>
<li>
<p><b>(a)</b> under intellectual property rights (other than patent or trademark) Licensable by
Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

<p><b>(b)</b> under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

<p><b>(c)</b> The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

<p><b>(d)</b> Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.
3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to
offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.
<li>
<p><b>4. Versions of the License.</b></p>
<ul>
<li><p><b>4.1. New Versions.</b></p></li>
<p>Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.</p></li>
<li><p><b>4.2. Effect of New Versions.</b></p></li>
<p>You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.</p></li>
<li><p><b>4.3. Modified Versions.</b></p></li>
<p>When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a)&nbsp;rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b)&nbsp;otherwise make it clear that the license contains terms which differ from this License.</p></li>
</ul>
</li>
5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software.
against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a &ldquo;commercial item,&rdquo; as that term is defined in 48&nbsp;C.F.R.&nbsp;2.101 (Oct. 1995), consisting of &ldquo;commercial computer software&amp;rdquo; (as that term is defined at 48 C.F.R. &sect;&nbsp;252.227-7014(a)(1)) and &ldquo;commercial computer software documentation&amp;rdquo; as such terms are used in 48&nbsp;C.F.R.&nbsp;12.212 (Sept. 1995). Consistent with
48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to
constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.
This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

</p>
</li>
<li>
<p>
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

</p>
</li>
</ul>

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

</p>

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

</p>

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

</p>
</li>
3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise)
that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail
to address new problems or concerns.
</p>
<p>
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
</p>
</li>
<li>
<p>
If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.
</p>
</li>
<li>
<p>
NO WARRANTY
</p>
</li>
<li>
<p>
BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
</p>
</li>
<li>
<p>
IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

</p>
</li>

</i>
<p>END OF TERMS AND CONDITIONS</p>
</li>

</i>
<p>How to Apply These Terms to Your New Programs</p>
</i>
<p>If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.</p>
</i>
<p>To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.</p>
</i>
<ul>
<li>
<p>One line to give the program's name and a brief idea of what it does.</p>
</li>
<li>
<p>Copyright (C) <year> <name of author></p>
</li>
<li>
<p>This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.</p>
</li>
<li>
<p>This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
</p>
</li>
</ul>
<p>This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
</p>
</li>

<li style="background-color:yellow;">
<p><b>"CLASSPATH" EXCEPTION TO THE GPL VERSION 2</b></p>
</li>

<p>Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words<br />
"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."
</p>

<p>Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.
</p>

<p>As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.
</p>
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of any computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which
Open Source Used In DNAC Platform Antelope 868

modifications are made and (b) associated documentation included in or with such code.</blockquote>

1.13. &ldquo;You&rdquo; (or &ldquo;Your&rdquo;) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, &ldquo;You&rdquo; includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, &ldquo;control&rdquo; means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.</p>

2. License Grants.</p>

2.1. The Initial Developer Grant.</p>

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:</p>

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and</p>

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).</p>

The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.</p>

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.</p>

2.2. Contributor Grant.</p>

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:</p>

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and</p>

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).</p>

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.</p>

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.</p>

3. Distribution Obligations.</p>

3.1. Availability of Source Code.</p>

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this
License. You must include a copy of this License with every copy of the Source Code form of the Covered Software you distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.\</blockquote>

\begin{blockquote}
3.2. Modifications.\end{blockquote}

\begin{blockquote}
The Modifications that you create or to which you contribute are governed by the terms of this License. You represent that you believe your Modifications are your original creation(s) and/or you have sufficient rights to grant the rights conveyed by this License.\end{blockquote}

\begin{blockquote}
3.3. Required Notices.\end{blockquote}

\begin{blockquote}
You must include a notice in each of your Modifications that identifies you as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.\end{blockquote}

\begin{blockquote}
3.4. Application of Additional Terms.\end{blockquote}

\begin{blockquote}
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by you alone, and you hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms you offer.\end{blockquote}

\begin{blockquote}
3.5. Distribution of Executable Versions.\end{blockquote}

\begin{blockquote}
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of your choice, which may contain terms different from this License, provided that you are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If you distribute the Covered Software in Executable form under a different license, you must make it absolutely clear that any terms which differ from this License are offered by you alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms you offer.\end{blockquote}

\begin{blockquote}
3.6. Larger Works.\end{blockquote}

\begin{blockquote}
You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, you must make sure the requirements of this License are fulfilled for the Covered Software.\end{blockquote}

\begin{p}
4. Versions of the License.\end{p}

\begin{blockquote}
4.1. New Versions.\end{blockquote}

\begin{blockquote}
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.\end{blockquote}

\begin{blockquote}
4.2. Effect of New Versions.\end{blockquote}

\begin{blockquote}
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which you originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, you must distribute and make the Covered Software available under the terms of the version of the License under which you originally received the Covered Software. Otherwise, you may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.\end{blockquote}
4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY
AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. § 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.
The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as
a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the distribution of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version," you have the option of following the terms and
conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms. That is the role of license notices in public domain programs; these notices tell the users that this program is free software and how to redistribute and change it.

Example notice:

Copyright (C) year name of author

This is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with Open Source Used In DNAC Platform Antelope 875
ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.</blockquote>
<p>The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.</p>
<p>You should also get your employer (if you work as a programmer) or your school, if any, to sign a `copyright disclaimer' for the program, if necessary. Here is a sample; alter the names:</p>
<quote>
Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.
</quote>
<quote>
signature of Ty Coon, 1 April 1989<br />
Ty Coon, President of Vice</quote>
<p>This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.</p>
<li style="background-color:yellow;">
<p><strong>`CLASSPATH' EXCEPTION TO THE GPL VERSION 2</strong></p>
<blockquote>
Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words `Oracle designates this particular file as subject to the `CLASSPATH' exception as provided by Oracle in the License file that accompanied this code.'
</blockquote>
<blockquote>
Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.
</blockquote>
<blockquote>
As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.</blockquote>
</li>
</html>

This copy of Jackson JSON processor is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:
A copy is also included with both the downloadable source code package and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution, the location should be "release-notes/asl"; and in jar "META-INF/"
The project is licensed under the Confluent Community License, except for client libs, which is under the Apache 2.0 license.

See LICENSE file in each subfolder for detailed license agreement.
Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License

Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
</head>
<body>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Home page of The Apache Software Foundation">
    <meta name="msapplication-TileColor" content="#603cba">
    <meta name="msapplication-TileImage" content="/favicons/mstile-144x144.png">
    <meta name="msapplication-config" content="/favicons/browserconfig.xml">
    <meta name="theme-color" content="#303284">
    <title>Licenses</title>
    <link href='https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700%7cDroid+Serif:400,700' rel='stylesheet' type='text/css'>
    <link href="/css/min.bootstrap.css" rel="stylesheet">
    <link href="/css/styles.css" rel="stylesheet">
  </head>
  <body>
    <!-- Navigation -->
    <header>
      <nav class="navbar navbar-default navbar-fixed-top">
        <div class="container">
          <!-- Navigation Content -->
        </div>
      </nav>
    </header>
    <div class="container">
      <!-- Main Content -->
    </div>
  </body>
</html>
The Apache Software Foundation uses various licenses to distribute software and documentation, to accept regular contributions from individuals and corporations, and to accept larger grants of existing software products. These licenses help us achieve our goal of providing reliable and long-lived software products through collaborative open source software development. In all cases, contributors retain full rights to use their original contributions for any other purpose outside of Apache while providing the ASF and its projects the right to distribute and build upon their work within Apache.

All software produced by The Apache Software Foundation or any of its projects or subjects is licensed according to the terms of the documents listed below.

Apache License, Version 2.0 (current)

http://www.apache.org/licenses/LICENSE-2.0

The 2.0 version of the Apache License was approved by the ASF in 2004. The goals of this license revision have been to reduce the number of frequently asked questions, to allow the license to be reusable without modification by any project (including non-ASF projects), to allow the license to be included by reference instead of listed in every file, to clarify the license on submission of contributions, to require a patent license on contributions that necessarily infringe the contributor's own patents, and to move comments regarding Apache and other inherited attribution notices to a location outside the license terms (the NOTICE file).

The result is a license that is supposed to be compatible with other open source licenses, while remaining true to the original goals of the Apache Group and supportive of collaborative development across both nonprofit and commercial organizations. The Apache Software Foundation is still trying to determine if this version of the Apache License is compatible with the GPL.

All packages produced by the ASF are implicitly licensed under the Apache License, Version 2.0, unless otherwise explicitly stated. More developer documentation on how to apply the Apache License to your work can be found in Applying the Apache License, Version 2.0.

Apache License, Version 1.1 (historic)
The 1.1 version of the Apache License was approved by the ASF in 2000. The primary change from the 1.0 license is in the 'advertising clause' (section 3 of the 1.0 license); derived products are no longer required to include attribution in their advertising materials, only in their documentation.<p></p>Individual packages licensed under the 1.1 version may have used different wording due to varying requirements for attribution or mark identification, but the binding terms were all the same.<p></p>The ASF desires that all contributors of ideas, code, or documentation to any Apache projects complete, sign, and submit (via fax or email) an <a href="icla.pdf">Individual Contributor License Agreement</a> (ICLA). The purpose of this agreement is to clearly define the terms under which intellectual property has been contributed to the ASF and thereby allow us to defend the project should there be a legal dispute regarding the software at some future time. A signed ICLA is required to be on file before an individual is given commit rights to an ASF project.<p></p>For a corporation that has assigned employees to work on an Apache project, a <a href="cla-corporate.txt">Corporate CLA</a> (CCLA) is available for contributing intellectual property via the corporation, that may have been assigned as part of an employment agreement. Note that a Corporate CLA does not remove the need for every developer to sign their own ICLA as an individual, to cover any of their contributions which are not owned by the corporation signing the CCLA.<p></p>The ICLA is not tied to any employer you may have, so it is recommended to use one's personal email address in the contact details, rather than an @work address.<p></p>Your Full name will be published unless you provide an alternative Public name. For example if your full name is Andrew Bernard Charles Dickens, but you wish to be known as Andrew Dickens, please enter the latter as your Public name.<p></p>The email address and other contact details are not published.<p></p>When an individual or corporation decides to donate a body of existing software or documentation to one of the Apache projects, they need to execute a formal <a href="software-grant-template.pdf">Software Grant Agreement</a> (SGA) with the ASF. Typically, this is done after negotiating approval with the ASF <a href="http://incubator.apache.org/">Incubator</a> or one of the PMCs, since the ASF will not accept software unless there is a viable community available to support a collaborative project.<p></p>
Documents may be submitted by fax or email.

If submitting by fax, please print, sign, and fax all pages of the document to +1-919-573-9199. Please send documents right-side-up, first page first; and send only one document per fax.

If submitting by email, please fill the form with a pdf viewer, print, sign, scan all pages into a single pdf file, and email the pdf file as an attachment to secretary@apache.org. If possible, send the attachment from the email address in the document. Please send only one document per email.

If you prefer to sign electronically, please fill the form, save it locally (e.g. icla.pdf), and sign the file by preparing a detached PGP signature. For example,

```
gpg --armor --detach-sign icla.pdf
```

The above will create a file icla.pdf.asc. Send both the file and signature as attachments in the same email to secretary@apache.org. Please send only one document (file plus signature) per email. Please do not submit your public key to Apache. Instead, please upload your public key to pgpkeys.mit.edu.

The files typically are named icla.pdf and icla.pdf.asc for individual agreements; ccla.pdf and ccla.pdf.asc for corporate agreements; software-grant.pdf and software-grant.pdf.asc for grants.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

Footer -->
This product currently only contains code developed by authors of specific components, as identified by the source code files; if such notes are missing files have been created by Tatu Saloranta.

For additional credits (generally to people who reported problems) see CREDITS file.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.199 batch 0.4.0

1.199.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Shopify Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.200 buffer-equal-constant-time 1.0.1 1.0.1

1.200.1 Available under license:
Copyright (c) 2013, GoInstant Inc., a salesforce.com company
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of salesforce.com, nor GoInstant, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.201 xmlbuilder 8.2.2
1.201.1 Available under license:
   The MIT License (MIT)

Copyright (c) 2013 Ozgur Ozcitak

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.202 hive-upgrade-acid 3.1.2
1.202.1 Available under license:

   Apache License
   Version 2.0, January 2004
   http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.204 libedit 3.1-20150325-1ubuntu2

1.204.1 Available under license:

Copyright (c) 1992, 1993
The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.205 jsreport-authorization 2.4.0

1.205.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked
Open Source Used In DNAC Platform Antelope 906

Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

---

1.206 geo-ip 2.14.0

1.206.1 Available under license:

No license file was found, but licenses were detected in source scan.

public DomainResponse("
public DomainResponse()

Found in path(s):
* /opt/cola/permits/1140689171_1614870592.56/0/geoip2-2-14-0-sources-1-jar/com/maxmind/geoip2/model/DomainResponse.java
No license file was found, but licenses were detected in source scan.

=name>Apache License, Version 2.0</name>
?url>http://www.apache.org/licenses/LICENSE-2.0.html</url>

Found in path(s):
* /opt/cola/permits/1140689171_1614870592.56/0/geoip2-2-14-0-sources-1-jar/META-INF/maven/com.maxmind.geoip2/geoip2/pom.xml
No license file was found, but licenses were detected in source scan.

public DomainResponse domain(InetAddress ipAddress) throws IOException,

Found in path(s):
* /opt/cola/permits/1140689171_1614870592.56/0/geoip2-2-14-0-sources-1-jar/com/maxmind/geoip2/DatabaseReader.java

1.207 js-base64 2.6.4 2.6.4

1.207.1 Available under license:

Copyright (c) 2014, Dan Kogai
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the name of {{{project}}} nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
1.208 jersey-container-servlet-core 2.5.1

1.208.1 Available under license:

CDDL+GPL License
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation,
method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or
alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.
6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. § 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and
supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

http://glassfish.java.net/public/CDDL+GPL_1_1.html
cat jersey-container-servlet-core-2.5.1-sources/org/glassfish/jersey/servlet/internal/ThreadLocalInvoker.java
/*
 * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
 *
 * Copyright (c) 2010-2013 Oracle and/or its affiliates. All rights reserved.
 *
 * The contents of this file are subject to the terms of either the GNU
 * General Public License Version 2 only ("GPL") or the Common Development
 * and Distribution License("CDDL") (collectively, the "License"). You
 * may not use this file except in compliance with the License. You can
 * obtain a copy of the License at
 * http://glassfish.java.net/public/CDDL+GPL_1_1.html
 * or packager/legal/LICENSE.txt. See the License for the specific
The GNU General Public License (GPL) Version 2, June 1991

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the
executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent
modules to produce an executable, regardless of the license terms of these independent modules, and to copy and
distribute the resulting executable under terms of your choice, provided that you also meet, for each linked
independent module, the terms and conditions of the license of that module. An independent module is a module
which is not derived from or based on this library. If you modify this library, you may extend this exception to your
version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement
from your version.

1.209 babel-plugin-transform-es2015-
modules-umd 6.24.1

1.209.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is
being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.210 drill-format-image 1.19.0
1.210.1 Available under license:

Drill: Contrib: Format: Image
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and...
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.211 drill-storage-splunk 1.19.0

1.211.1 Available under license:

Drill : Contrib : Storage : Splunk
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS.
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "["]
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.212 unpipe 1.0.0

1.212.1 Available under license :
(The MIT License)

Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.213 http-errors 1.6.3 1.6.3

1.213.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com
Copyright (c) 2016 Douglas Christopher Wilson doug@somethingdoug.com
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.214 batik-swng 1.13

1.214.1 Available under license:
Apache Batik
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for Standardization for the definition of character entities used in the software's documentation.

This product includes images from the Tango Desktop Project (http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme (http://www.jesusda.com/projects/pasodoble).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

"Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of MERCHANTABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
1.215 arraybuffer.slice 0.0.7
1.215.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.216 wtf-8 1.0.0
1.216.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.217 jersey-server 2.32

1.217.1 Available under license:

# Notice for Jersey
This content is produced and maintained by the Eclipse Jersey project.

* Project home: https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright
All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

Angular JS, v1.6.6
* License MIT (http://www.opensource.org/licenses/mit-license.php)
* Project: http://angularjs.org
* Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1
* License: all the source code provided by AOP Alliance is Public Domain.
* Copyright: Material in the public domain is not protected by copyright
Bean Validation API 2.0.2
* License: Apache License, 2.0
* Project: http://beanvalidation.org/1.1/
* Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
  by the @authors tag.

Hibernate Validator CDI, 6.1.2.Final
* License: Apache License, 2.0
* Project: https://beanvalidation.org/
* Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

Bootstrap v3.3.7
* License: MIT license (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* Project: http://getbootstrap.com
* Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0
* License: Apache License, 2.0
* Copyright (C) 2009 The Guava Authors

javax.inject Version: 1
* License: Apache License, 2.0
* Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA
* License: Apache License, 2.0
* Project: http://www.javassist.org/
* Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.10.1
* License: Apache License, 2.0
* Project: https://github.com/FasterXML/jackson-jaxrs-providers
* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4
* License: jquery.org/license
* Project: jquery.org
* Copyright: (c) jQuery Foundation

jQuery Barcode plugin 0.3
* License: MIT & GPL (http://www.opensource.org/licenses/mit-license.php &
  http://www.gnu.org/licenses/gpl.html)
* Project: http://www.pasella.it/projects/jquery/barcode
* Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266
* License: CC0
* No copyright
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

KineticJS, v4.7.1
* License: MIT license (http://www.opensource.org/licenses/mit-license.php)
* Copyright: Eric Rowell

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0
* License: Apache License, 2.0
* Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents
* License: W3C License
* Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/
This content is produced and maintained by the Eclipse Jersey project.

* https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright
All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses
This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU
General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright: (c) 2000-2011 INRIA, France Telecom. All rights reserved.

W3.org documents
* License: W3C License
# Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.
"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients’ rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (“notices”) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively “Losses”) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice
"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These
restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis
or for a fee, you must give the recipients all the rights that you have.
You must make sure that they, too, receive or can get the source code.
And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software patents.
We wish to avoid the danger that redistributors of a free program will
individually obtain patent licenses, in effect making the program
proprietary. To prevent this, we have made it clear that any patent must
be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a
notice placed by the copyright holder saying it may be distributed under
the terms of this General Public License. The "Program", below, refers
to any such program or work, and a "work based on the Program" means
either the Program or any derivative work under copyright law: that is
to say, a work containing the Program or a portion of it, either
verbatim or with modifications and/or translated into another language.
(Hereinafter, translation is included without limitation in the term
"modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of running
the Program is not restricted, and the output from the Program is
covered only if its contents constitute a work based on the Program
(independent of having been made by running the Program). Whether that
is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source
code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will
not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM
(INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED
INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR
OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to
attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
    `show w'. This is free software, and you are welcome to redistribute
    it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

---

## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.218 libpng 1.2.54 1ubuntu1.1

1.218.1 Available under license :

This is the pre-packaged Debian Linux version of the libpng graphics library. It was packaged by Philippe Troin <phil@fifi.org> from sources originally retrieved from http://libpng.org/pub/png/libpng.html
The packaging itself is
copyright (C) 2001 Philippe Troin <phil@fifi.org>,
copyright 2002 Junichi Uekawa <dancer@debian.org>,
copyright 2003 Josselin Mouette <joss@debian.org> and
copyright 2006-2009 Anibal Monsalve Salazar <anibal@debian.org>.
It is licensed under the GNU General Public License v2 or later. On
Debian systems, the GPL is in /usr/share/common-licenses/GPL-2.

Here is the copyright and license for libpng:

libpng version 1.2.26 - April 2, 2008
Copyright (c) 1998-2008 Glenn Randers-Pehrson
(Version 0.96 Copyright (c) 1996, 1997 Andreas Dilger)
(Version 0.88 Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.)

Authors and maintainers:
libpng versions 0.71, May 1995, through 0.88, January 1996: Guy Schalnat
libpng versions 0.89c, June 1996, through 0.96, May 1997: Andreas Dilger
libpng versions 0.97, January 1998, through 1.2.26 - April 2, 2008: Glenn
See also "Contributing Authors", below.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following
this sentence.

libpng versions 1.2.6, August 15, 2004, through 1.2.26, April 2, 2008, are
Copyright (c) 2004, 2006-2008 Glenn Randers-Pehrson, and are
distributed according to the same disclaimer and license as libpng-1.2.5
with the following individual added to the list of Contributing Authors:

    Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5, October 3, 2002, are
Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are
distributed according to the same disclaimer and license as libpng-1.0.6
with the following individuals added to the list of Contributing Authors:

    Simon-Pierre Cadieux
    Eric S. Raymond
    Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the
library or against infringement. There is no warranty that our
efforts or the library will fulfill any of your particular purposes
or needs. This library is provided with all faults, and the entire
risk of satisfactory quality, performance, accuracy, and effort is with
the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are
Copyright (c) 1998, 1999, 2000 Glenn Randers-Pehrson, and are
distributed according to the same disclaimer and license as libpng-0.96,
with the following individuals added to the list of Contributing Authors:

    Tom Lane
    Glenn Randers-Pehrson
    Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are
Copyright (c) 1996, 1997 Andreas Dilger
Distributed according to the same disclaimer and license as libpng-0.88,
with the following individuals added to the list of Contributing Authors:

    John Bowler
    Kevin Bracey
    Sam Bushell
    Magnus Holmgren
    Greg Roelofs
    Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, “Contributing Authors”
is defined as the following set of individuals:

    Andreas Dilger
    Dave Martindale
    Guy Eric Schalnat
    Paul Schmidt
    Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors
and Group 42, Inc. disclaim all warranties, expressed or implied,
including, without limitation, the warranties of merchantability and of
fitness for any purpose. The Contributing Authors and Group 42, Inc.
assume no liability for direct, indirect, incidental, special, exemplary,
or consequential damages, which may result from the use of the PNG
Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this
source code, or portions hereof, for any purpose, without fee, subject
to the following restrictions:
1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

---------------------------------------------------------------------------

Copyright (c) 1998-2008 Greg Roelofs. All rights reserved.

This software is provided "as is," without warranty of any kind, express or implied. In no event shall the author or contributors be held liable for any damages arising in any way from the use of this software.

The contents of this file are DUAL-LICENSED. You may modify and/or redistribute this software according to the terms of one of the following two licenses (at your option):

LICENSE 1 ("BSD-like with advertising clause"):

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, disclaimer, and this list of conditions.
2. Redistributions in binary form must reproduce the above copyright notice, disclaimer, and this list of conditions in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

   This product includes software developed by Greg Roelofs and contributors for the book, "PNG: The Definitive Guide," published by O'Reilly and Associates.

LICENSE 2 (GNU GPL v2 or later):

---
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

---------------------------------------------------------------------------

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

This code is released under the libpng license.

libpng versions 1.0.7, July 1, 2000, through 1.2.54, November 12, 2015, are Copyright (c) 2000-2002, 2004, 2006-2015 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Cosmin Truta
Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.
libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are
Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from
libpng-0.96, and are distributed according to the same disclaimer and
license as libpng-0.96, with the following individuals added to the list
of Contributing Authors:

   Tom Lane
   Glenn Randers-Pehrson
   Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are
Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88,
and are distributed according to the same disclaimer and license as
libpng-0.88, with the following individuals added to the list of
Contributing Authors:

   John Bowler
   Kevin Bracey
   Sam Bushell
   Magnus Holmgren
   Greg Roelofs
   Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors"
is defined as the following set of individuals:

   Andreas Dilger
   Dave Martindale
   Guy Eric Schalnat
   Paul Schmidt
   Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors
and Group 42, Inc. disclaim all warranties, expressed or implied,
including, without limitation, the warranties of merchantability and of
fitness for any purpose. The Contributing Authors and Group 42, Inc.
assume no liability for direct, indirect, incidental, special, exemplary,
or consequential damages, which may result from the use of the PNG
Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this
source code, or portions hereof, for any purpose, without fee, subject
to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not
be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any
source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without
fee, and encourage the use of this source code as a component to
supporting the PNG file format in commercial products. If you use this
source code in a product, acknowledgment is not required but would be
appreciated.

END OF COPYRIGHT NOTICE, DISCLAIMER, and LICENSE.

A "png_get_copyright" function is available, for convenient use in "about"
boxes and the like:

    printf("%s", png_get_copyright(NULL));

Also, the PNG logo (in PNG format, of course) is supplied in the
files "pngbar.png" and "pngbar.jpg (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is
a certification mark of the Open Source Initiative. OSI has not addressed
the additional disclaimers inserted at version 1.0.7.

Glenn Randers-Pehrson
glennrp at users.sourceforge.net
November 12, 2015
This is the pre-packaged Debian Linux version of the libpng graphics
library. It was packaged by Philippe Troin <phil@fifi.org>
from sources originally retrieved from ftp://swrinde.nde.swri.edu/pub/png/src/

The packaging itself is
copyright (C) 2001 Philippe Troin <phil@fifi.org>,
copyright 2002 Junichi Uekawa <dancer@debian.org>,
copyright 2003 Josselin Mouette <joss@debian.org> and
copyright 2006 Anibal Monsalve Salazar <anibal@debian.org>.
It is licensed under the GNU General Public License. On
Debian systems, the GPL is in /usr/share/common-licenses/GPL.

Here is the copyright and license for libpng:

    GNU GENERAL PUBLIC LICENSE
        Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.
1.219 commander 2.20.3 2.20.3
1.219.1 Available under license :
(The MIT License)

Copyright (c) 2011 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.220 datanucleus-api-jdo 4.2.4
1.220.1 Available under license :

========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the DataNucleus distribution.     ==
========================================================================

This product includes software developed by many individuals,
including the following:

Erik Bengtson
Andy Jefferson

========================================================================
This product has included contributions from some individuals,
including the following:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2008-2008 DataNucleus

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.221 orc-tools 1.5.6
1.221.1 Available under license:

ORC Tools
Copyright 2013-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.222 parquet-format 2.3.0-incubating
1.222.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

--------------------------------------------------------------------------------

This product depends on SLF4J and includes SLF4J in this binary artifact. SLF4J is a simple logging facade for Java.

Copyright: 2004-2013 QOS.ch.
Home page: http://www.slf4j.org/
License: http://slf4j.org/license.html (MIT license)

The following is the SLF4J license (MIT):
Copyright (c) 2004-2013 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

------------------------------------------------------------------------------------------------

This product depends on Apache Thrift and includes it in this binary artifact.

Copyright: 2006-2010 The Apache Software Foundation.
Home page: https://thrift.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

Apache Parquet Format (Incubating)
Copyright 2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.223 body-parser 1.19.0
1.223.1 Available under license :
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.224 get-func-name 2.0.0

1.224.1 Available under license:
Copyright (c) 2013 Jake Luer <jake@alogicalparadox.com> (http://alogicalparadox.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.225 setprototypeof 1.1.0 1.1.0

1.225.1 Available under license:
Copyright (c) 2015, Wes Todd

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.226 emojis-list 3.0.0

1.226.1 Available under license:

The MIT License (MIT)

Copyright 2015 Kiko Beats

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.227 table 3.8.3

1.227.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.228 validatejs 0.12.0 0.12.0

1.228.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013-2016 Nicklas Ansman, 2013 Wrapp

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.229 is-ci 2.0.0

1.229.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016-2018 Thomas Watson Steen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.230 moment 2.24.0

1.230.1 Available under license:

Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.231 fast-csv 3.7.0
1.231.1 Available under license:
The MIT License

Copyright (c) 2011-2019 C2FO

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.232 wrap-ansi 6.2.0
1.232.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**1.233 @babel/plugin-syntax-object-rest-spread 7.8.3**

**1.233.1 Available under license :**

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**1.234 jackson-dataformat-xml 2.9.10**

**1.234.1 Available under license :**

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

## 1.235 oas-linter 3.2.1 3.2.1

### 1.235.1 Available under license:

BSD 3-Clause License

Copyright (c) 2016, Mermade Software
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.236 is-primitive 2.0.0

1.236.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.237 jul-to-slf4j 1.7.5

1.238 batik-ext 1.13

1.238.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2006 World Wide Web Consortium,
 * (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved. This
* work is distributed under the W3C(r) Software License [1] in the hope that
* it will be useful, but WITHOUT ANY WARRANTY; without even the implied
* warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

*/

Found in path(s):
* /opt/cola/permits/1146690638_1616599313.36/0/batik-ext-1-13-sources-1-
  jar/org/apache/batik/w3c/dom/events/CustomEvent.java
* /opt/cola/permits/1146690638_1616599313.36/0/batik-ext-1-13-sources-1-
  jar/org/apache/batik/w3c/dom/events/KeyboardEvent.java
* /opt/cola/permits/1146690638_1616599313.36/0/batik-ext-1-13-sources-1-
  jar/org/apache/batik/w3c/dom/events/MutationNameEvent.java
* /opt/cola/permits/1146690638_1616599313.36/0/batik-ext-1-13-sources-1-
  jar/org/apache/batik/w3c/dom/events/TextEvent.java

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE
2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1146690638_1616599313.36/0/batik-ext-1-13-sources-1-jar/META-
  INF/maven/org.apache.xmlgraphics/batik-ext/pom.xml

No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146690638_1616599313.36/0/batik-ext-1-13-sources-1-jar/org/apache/batik/w3c/dom/ElementTraversal.java
* /opt/cola/permits/1146690638_1616599313.36/0/batik-ext-1-13-sources-1-jar/org/apache/batik/w3c/dom/Location.java
* /opt/cola/permits/1146690638_1616599313.36/0/batik-ext-1-13-sources-1-jar/org/apache/batik/w3c/dom/Window.java

1.239 babel-plugin-transform-es2015-parameters 6.24.1

1.239.1 Available under license :
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

---

Open Source Used In DNAC Platform Antelope 999
Open Source Used In DNAC Platform Antelope 1000

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.240 jtoml 1.0.0

1.241 color-convert 2.0.1

1.241.1 Available under license :
Copyright (c) 2011-2016 Heather Arthur <fayearthur@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
1.242 concat-map 0.0.1

1.242.1 Available under license:

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.243 core-js 1.2.7 1.2.7

1.243.1 Available under license:

Copyright (c) 2015 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.244 caniuse-lite 1.0.30000874 1.0.30000874
1.244.1 Available under license :
Attribution 4.0 International

Creative Commons Corporation ("Creative Commons") is not a law firm and
does not provide legal services or legal advice. Distribution of
Creative Commons public licenses does not create a lawyer-client or
other relationship. Creative Commons makes its licenses and related
information available on an "as-is" basis. Creative Commons gives no
warranties regarding its licenses, any material licensed under their
terms and conditions, or any related information. Creative Commons
disclaims all liability for damages resulting from their use to the
fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and
conditions that creators and other rights holders may use to share
original works of authorship and other material subject to copyright
and certain other rights specified in the public license below. The
following considerations are for informational purposes only, are not
exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are
intended for use by those authorized to give the public
permission to use material in ways otherwise restricted by
copyright and certain other rights. Our licenses are
irrevocable. Licensors should read and understand the terms
and conditions of the license they choose before applying it.
Licensors should also secure all rights necessary before
applying our licenses so that the public can reuse the
material as expected. Licensors should clearly mark any
material not subject to the license. This includes other CC-
licensed material, or material used under an exception or
limitation to copyright. More considerations for licensors:
wiki.creativecommons.org/Considerations_for_licensors

Considerations for the public: By using one of our public
licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensor's permission is not necessary for any reason—for example, because of any applicable exception or limitation to copyright—then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public:

wiki.creativecommons.org/Considerations_for_licensees

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation,
performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

i. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.
a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

   a. reproduce and Share the Licensed Material, in whole or in part; and

   b. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

   a. Offer from the Licensor -- Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

   b. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or
may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

   a. retain the following if it is supplied by the Licensor with the Licensed Material:

      i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

      ii. a copyright notice;
iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.
For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 -- Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.
Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the Licensor. The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.

1.245 jackson-jaxrs-base 2.10.4

1.245.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.246 Iru-cache 4.1.1

1.246.1 Available under license :

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.247 unicode-property-aliases-ecmascript

1.1.0

1.247.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.248 is-property 1.0.2 1.0.2

1.248.1 Available under license :

The MIT License (MIT)

The MIT License (MIT)

Copyright (c) 2013 Mikola Lysenko

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.249 buffer-more-ints 1.0.0

1.249.1 Available under license:

Copyright (c) 2012 David Wragg. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.250 supports-color 5.5.0

1.250.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.251 which-boxed-primitive 1.0.2

1.251.1 Available under license :
MIT License

Copyright (c) 2019 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.252 commons-lang3 3.9

1.252.1 Available under license :
Apache Commons Lang
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. 
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software 
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.

1.253 core-js-compat 3.9.1
1.253.1 Available under license :
Copyright (c) 2014-2021 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy 
of this software and associated documentation files (the "Software"), to deal 
in the Software without restriction, including without limitation the rights 
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom the Software is 
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in 
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR 
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, 
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE 
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER 
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, 
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN 
THE SOFTWARE.

1.254 commons-validator 1.7
1.254.1 Available under license :
Apache Commons Validator
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at 
The Apache Software Foundation (http://www.apache.org/).

Apache License 
Version 2.0, January 2004 
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner.
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.255 curator-recipes 5.1.0
1.255.1 Available under license :

Curator Recipes
Copyright 2011-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.256 babel-types 6.26.0 6.26.0

1.256.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
1.257 hive-storage-api 2.7.0

1.257.1 Available under license:

Hive Storage API
Copyright 2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability.
incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.258 jakarta-annotation-api 1.3.5

1.258.1 Available under license :

# Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from
and are Distributed by that particular Contributor. A Contribution
"originates" from a Contributor if it was added to the Program by
such Contributor itself or anyone acting on such Contributor's behalf.
Contributions do not include changes or additions to the Program that
are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which
are necessarily infringed by the use or sale of its Contribution alone
or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this
Agreement.

"Recipient" means anyone who receives the Program under this Agreement
or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other
form, that is based on (or derived from) the Program and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that
results from an addition to, deletion from, or modification of the
contents of the Program, including, for purposes of clarity any new file
in Source Code form that contains any contents of the Program. Modified
Works shall not include works that contain only declarations,
interfaces, types, classes, structures, or files of the Program solely
in each case in order to link to, bind by name, or subclass the Program
or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available
in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making
modifications, including but not limited to software source code,
documentation source, and configuration files.

"Secondary License" means either the GNU General Public License,
Version 2.0, or any later versions of that license, including any
exceptions or additional permissions as identified by the initial
Contributor.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in
accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

   iii) does not attempt to limit or alter the recipients’ rights in the Source Code under section 3.2; and

   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

   a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

   b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (“notices”) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program,
the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS
7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient
receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this
License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source
code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH
YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

    Yoyodyne, Inc., hereby disclaims all copyright interest in the
    program `Gnomovision' (which makes passes at compilers) written by
    James Hacker.

    signature of Ty Coon, 1 April 1989
    Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

---

## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is
making a combined work based on this library. Thus, the terms and
conditions of the GNU General Public License version 2 cover the whole
combination.

As a special exception, the copyright holders of this library give you
permission to link this library with independent modules to produce an
executable, regardless of the license terms of these independent
modules, and to copy and distribute the resulting executable under
terms of your choice, provided that you also meet, for each linked
independent module, the terms and conditions of the license of that
module. An independent module is a module which is not derived from or
based on this library. If you modify this library, you may extend this
exception to your version of the library, but you are not obligated to
do so. If you do not wish to do so, delete this exception statement
from your version.
# Notices for Jakarta Annotations

This content is produced and maintained by the Jakarta Annotations project.

* Project home: https://projects.eclipse.org/projects/ee4j.ca

## Trademarks

Jakarta Annotations is a trademark of the Eclipse Foundation.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/common-annotations-api

## Third-party Content

## Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

1.259 jackson 2.10.5

1.259.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.salaranta@iki.fi), and has been in development since 2007.
It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses. This copy of Jackson JSON processor annotations is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

This copy of Jackson JSON processor streaming parser/generator is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

### 1.260 protobuf-java 3.11.0

#### 1.260.1 Available under license :

```
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
```
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.261 utils-merge 1.0.0 1.0.0

1.261.1 Available under license:
(The MIT License)

Copyright (c) 2013 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.262 jdo-api 3.0.1

1.263 jython 2.5.2
1.263.1 Available under license:

# Copyright (c) 1998-2000 John Aycock
#
# Permission is hereby granted, free of charge, to any person obtaining
# a copy of this software and associated documentation files (the
# "Software"), to deal in the Software without restriction, including
# without limitation the rights to use, copy, modify, merge, publish,
# distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so, subject to
# the following conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
# IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
# CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
# TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
/*
 * Copyright 2000 Finn Bock
 * *
 * This program contains material copyrighted by:
 * Copyright (c) 1997-2000 by Secret Labs AB. All rights reserved.
 * *
 * This version of the SRE library can be redistributed under CNRI's
 * Python 1.6 license. For any other use, please contact Secret Labs
 * AB (info@pythonware.com).
 *
 * Portions of this engine have been developed in cooperation with
 * CNRI. Hewlett-Packard provided funding for 1.6 integration and
 * other compatibility work.
 */
/*
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2002 The Apache Software Foundation. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation

* and/or other materials provided with the distribution.
* 
* 3. The end-user documentation included with the redistribution, if any, must
* include the following acknowledgement: "This product includes software
* developed by the Apache Software Foundation (http://www.apache.org/)."
* Alternately, this acknowledgement may appear in the software itself, if and
* wherever such third-party acknowledgements normally appear.
* 
* 4. The names "The Jakarta Project", "Ant", and "Apache Software Foundation"
* must not be used to endorse or promote products derived from this software
* without prior written permission. For written permission, please contact
* apache@apache.org.
* 
* 5. Products derived from this software may not be called "Apache" nor may
* "Apache" appear in their names without prior written permission of the Apache
* Group.
* 
* THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
* INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
* FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE
* SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
* OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* 
* This software consists of voluntary contributions made by many individuals on
* behalf of the Apache Software Foundation. For more information on the Apache
* Software Foundation, please see <http://www.apache.org/>.
*/
/*
[The "BSD licence"]
Copyright (c) 2005-2008 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*
===============================================================================

The Jython License
===============================================================================

A. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING JYTHON
===============================================================================

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2
===============================================================================

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Jython") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Jython alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2007 Python Software Foundation; All Rights Reserved" are retained in Jython alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Jython or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Jython.

4. PSF is making Jython available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS.
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF JYTHON WILL NOT
INFRINGEMENT ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF JYTHON
FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING JYTHON,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any
relationship of agency, partnership, or joint venture between PSF and
Licensee. This License Agreement does not grant permission to use PSF
trademarks or trade name in a trademark sense to endorse or promote
products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Jython, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.

Jython 2.0, 2.1 License
--------------------------------------------
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in
  the documentation and/or other materials provided with the distribution.

- Neither the name of the Jython Developers nor the names of
  its contributors may be used to endorse or promote products
  derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROBLEMS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JPython 1.1.x Software License.

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and using JPython version 1.1.x in source or binary form and its associated documentation as provided herein ("Software").

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a non-exclusive, non-transferable, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright 1996-1999 Corporation for National Research Initiatives; All Rights Reserved" are both retained in the Software, alone or in any derivative version prepared by Licensee.

Alternatively, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes), provided, however, that such text is displayed prominently in the Software alone or in any derivative version prepared by Licensee: "JPython (Version 1.1.x) is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1006. The License may also be obtained from a proxy server on the Web using the following URL: http://hdl.handle.net/1895.22/1006."

3. In the event Licensee prepares a derivative work that is based on or incorporates the Software or any part thereof, and wants to make the derivative work available to the public as provided herein, then Licensee hereby agrees to indicate in any such work, in a prominently visible way, the nature of the modifications made to CNRI's Software.

4. Licensee may not use CNRI trademarks or trade name, including JPython or CNRI, in a trademark sense to endorse or promote products or services of Licensee, or any third party. Licensee may use the mark JPython in connection with Licensee's derivative versions that are
based on or incorporate the Software, but only in the form
"JPython-based ___________________," or equivalent.

5. CNRI is making the Software available to Licensee on an "AS IS" basis.
   CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY
   OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY
   REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY
   PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE
   ANY THIRD PARTY RIGHTS.

6. CNRI SHALL NOT BE LIABLE TO LICENSEE OR OTHER USERS OF THE SOFTWARE FOR
   ANY INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF
   USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE
   THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. SOME STATES DO NOT
   ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY SO THE ABOVE DISCLAIMER
   MAY NOT APPLY TO LICENSEE.

7. This License Agreement may be terminated by CNRI (i) immediately upon
   written notice from CNRI of any material breach by the Licensee, if the
   nature of the breach is such that it cannot be promptly remedied; or
   (ii) sixty (60) days following notice from CNRI to Licensee of a
   material remediably breach, if Licensee has not remedied such breach
   within that sixty-day period.

8. This License Agreement shall be governed by and interpreted in all
   respects by the law of the State of Virginia, excluding conflict of law
   provisions. Nothing in this Agreement shall be deemed to create any
   relationship of agency, partnership, or joint venture between CNRI and
   Licensee.

9. By clicking on the "ACCEPT" button where indicated, or by installing,
   copying or otherwise using the Software, Licensee agrees to be bound by
   the terms and conditions of this License Agreement.

   [ACCEPT BUTTON]

B. HISTORY OF THE SOFTWARE

JPython was created in late 1997 by Jim Hugunin. Jim was also the
primary developer while he was at CNRI. In February 1999 Barry Warsaw
took over as primary developer and released JPython version 1.1.

In October 2000 Barry helped move the software to SourceForge
where it was renamed to Jython. Jython 2.0 and 2.1 were developed
under the Jython specific license below.

From the 2.2 release on, Jython contributors have signed
Python Software Foundation contributor agreements and releases are covered under the Python Software Foundation license version 2.

The standard library is covered by the Python Software Foundation license as well. See the Lib/LICENSE file for details.

The zxJDBC package was written by Brian Zimmer and originally licensed under the GNU Public License. The package is now covered by the Jython Software License.

The command line interpreter is covered by the Apache Software License. See the org/apache/LICENSE file for details.

1.264 mchange-commons-java 0.2.15

1.264.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.
We call this license the "Lesser" General Public License because it does less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.
The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a
table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.
This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
1.265 jetty-io 9.4.34.v20201102

1.265.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.
"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for
damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and

b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the
Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as
reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==============================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both
* The Apache 2.0 License  
http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License  

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.  
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.  
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.  
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.  
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api  
* javax.annotation:javax.annotation-api  
* javax.transaction:javax.transaction-api  
* javax.websocket:javax.websocket-api

-----

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

The following artifacts are CDDL + GPLv2 with classpath exception.
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.266 is-descriptor 0.1.6

1.266.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.267 drill-jdbc-all 1.18.0

1.267.1 Available under license:

JDBC JAR with all dependencies
Copyright 2020 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.268 esrecurse 4.3.0
1.268.1 Available under license:
No license file was found, but licenses were detected in source scan.

// Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
// * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Found in path(s):
* /opt/cola/permits/1162566273_1620862144.53/0/estools-esrecurse-v4-3-0-0-g8ceb637-1-tar-gz/estools-esrecurse-f285b57/test/traverse.js
* /opt/cola/permits/1162566273_1620862144.53/0/estools-esrecurse-v4-3-0-0-g8ceb637-1-tar-gz/estools-esrecurse-f285b57/gulpfile.babel.js
No license file was found, but licenses were detected in source scan.

/*
Copyright (C) 2014 Yusuke Suzuki <utatane.tea@gmail.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1162566273_1620862144.53/0/estools-esrecurse-v4-3-0-0-g8ceb637-1-tar-gz/estools-esrecurse-f285b57/esrecurse.js

No license file was found, but licenses were detected in source scan.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

Found in path(s):
* /opt/cola/permits/1162566273_1620862144.53/0/estools-esrecurse-v4-3-0-0-g8ceb637-1-tar-gz/estools-esrecurse-f285b57/README.md

1.269 component-emitter 1.2.1 1.2.1
1.269.1 Available under license :
(The MIT License)

Copyright (c) 2014 Component contributors <dev@component.io>

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.270 jedis 3.1.0

1.271 split 0.3.3 0.3.3

1.271.1 Available under license:

Copyright (c) 2011 Dominic Tarr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.272 jetty-webapp 9.4.41.v20210516

1.272.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and
object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.
Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the
risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformulated to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the
Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

=====================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse
The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------
Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.
1.273 @babel/plugin-transform-template-literals 7.13.0

1.273.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.274 spss-reader 1.3

1.274.1 Available under license:

No license file was found, but licenses were detected in source scan.

/ *
*Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file
*except in compliance with the License. You may obtain a copy of the License at
*
*http://www.apache.org/licenses/LICENSE-2.0
*
*Unless required by applicable law or agreed to in writing, software distributed under the
*License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
*ANY KIND, either express or implied. See the License for the specific language governing
*permissions and limitations under the License.
*/
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 BeDataDriven Groep BV
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file
 * except in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the
 * License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF
 * ANY KIND, either express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
1.275 istanbul-lib-processinfo 2.0.2
1.275.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.276 domelementtype 1.3.0 1.3.0
1.276.1 Available under license:

Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 ARISING IN ANY WAY OUT OF THE USE OF THIS,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.277 backo2 1.0.2
1.277.1 Available under license:
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.278 home-or-tmp 2.0.0 2.0.0
1.278.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.279 qs 6.7.0

1.279.1 Available under license:

Copyright (c) 2014 Nathan LaFreniere and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors

1.280 oauth-sign 0.9.0 0.9.0

1.280.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such
license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.281 color-name 1.1.1
1.281.1 Available under license:

The MIT License (MIT)
Copyright (c) 2015 Dmitry Ivanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.282 commons-lang3 2.6
1.282.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Lang
Copyright 2001-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.283 progress 2.0.3
1.283.1 Available under license :
(The MIT License)

Copyright (c) 2017 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.284 has-symbols 1.0.2
1.284.1 Available under license :
MIT License

Copyright (c) 2016 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.285 emojis-list 2.1.0 2.1.0
1.285.1 Available under license:
The MIT License (MIT)

Copyright 2015 Kiko Beats

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the Software), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.286 xmlhttprequest-ssl 1.5.5
1.286.1 Available under license:
Copyright (c) 2010 passive.ly LLC

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, 
copy, modify, merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom the 
Software is furnished to do so, subject to the following 
conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES 
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND 
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT 
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, 
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING 
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR 
OTHER DEALINGS IN THE SOFTWARE.

1.287 core-util-is 1.0.2 1.0.2

1.287.1 Available under license :
Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy 
of this software and associated documentation files (the "Software"), to 
deal in the Software without restriction, including without limitation the 
rights to use, copy, modify, publish, distribute, sublicense, and/or 
sell copies of the Software, and to permit persons to whom the Software is 
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in 
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR 
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, 
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE 
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER 
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING 
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS 
IN THE SOFTWARE.

1.288 fast-infoset 1.2.15
1.289 topo 1.1.0 1.1.0

1.289.1 Available under license:

Copyright (c) 2012-2014, Walmart and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE
FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/topo/graphs/contributors

1.290 @babel/plugin-syntax-class-properties

7.12.13

1.290.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.291 dicer 0.2.5 0.2.5

1.291.1 Available under license:
Copyright Brian White. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

1.292 calcite-avatica 1.17.0

1.292.1 Available under license:
Copyright (c) 2013 Coby Chapple
All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2004-2007 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Calcite -- Avatica
Copyright 2012-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Copyright 2008 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS.
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE CALCITE AVATICA SUBCOMPONENTS:

The Apache Calcite Avatica project contains subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

The MIT License

The Apache Calcite project bundles the following files under the MIT License:

- site
  Parts of the web site generated by Jekyll (http://jekyllrb.com/)
  Copyright (c) 2008-2015 Tom Preston-Werner
- site/_sass/_font-awesome.scss
  Font-awesome css files v4.1.0 (http://fortawesome.github.io/Font-Awesome/)
  Copyright (c) 2013 Dave Gandy
- site/_sass/_normalize.scss
  normalize.css v3.0.2 | git.io/normalize
  Copyright (c) Nicolas Gallagher and Jonathan Neal
- site/_sass/_gridism.scss
  Gridism: A simple, responsive, and handy CSS grid by @cobyism
  https://github.com/cobyism/gridism
  Copyright (c) 2013 Coby Chapple
- site/js/html5shiv.min.js
  HTML5 Shiv 3.7.2 | @afarkas @jdalton @jon_neal @rem
- site/js/respond.min.js
  Respond.js v1.4.2: min/max-width media query polyfill
  Copyright 2013 Scott Jehl
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Open Font License

The Apache Calcite project bundles the following fonts under the SIL Open Font License (OFL) - http://scripts.sil.org/OFL/

- site/fonts/fontawesome-webfont.*
  Font-awesome font files v4.0.3 (http://fortawesome.github.io/Font-Awesome/)

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-----------------------------------------------------------------------

APACHE CALCITE AVATICA SUBCOMPONENTS:

The Apache Calcite Avatica project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

3-clause BSD license

The Apache Calcite Avatica project bundles HSQLDB, which is available
under the following "3-clause BSD" license:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
/
* For work developed by the HSQL Development Group:
*
* Copyright (c) 2001-2018, The HSQL Development Group
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000, The Hypersonic SQL Group.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPersonic SQL GROUP.
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* This software consists of voluntary contributions made by many individuals
* on behalf of the Hypersonic SQL Group.
*/
#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to you under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Database of contributors to Apache Calcite.
# Pages such as developer.md use this data.
- name: Alan Gates
  apacheId: gates
  githubId: alanfgates
  org: Hortonworks
  role: PMC
- name: Aman Sinha
  apacheId: amansinha
  githubId: amansinha100
  org: MapR
  role: PMC
- name: Ashutosh Chauhan
  apacheId: hashutosh
  githubId: ashutoshc
  org: Hortonworks
  role: PMC
- name: Francis Chuang
  apacheId: francischuang
  githubId: F21
org: Boostport
role: Committer
- name: James R. Taylor
  apacheId: jamestaylor
  githubId: JamesRTaylor
  org: Salesforce
  role: PMC
- name: Jacques Nadeau
  apacheId: jacques
  githubId: jacques-n
  org: Dremio
  role: PMC
- name: Jess Camacho Rodriguez
  apacheId: jcamacho
  githubId: jcamachor
  org: Hortonworks
  role: PMC Chair
- name: Jinfeng Ni
  apacheId: jni
  githubId: jinfengni
  org: MapR
  role: PMC
- name: John Pullokkaran
  apacheId: jpullokk
  githubId: jpullokkaran
  org: Hortonworks
  role: PMC
- name: Josh Elser
  apacheId: elserj
  githubId: joshelser
  org: Hortonworks
  role: PMC
- name: Julian Hyde
  apacheId: jhyde
  githubId: julianhyde
  org: Hortonworks
  role: PMC
  homepage: http://people.apache.org/~jhyde
- name: Maryann Xue
  apacheId: maryannxue
  githubId: maryannxue
  org: Intel
  role: Committer
- name: Michael Mior
  apacheId: mmior
  githubId: michaelmior
  org: University of Waterloo
  role: Committer
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright (c) 2013 Scott Jehl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

https://fortawesome.github.io/Font-Awesome/license/

Font License

Applies to all desktop and webfont files in the following directory: font-awesome/fonts/.
License: SIL OFL 1.1
URL: http://scripts.sil.org/OFL

Code License

Applies to all CSS and LESS files in the following directories: font-awesome/css/, font-awesome/less/, and font-awesome/scss/.
License: MIT License
URL: http://opensource.org/licenses/mit-license.html

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE CALCITE AVATICA SUBCOMPONENTS:

The Apache Calcite Avatica project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

3-clause BSD license

The Apache Calcite Avatica project bundles Protocol Buffers, which is available under the following "3-clause BSD" license:

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

/* Copyright (c) 2001-2018, The HSQL Development Group
 * All rights reserved.
 */

* Redistributions and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 *
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 *
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 *
* Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
 *
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

The MIT License (MIT)

Copyright (c) 2014 Alexander Farkas (aFarkas)

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2008-present Tom Preston-Werner and Jekyll contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.293 aws-java-sdk-kms 1.11.375

1.293.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
 * with
 * the License. A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
 * permissions
 * and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/CreateGrantRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/InvalidAliasNameException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/DecryptResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/transform/TagResourceRequestMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/transform/ListAliasesRequestMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/GetKeyPolicyResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/transform/GetParametersForImportResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/transform/ScheduleKeyDeletionRequestProtocolMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/transform/RevokeGrantResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/ExpiredImportTokenException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
  jar/com/amazonaws/services/kms/model/OriginType.java
* /opt/ cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/ReEncryptResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/KeyListEntryMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/CancelKeyDeletionRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/AliasListEntryMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/UpdateKeyDescriptionResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/KeyManagerType.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/GenerateDataKeyResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/UpdateKeyDescriptionRequestProtocolMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/KMSInvalidStateException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/InvalidGrantTokenException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/UnsupportedOperationException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/ListGrantsRequestProtocolMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/CreateGrantResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/decryptRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/EncryptRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/ListRetirableGrantsRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/GetKeyRotationStatusResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/GetKeyRotationStatusRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/UntagResourceResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/DeleteImportedKeyMaterialResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/CreateGrantResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/ReEncryptRequestProtocolMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/ReEncryptResult.java
Open Source Used In DNAC Platform Antelope 1145
jar/com/amazonaws/services/kms/model/MalformedPolicyDocumentException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/PutKeyPolicyRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/DisableKeyRotationRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/TagJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/ImportKeyMaterialResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/CreateAliasRequestProtocolMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/CancelKeyDeletionResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/EnableKeyResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/UpdateAliasResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/TagException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/GetKeyPolicyRequestProtocolMarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/GetParametersForImportResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/UntagResourceRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/ListKeysResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/TagResourceResult.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/DataKeySpec.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/InvalidKeyUsageException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/AlreadyExistsException.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/RetireGrantRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/KeyListEntry.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/GenerateDataKeyWithoutPlaintextRequest.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/KeyMetadata.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/GenerateDataKeyWithoutPlaintextResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
jar/com/amazonaws/services/kms/model/transform/GenerateDataKeyWithoutPlaintextResultJsonUnmarshaller.java
* /opt/cola/permits/1135892260_1613626809.29/0/aws-java-sdk-kms-1-11-375-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
 * with
 * the License. A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
 * permissions
 * and limitations under the License.
 */

Found in path(s):
*/

1.294 jetty-jmx 9.4.34.v20201102

1.294.1 Available under license:
This program and the accompanying materials are made available under the
terms of the Eclipse Public License 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0, or the Apache Software License
2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC

LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution.
No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses
to its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other
intellectual property rights of any other entity. Each Contributor
disclaims any liability to Recipient for claims brought by any other
entity based on infringement of intellectual property rights or
otherwise. As a condition to exercising the rights and licenses granted
hereunder, each Recipient hereby assumes sole responsibility to secure
any other intellectual property rights needed, if any. For example, if a
third party patent license is required to allow Recipient to distribute
the Program, it is Recipient's responsibility to acquire that license
before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient
copyright rights in its Contribution, if any, to grant the copyright
license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under
its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties
      and conditions, express and implied, including warranties or
      conditions of title and non-infringement, and implied warranties or
      conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for
       damages, including direct, indirect, special, incidental and
       consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are
       offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such
       Contributor, and informs licensees how to obtain it in a reasonable
       manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
   Contributors may not remove or alter any copyright notices contained
   within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.
4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement , including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

===================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
===================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
  * org.eclipse.jetty.orbit:org.eclipse:jdt.core

The following artifacts are EPL and ASL2.
  * org.eclipse.jetty.orbit:javax.security.auth.message
The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.
1.295.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*global unescape, module, define, window, global*/

/*
UriTemplate Copyright (c) 2012-2013 Franz Antesberger. All Rights Reserved.
Available via the MIT license.
*/

(function (exportCallback) {
    "use strict";

    Found in path(s):
    * /opt/ws_local/PERMITS_SQL/1063105573_1593668271.03/0/uritemplate-js-0-3-4-zip/uritemplate-js-0.3.4/src/pre.txt
    No license file was found, but licenses were detected in source scan.

    /*
    UriTemplate Copyright (c) 2012-2013 Franz Antesberger. All Rights Reserved.
    Available via the MIT license.
    */

    Found in path(s):
    * /opt/ws_local/PERMITS_SQL/1063105573_1593668271.03/0/uritemplate-js-0-3-4-zip/uritemplate-js-0.3.4/bin/uritemplate.js
    No license file was found, but licenses were detected in source scan.

    URI Template JS
    ===============

    This is a javascript implementation of [RFC6570](http://tools.ietf.org/html/rfc6570) - URI Template,
    and can expand templates up to and including Level 4 in that specification.

    It exposes a constructor function UriTemplate with the two methods:

    * (static) parse(uriTemplateText) returning an instance of UriTemplate
    * expand(variables) returning an string

    Be aware, that a parsed UriTemplate is frozen, so it is stateless. You can reuse instances of UriTemplates.

    Requirements
    ============

    You can use uritemplate.js in any even not so modern browser (Tested even with IE8 in IE7-Mode), see file demo.html.
    But you can also use it with node:
**npm install uritemplate**

and then in a node application:

```javascript
var
    UriTemplate = require('uritemplate'),
    template;

template = UriTemplate.parse('{?query*}');

template.expand({query: {first: 1, second: 2}});

--> "?first=1&second=2"
```

or within a html document (see also demo.html):

```html
<script type="text/javascript" src="bin/uritemplate.js"></script>

<script type="text/javascript">
    var template = UriTemplate.parse('{?query*}');
    alert(template.expand({query: {first: 1, second: 2}}));
</script>
```

If you want to clone the git project, be aware of the submodule uritemplate-test.
So you have to to:

```
git clone --recursive https://github.com/fxa/uritemplate-js.git
```

Build
-----

```
  npm install
  npm test
```

Tests
-----

The integration tests are taken from https://github.com/uri-templates/uritemplate-test as a submodule. The tests are integrated in the Jakefile.

License
-------

Copyright 2013 Franz Antesberger

MIT License, see http://mit-license.org/

Release Notes
--------------

* 0.3.3 fixed https://github.com/fxa/uritemplate-js/issues/12 Pct-Encoding-Error with chars < 0x100 (missing 0-padding). Thanks to https://github.com/pfraze!
* 0.3.2 fixed https://github.com/fxa/uritemplate-js/issues/11 Problems with older IE versions. Thanks to anozaki!
* 0.3.1 fixed https://github.com/fxa/uritemplate-js/issues/10 thank you, Paul-Martin!
* 0.3.0 introduced UriTemplateError as exception, so the interface changed from string to UriTemplateError (as the rfc suggested)
* 0.2.4 fixed double encoding according [RubenVerborgh] and some Prefix modifiers bugs
* 0.2.3 fixed bug with empty objects ('?empty=') with '{empty:{}}' shall expand to '?empty=')
* 0.2.2 fixed pct encoding bug with multibyte utf8 chars
* 0.2.1 fixed a bug in package.json
* 0.2.0 heavy project refactoring, splitting source files, introducing jshint (preparation of next steps)

Next Steps
----------
* Implementing more unit tests (now only a view tests are implemented)
* A new method extract(uri), which tries to extract the variables from a given uri.
  This is harder, than you might think

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1063105573_1593668271.03/0/uritemplate-js-0-3-4-zip/uritemplate-js-0.3.4/README.md
No license file was found, but licenses were detected in source scan.

/*
RequireJS 0.27.1 Copyright (c) 2010-2011, The Dojo Foundation All Rights Reserved.
Available via the MIT or new BSD license.
see: http://github.com/jrburke/requirejs for details
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1063105573_1593668271.03/0/uritemplate-js-0-3-4-zip/uritemplate-js-0.3.4/require.js

1.296 nan 1.3.0

1.296.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.297 @babel/types 7.13.14
1.297.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.298 hbase-protocol 2.4.2
1.298.1 Available under license:
Apache HBase - Protocol
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.299 streamsearch 0.1.2

1.299.1 Available under license :
Copyright Brian White. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

1.300 jsreport-pptx 0.7.0

1.300.1 Available under license :
The MIT License (MIT)

Copyright (c) 2016 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.301 javax-el 2.2.4

1.301.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
Version 1.0

* 

1. Definitions.

- 1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

- 1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

- 1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

- 1.4. "Executable" means the Covered Software in any form other than Source Code.

- 1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

- 1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

- 1.7. "License" means this document.

- 1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the
initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

Open Source Used In DNAC Platform Antelope 1174
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have 
made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first 
distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from 
the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the 
combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property 
claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, 
reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or 
portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a 
Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that 
Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to 
make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor 
(or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor 
Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes 
or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor 
have deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of 
Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except 
as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the 
absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.
6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire
Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

1.302 debug 4.1.1

1.302.1 Available under license :
(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT
LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.303 ratelimitj-inmemory 0.4.0
1.303.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.304 pkg-dir 4.2.0

1.304.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0. To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.306 ansi-styles 3.2.1

1.306.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.307 array-includes 3.1.1

1.307.1 Available under license:
The MIT License (MIT)

Copyright (C) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.308 lodash.reduce 4.5.0 4.5.0

1.308.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.309 babel-cli 6.26.0
1.309.1 Available under license : MIT

1.310 regjsgen 0.2.0
1.310.1 Available under license:
Copyright 2014 Benjamin Tan <demoneaux@gmail.com> (https://d10.github.io/)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.311 http-errors 1.6.2 1.6.2
1.311.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com
Copyright (c) 2016 Douglas Christopher Wilson doug@somethingdoug.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.312 lodash.clonedeep 4.5.0 4.5.0

1.312.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====
Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.313 randexp 0.5.3 0.5.3
1.313.1 Available under license :
MIT License

Copyright (C) 2011 by fent

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.314 @babel/helper-simple-access 7.13.12
1.314.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.315 aws-java-sdk-s3 1.11.375
1.315.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
 * for applicable license terms and NOTICE.txt for applicable notices.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/internal/AbstractRepeatableCipherInputStream.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/DeleteVersionRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/package-info.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SetBucketLoggingConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SetBucketCrossOriginConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SetBucketVersioningConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SetBucketTaggingConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/BucketVersioningConfiguration.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/MultiFactorAuthentication.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/ObjectListing.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/GroupGrantee.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/S3VersionSummary.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SetBucketLifecycleConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/internal/MD5DigestCalculatingInputStream.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/S3ObjectSummary.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/Permission.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 */
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/
/**
* Interface for providing encryption materials.
* Implementations are free to use any strategy for providing encryption
* materials, such as simply providing static material that doesn't change,
* or more complicated implementations, such as integrating with existing
* key management systems.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/EncryptionMaterialsProvider.java
No license file was found, but licenses were detected in source scan.

/*/ 
* Copyright 2014-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* 
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at 
* 
* http://aws.amazon.com/apache2.0
* 
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/NotificationConfiguration.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/PutInstructionFileRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/SSECustomerKey.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/MaterialsDescriptionProvider.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/SSEAwsKeyManagementParamsProvider.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/auth/DecodedStreamBuffer.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/transfer/MultipleFileTransferStateChangeListener.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/KeyWrapException.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/TopicConfiguration.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/UploadObjectRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/S3Event.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/EncryptedInitiateMultipartUploadRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/SSECustomerKeyProvider.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/internal/FileDeletionEvent.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/KMSEncryptionMaterialsProvider.java

No license file was found, but licenses were detected in source scan.

/
* Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/internal/crypto/MultipartUploadContext.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/GetBucketPolicyRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/AmazonS3ClientParams.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/GetBucketWebsiteConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/internal/S3AbortableInputStream.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/internal/InitiateMultipartUploadHeaderHandler.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/RoutingRuleCondition.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/GetBucketAclRequest.java  
No license file was found, but licenses were detected in source scan.

/  
* Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
*  
* Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
* You may obtain a copy of the License at:  
*  
* http://aws.amazon.com/apache2.0  
*  
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES  
* OR CONDITIONS OF ANY KIND, either express or implied. See the  
* License for the specific language governing permissions and  
* limitations under the License.  
*/

Found in path(s):  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/inventory/InventoryOptionalField.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/DeleteBucketMetricsConfigurationRequest.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/analytics/AnalyticsFilterPredicate.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/S3CredentialsProviderChain.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/metrics/MetricsAndOperator.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/analytics/AnalyticsConfiguration.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/inventory/InventoryPredicateVisitor.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/OwnerOverride.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SetBucketAnalyticsConfigurationResult.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/metrics/MetricsPrefixPredicate.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/inventory/InventoryFrequency.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/lifecycle/LifecyclePredicateVisitor.java  
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/transfer/internal/CompleteMultipartDownload.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/SetBucketInventoryConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleNAryOperator.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/LifecycleFilterPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/analytics/AnalyticsExportDestination.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/analytics/AnalyticsPrefixPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/analytics/AnalyticsPrefixPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/analytics/AnalyticsPrefixPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/analytics/AnalyticsExportDestination.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/analytics/AnalyticsPrefixPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleFilterPredicate.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * 
 * http://aws.amazon.com/apache2.0
 * 
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/transfer/internal/CopyImpl.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/transfer/transfer/DoubleStackEndpointBuilder.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleFilterPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleFilterPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/LifecycleTagPredicate.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/lifecycle/Life...
jar/com/amazonaws/services/s3/model/MultipartUploadListing.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/OutputLocation.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/StorageClass.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/metrics/S3ServiceMetric.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/transfer/internal/CompleteMultipartCopy.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/AmazonS3.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/Encryption.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/internal/ObjectExpirationHeaderHandler.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/ExpressionType.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/Unmarshallers.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/GetObjectRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/ObjectRestoreHeaderHandler.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/ObjectTaggingProvider.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/EncryptionMaterialsAccessor.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/ObjectRestoreHeaderHandler.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/ObjectExpirationHeaderHandler.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/GetObjectRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/ObjectExpirationHeaderHandler.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/ObjectExpirationHeaderHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/KMSEncryptionMaterials.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2014-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
* for applicable license terms and NOTICE.txt for applicable notices.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/internal/crypto/SecuredCEK.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/internal/crypto/KMSSecuredCEK.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2012-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
with
* the License. A copy of the License is located at
 *
* http://aws.amazon.com/apache2.0
 *
* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR
* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
permissions
* and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/EncryptionConfiguration.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/ServerSideEncryptionByDefault.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/ServerSideEncryptionRule.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/inventory/ServerSideEncryptionKMS.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/DeleteBucketEncryptionResult.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/inventory/InventoryEncryption.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/SseKmsEncryptedObjects.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/inventory/ServerSideEncryptionS3.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/DeleteBucketEncryptionRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/SetBucketEncryptionResult.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/ServerSideEncryptionConfiguration.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/SourceSelectionCriteria.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/GetBucketEncryptionRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/GetBucketEncryptionResult.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/SetBucketEncryptionRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/SseKmsEncryptedObjectsStatus.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2011-2018 Amazon Technologies, Inc.
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:
* http://aws.amazon.com/apache2.0
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/CopyMonitor.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/GenericBucketRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/BucketCrossOriginConfiguration.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/CopyPartRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/internal/HeaderHandler.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/internal/HeaderHandler.java
/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * 
 *  http://aws.amazon.com/apache2.0
 * 
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

// The Amazon S3 DevPay token header is a special exception and can be safely moved

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * 
 *  http://aws.amazon.com/apache2.0
 * 
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

// The Amazon S3 DevPay token header is a special exception and can be safely moved

No license file was found, but licenses were detected in source scan.
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (c) 2017. Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/AmazonS3EncryptionClientBuilder.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/AmazonS3EncryptionClientParams.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/transfer/KeyFilter.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/AmazonS3EncryptionClientParamsWrapper.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (c) 2016. Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/AmazonS3EncryptionClientBuilder.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/AmazonS3EncryptionClientParams.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
  jar/com/amazonaws/services/s3/model/WebsiteConfiguration.java

---

Open Source Used In DNAC Platform Antelope 1214
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/BucketAccelerateStatus.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/internal/S3RequesterChargedHeaderHandler.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/transform/ObjectTaggingXmlFactory.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SetObjectTaggingResult.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/DeleteObjectTaggingResult.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/ObjectTagging.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/internal/DeleteObjectTaggingHeaderHandler.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/DeleteObjectTaggingRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/GetObjectAclRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/S3AccelerateUnsupported.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/S3VersionResult.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/GetObjectTaggingRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/Tag.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/GetObjectTaggingResult.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/transform/ServerSideEncryptionByDefaultStaxUnmarshaller.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/DeleteBucketReplicationConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/transform/QueueConfigurationStaxUnmarshaller.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/ReplicationDestinationConfig.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/ReplicationRule.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/SetObjectAclRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/GetBucketNotificationConfigurationRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/internal/CompleteMultipartUploadRetryCondition.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/BucketReplicationConfiguration.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/transform/NotificationConfigurationStaxUnmarshaller.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/transform/FilterRuleStaxUnmarshaller.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/transfer/TransferCompletionFilter.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/transform/GetBucketEncryptionStaxUnmarshaller.java
Licensed under the Apache License, Version 2.0 (the "License").
You may not use this file except in compliance with the License.
A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
permissions and limitations under the License.
*/

No license file was found, but licenses were detected in source scan.

Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
* for applicable license terms and NOTICE.txt for applicable notices.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/internal/eventstreaming/Header.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/internal/eventstreaming/Utils.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/internal/eventstreaming/Prelude.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/internal/eventstreaming/HeaderValue.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/internal/eventstreaming/Message.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/internal/eventstreaming/HeaderType.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/internal/eventstreaming/Checksums.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016-2018 Amazon Technologies, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* 
* http://aws.amazon.com/apache2.0
*
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/model/AbortIncompleteMultipartUpload.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1.jar/com/amazonaws/services/s3/internal/S3RequesterChargedResult.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
* for applicable license terms and NOTICE.txt for applicable notices.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* 
* http://aws.amazon.com/apache2.0
Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/StaticEncryptionMaterialsProvider.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/Download.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/DownloadImpl.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/iterable/S3Versions.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/iterable/S3Objects.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2012-2018 Amazon Technologies, Inc.
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* *
* http://aws.amazon.com/apache2.0
* *
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
* /

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/MultipleFileUpload.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/MultipleFileTransfer.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/S3ObjectInputStream.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/DownloadMonitor.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/DownloadMonitor.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/AbstractTransfer.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/TransferStateChangeListener.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/MultipleFileUploadImpl.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/MultipleFileDownloadImpl.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/MultipleFileTransferMonitor.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/MultipleFileDownload.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not
 * use this file except in compliance with the License. A copy of the License is
 * located at
 * 
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/waiters/HeadBucketFunction.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/waiters/HeadObjectFunction.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * 
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SelectObjectContentEventStream.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/Progress.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/internal/eventstreaming/MessageDecoder.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-2018 Amazon Technologies, Inc.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 * 
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/JSONOutput.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SelectRecordsInputStream.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/Stats.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/CompressionType.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SelectObjectContentEventException.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/transform/SelectObjectContentEventUnmarshaller.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SelectObjectContentEventVisitor.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SelectObjectContentEvent.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SelectObjectContentRequest.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/RequestProgress.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/JSONType.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/SelectObjectContentResult.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/model/JSONInput.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 */
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/S3ProgressPublisher.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/S3ProgressListenerChain.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/internal/S3ProgressListener.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/PauseException.java
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-jar/com/amazonaws/services/s3/transfer/PauseStatus.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * 
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
/**
* Gets the optional <code>versionIdMarker</code> parameter indicating where in the
* sorted list of all versions in the specified bucket to begin returning
* results. Results are always ordered first lexicographically (i.e.
* alphabetically) and then from most recent version to least recent
* version.
* <p>
* A key marker must be specified when specifying a <code>versionIdMarker</code>.
* Results begin immediately after the version with the specified key and
* version ID.
* </p>
*<p>
* @return The optional <code>versionIdMarker</code> parameter indicating where in the
* sorted list of all versions in the specified bucket to begin
* returning results.
* @see ListVersionsRequest#setVersionIdMarker(String)
* @see ListVersionsRequest#withVersionIdMarker(String)
*/

Found in path(s):
* /opt/cola/permits/1180370208_1626285830.02/0/aws-java-sdk-s3-1-11-375-sources-1-
jar/com/amazonaws/services/s3/model/ListVersionsRequest.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
* /
/**
* Synchronous client for accessing Amazon S3.
* <h1>Appendix: Amazon S3 client-side encryption meta information</h1>
* <h2>Introduction</h2>
* <p>
* This appendix summarizes the current crypto related meta information
* associated with an S3 object encrypted using SDK client-side encryption. In
* SDK’s terminology, these meta information are classified as "user metatdata".
* In general, the meta information is just a map of key/value pairs. Physically
* they are persisted either as metadata of the encrypted S3 object, or as a
* separate S3 object called an "instruction file" (which is just a JSON file in
* plaintext.) For all references to the Cipher Algorithm Names, Modes, and Padding,
* such as <code>"AES/CBC/PKCS5Padding"</code>, please refer to Oracle's <a
* href="http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html"
* >Java<sup>®</sup> Cryptography Architecture Standard Algorithm Name Documentation</a>
* for more details.
*
* <h2>S3 metadata vs instruction file</h2>
* <p>
* Note when a key/value pair is stored as "user metadata" as part of the S3
* metadata, the SDK always adds an additional prefix of <b>x-amz-meta-</b> to
* the key name. In contrast, the SDK doesn't add any prefix when the key names
* are stored in an instruction file. (The reason has to do with how the http
* headers are handled.)
*
* For example, <b>x-amz-key</b> would get stored as <b>x-amz-meta-x-amz-key</b>
* in S3 metadata. In contrast, <b>x-amz-key</b> would simply be stored as is
* (ie without the prefix) in an instruction file.
*
* All the keys described below are considered "user metadata" by the SDK.
*
* <h2>S3 v1 crypto meta information (supported by all SDK's)</h2>
* <p>
* For non-KMS client side encryption, this v1 crypto meta information is used
* to support the "Encryption Only" crypto mode (which involves the use of
* <code>AES/CBC/PKCS5Padding</code> for content encryption, and
* <code>AES/ECB</code> for encrypting the one-time randomly generated data
* key). For KMS client-side encryption, the v2 crypto meta information is used.
* <p>
* <table border="1">
* <tr aligned="left">
* <th>key</th>
* <th>description</th>
* </tr>
* <tr>
* <td width="200"><b>x-amz-key</b></td>
* <td>Content encrypting key (cek) in encrypted form, base64 encoded. The cek
* is randomly generated per S3 object, and is always an AES 256-bit key. The
* corresponding cipher is always "AES/CBC/PKCS5Padding".</td>
* </tr>
* <tr>
* <td><b>x-amz-iv</b></td>
* <td>Randomly generated IV (per S3 object), base64 encoded</td>
* </tr>
* <tr>
* <td><b>x-amz-matdesc </b></td>
Customer provided material description in JSON (UTF8) format. Used to identify the client-side master key (ie used to encrypt/wrap the generated content encrypting key).

Unencrypted content length (optional but should be specified whenever possible).

This v2 crypto meta information is used to support both authenticated encryption (which involves the use of `AES/GCM/NoPadding` for content encryption, and key wrapping for the one-time randomly generated data key), and KMS client-side encryption (which can either be encryption-only or authenticated encryption.)

Client provided material description in JSON format. (Same as v1). Used to identify the client-side master key.

For KMS client side encryption, the KMS Customer Master Key ID is stored as part of the material description, `<b>x-amz-matdesc</b>`, under the key-name `<code>`kms_cmk_id`</code>`.

Unencrypted content length. (optional but should be specified whenever possible).
* possible. Same as v1.)</td>
* </tr>
* <tr>
* <td><b>x-amz-wrap-alg</b></td>
* <td>Key wrapping algorithm used.
* <ul>
* <li>Supported values:
* <code>"AESWrap", "RSA/ECB/OAEPWithSHA-256AndMGF1Padding", "kms"</code></li>
* <li>No standard key wrapping is used if this meta information is absent</li>
* <li>Always set to <b>"kms"</b> if KMS is used for client-side encryption</li>
* </ul>
* </td>
* </tr>
* <tr>
* <td><b>x-amz-cek-alg</b></td>
* <td>Content encryption algorithm used.
* <ul>
* <li>Supported values:
* <code>"AES/GCM/NoPadding", "AES/CBC/PKCS5Padding"</code></li>
* <li>Default to <code>"AES/CBC/PKCS5Padding"</code> if this key is absent</li>
* </ul>
* </td>
* </tr>
* <tr>
* <td><b>x-amz-tag-len</b></td>
* <td>Tag length (in bits) when AEAD is in use.
* <ul>
* <li>Only applicable if AEAD is in use. This meta information is absent
* otherwise, or if KMS is in use</li>
* <li>Supported value: <code>"128"</code></li>
* </ul>
* </td>
* </tr>
* </table>

<p>
<h3>KMS Integration for client-side encryption</h3>
* <ol>
* <li>All client-side KMS protected S3 objects are stored in v2 crypto meta information
* format. However, the key wrapping algorithm, <b>x-amz-wrap-alg</b> is always
* set to <b>"kms"</b>.</li>
* <li>The KMS Customer Master Key ID is currently stored as part of the
* material description, <b>x-amz-matdesc</b>, under the key-name
* <b>"kms_cmk_id"</b>.</li>
* </ol>
<p>
<h4>Sample S3 crypto meta information for KMS client-side encryption</h4>
<p>
<h5>CryptoMode: Encryption Only</h5>
* <pre>
* x-amz-meta-x-amz-key-v2:
* CiAJ2GqwpJRnt4izuwoaIC2YdNRNSMElG5rw3pEI+J8hKnAQBEBAwB4DhqKSUSZ7elS7o8KGPtmtHTUTUj
BLJRua8N6RCPiAAAABB+MHwGCSqGSIb3DQEHBqBvMG0CAQAwAyJKoZhlvcNAQeBMB4GCWC-GSAFl
AwQBLjARBAyD0rzN3KHViWixDVCARCAOwhZ6xA8rob3Z0mNx3uKFicHKVvIC/PKIcALvG1oVLnf86m
cZJNPtxMkKkdVuj3Z1zGjZTuF5sOh
* x-amz-iv: xOHb+M6yOuogwG5SHJM3Q==
* x-amz-unencrypted-content-length: 32768
* x-amz-cek-alg: AES/CBC/PKCS5Padding
* x-amz-wrap-alg: kms
* x-amz-matdesc: {"kms_cmk_id":"d3fd2273-4ca0-4da5-b1c2-c89742ec6a26"}
* </pre>
* 
* <h5>CryptoMode: Authenticated Encryption</h5>

* <pre>
* x-amz-key-v2:
CiAJ2GqwpJRnt4izuwoaIC2YdNRNSMElG5rw3pEI+J8hKnAQBEBAwB4DhqKSUSZ7elS7o8KGPtmtHTUTUj
BLJRua8N6RCPiAAAABB+MHwGCSqGSIb3DQEHBqBvMG0CAQAwAyJKoZhlvcNAQeBMB4GCWC-GSAFl
AwQBLjARBAyFyW4hQmwcPK1ECARCAO1VzomKxA9DhYsNMm/BEbbjle2vLA7IY7fOkUoSqrvgXxp
NID9Qal7928m1Djsd0bPjHefjmsydC
* x-amz-iv: VZBksyij6DhrUMye
* x-amz-unencrypted-content-length: 32768
* x-amz-cek-alg: AES/GCM/NoPadding
* x-amz-wrap-alg: kms
* x-amz-matdesc: {"kms_cmk_id":"d3fd2273-4ca0-4da5-b1c2-c89742ec6a26"}
* x-amz-tag-len: 128
* </pre>
* 
* <h2>S3 metadata of an instruction file</h2>
* 
* The following key is always included as part of the S3 metadata of the
* instruction file (which is just an S3 object). Since the SDK treats this key
* as "user metadata", the key name is always prefixed with <b>x-amz-meta-</b>
* when physically stored.
* 
* <p>
* <table border=1>
* <tr aligned="left">
* <th>key</th>
* <th>description</th>
* </tr>
* <tr>
* <td width="200"><b>x-amz-crypto-instr-file</b></td>
* <td>the presence of this key is used to indicate this S3 object is an
* instruction file (of an associated client-side encrypted S3 object).</td>
* </tr>
* </table>
1.316 debug 3.2.7

1.316.1 Available under license:

(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
1.317 static-extend 0.1.2

1.317.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.318 jline-terminal-jansi 3.12.1

1.318.1 Available under license:
No license file was found, but licenses were detected in source scan.

2017, the original author or authors.

This software is distributable under the BSD license. See the terms of the BSD license in the documentation provided with this software.

https://opensource.org/licenses/BSD
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2017, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opensource.org/licenses/BSD-3-Clause
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009-2018 the original author(s).
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

No license file was found, but licenses were detected in source scan.
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2019, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
/*
 * /opt/ws_local/PERMITS_SQL/1057261525_1592255684.35/0/jline-terminal-jansi-3-12-1-sources-
 * jar/org/jline/terminal/impl/jansi/win/WindowsAnsiWriter.java

1.319 swagger-methods 1.0.8 1.0.8

1.319.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 James Messinger

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.320 invariant 2.2.4
1.320.1 Available under license:
MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.321 signal-exit 3.0.3

1.321.1 Available under license:
The ISC License

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.322 passport-http 0.3.0
1.322.1 Available under license :
(The MIT License)

Copyright (c) 2011-2013 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.323 error_prone_annotations 2.1.3
1.323.1 Available under license :
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of
fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following
You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.324 netty-codec 4.0.48.Final

1.324.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io.netty.handler.codec/MessageToMessageCodec.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io.netty.handler.codec/base64/Base64Encoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io.netty.handler.codec/string/StringDecoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io.netty.handler.codec/ByteToMessageDecoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io.netty.handler.codec/serialization/package-info.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io.netty.handler.codec/serialization/CompactObjectOutputStream.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io.netty.handler.codec/serialization/package-info.java
jar/io/netty/handler/codec/compression/JZlibEncoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec/marshalling/ThreadLocalMarshallerProvider.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_serialization/ReferenceMap.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_string/package-info.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_serialization/ObjectDecoderInputStream.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_protobuf/ProtobufVarint32LengthFieldPrepender.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_protobuf/ProtobufEncoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_DecoderException.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_ReplayingDecoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_compression/DecompressionException.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_protobuf/ProtobufVarint32FrameDecoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_bytes/ByteArrayEncoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_serialization/ClassLoaderClassResolver.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_marshalling/LimitingByteInput.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_marshalling/LengthFieldPrepender.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_Delimiters.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_LineBasedFrameDecoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_protobuf/ProtobufDecoder.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_serialization/WeakReferenceMap.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_marshalling/ChannelBufferByteOutput.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_serialization/ClassResolver.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_lengthFieldPrepender.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec_CodecException.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
* */

Found in path(s):
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io/netty/handler/codec/ProtocolDetectionResult.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io/netty/handler/codec/ProtocolDetectionState.java
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~ 
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~ 
~ http://www.apache.org/licenses/LICENSE
~ 2.0
~ 
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2012 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
Open Source Used In DNAC Platform Antelope  1246

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/
* Written by Robert Harder and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
*/
/**
* Enumeration of supported Base64 dialects.
* <p>
* The internal lookup tables in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/">Robert Harder's Public Domain
* Base64 Encoder/Decoder</a>.
*/

Found in path(s):
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
  jar/io/netty/handler/codec/base64/Base64Dialect.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* A decoder that splits the received{@link ByteBuf}s dynamically by the
* value of the length field in the message. It is particularly useful when you
* decode a binary message which has an integer header field that represents the
* length of the message body or the whole message.
* <p>
* {@link LengthFieldBasedFrameDecoder} has many configuration parameters so
that it can decode any message with a length field, which is often seen in
proprietary client-server protocols. Here are some examples that will give
you the basic idea on which option does what.

- The value of the length field in this example is 12 (0x0C) which
  represents the length of "HELLO, WORLD". By default, the decoder assumes
  that the length field represents the number of the bytes that follows the
  length field. Therefore, it can be decoded with the simplistic parameter
  combination.

```java
lengthFieldOffset   = 0
lengthFieldLength   = 2
lengthAdjustment    = 0
initialBytesToStrip = 0 (= do not strip header)
```

- Because we can get the length of the content by calling
  `ByteBuf#readableBytes()`, you might want to strip the length
  field by specifying `initialBytesToStrip`. In this example, we
  specified `2`, that is same with the length of the length field, to
  strip the first two bytes.

```java
lengthFieldOffset   = 0
lengthFieldLength   = 2
lengthAdjustment    = 0
initialBytesToStrip = 2 (= the length of the Length field)
```

- In most cases, the length field represents the length of the message body.
* only, as shown in the previous examples. However, in some protocols, the * length field represents the length of the whole message, including the * message header. In such a case, we specify a non-zero * <tt>lengthAdjustment</tt>. Because the length value in this example message * is always greater than the body length by <tt>2</tt>, we specify <tt>-2</tt> * as <tt>lengthAdjustment</tt> for compensation.

* <pre>
* lengthFieldOffset   = 0
* lengthFieldLength   = 2
* <b>lengthAdjustment</b>    = <b>-2</b> (= the length of the Length field)
* initialBytesToStrip = 0
*
* BEFORE DECODE (14 bytes) AFTER DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content |----->| Length | Actual Content |
* | 0x000E | "HELLO, WORLD" |      | 0x000E | "HELLO, WORLD" |
* +--------+----------------+      +--------+----------------+
* </pre>
*
* <h3>3 bytes length field at the end of 5 bytes header, do not strip header</h3>
*
* The following message is a simple variation of the first example. An extra * header value is prepended to the message. <tt>lengthAdjustment</tt> is zero * again because the decoder always takes the length of the prepended data into * account during frame length calculation.

* <pre>
* <b>lengthFieldOffset</b>   = <b>2</b> (= the length of Header 1)
* <b>lengthFieldLength</b>   = <b>3</b>
* lengthAdjustment    = 0
* initialBytesToStrip = 0
*
* BEFORE DECODE (17 bytes) AFTER DECODE (17 bytes)
* +----------+----------+----------------+      +----------+----------+----------------+
* | Header 1 |  Length  | Actual Content |----->| Header 1 |  Length  | Actual Content |
* | 0xCAFE  | 0x00000C | "HELLO, WORLD" |      | 0xCAFE  | 0x00000C | "HELLO, WORLD" |
* +----------+----------+----------------+      +----------+----------+----------------+
* </pre>
*
* <h3>3 bytes length field at the beginning of 5 bytes header, do not strip header</h3>
*
* This is an advanced example that shows the case where there is an extra * header between the length field and the message body. You have to specify a * positive <tt>lengthAdjustment</tt> so that the decoder counts the extra * header into the frame length calculation.

* <pre>
* lengthFieldOffset   = 0
* lengthFieldLength   = 3
* <b>lengthAdjustment</b>    = <b>2</b> (= the length of Header 1)
* initialBytesToStrip = 0
* 
* BEFORE DECODE (17 bytes)          AFTER DECODE (17 bytes)
* +-----------------+-----------------+      +-----------------+-----------------+
* | Length | Header 1 | Actual Content |----->| Length | Header 1 | Actual Content |
* | 0x00000C | 0xCAFE | "HELLO, WORLD" |      | 0x00000C | 0xCAFE | "HELLO, WORLD" |
* +-----------------+-----------------+      +-----------------+-----------------+
* </pre>
* 
* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,</h3>
* strip the first header field and the length field</h3>
* 
* This is a combination of all the examples above. There are the prepended
* header before the length field and the extra header after the length field.
* The prepended header affects the <tt>lengthFieldOffset</tt> and the extra
* header affects the <tt>lengthAdjustment</tt>. We also specified a non-zero
* <tt>initialBytesToStrip</tt> to strip the length field and the prepended
* header from the frame. If you don't want to strip the prepended header, you
* could specify <tt>0</tt> for <tt>initialBytesToSkip</tt>.
* 
* lengthFieldOffset   = 1 (= the length of HDR1)
* lengthFieldLength   = 2
* <b>lengthAdjustment</b>    = <b>1</b> (= the length of HDR2)
* <b>initialBytesToStrip</b> = <b>3</b> (= the length of HDR1 + LEN)
* 
* BEFORE DECODE (16 bytes)          AFTER DECODE (13 bytes)
* +--------+-----------------+      +-----------------+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Content |
* | 0xCA | 0x000C | 0xFE | "HELLO, WORLD" |      | 0xFE | "HELLO, WORLD" |
* +--------+-----------------+      +-----------------+
* </pre>
* 
* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,</h3>
* strip the first header field and the length field, the length field
* represents the length of the whole message</h3>
* 
* Let's give another twist to the previous example. The only difference from
* the previous example is that the length field represents the length of the
* whole message instead of the message body, just like the third example.
* We have to count the length of HDR1 and Length into <tt>lengthAdjustment</tt>. We
* Please note that we don't need to take the length of HDR2 into account
* because the length field already includes the whole header length.
* 
* lengthFieldOffset   = 1
* lengthFieldLength   = 2
* <b>lengthAdjustment</b>    = <b>-3</b> (= the length of HDR1 + LEN, negative)
* <b>initialBytesToStrip</b> = <b>3</b>
*
* BEFORE DECODE (16 bytes)  AFTER DECODE (13 bytes)
* +---------------+-----------+---------------+-----------+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Content |
* | 0xCA | 0x0010 | 0xFE | "HELLO, WORLD" |      | 0xFE | "HELLO, WORLD" |
* +---------------+-----------+---------------+-----------+
* </pre>
* @see LengthFieldPrepender
*/

Found in path(s):
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io/netty/handler/codec/LengthFieldBasedFrameDecoder.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-jar/io/netty/handler/codec/CodecOutputList.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
* under the License.
*/
/*
* Written by Robert Harder and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
*/
/**
* Utility class for {@link ByteBuf} that encodes and decodes to and from
* <a href="http://en.wikipedia.org/wiki/Base64">Base64</a> notation.
*<p>
* The encoding and decoding algorithm in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/">Robert Harder's Public Domain</a>
* Base64 Encoder/Decoder</p>.
*/

Found in path(s):
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec/base64/Base64.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec/compression/Crc32c.java
* /opt/cola/permits/1202446003_1631568114.9/0/netty-codec-4-0-48-final-sources-
jar/io/netty/handler/codec/compression/JdkZlibDecoder.java

1.325 batik-transcoder 1.13
1.325.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/svg2svg/OutputManager.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/keys/FloatKey.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/WMFHeaderProperties.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/TranscoderException.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/image/ImageTranscoder.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/keys/IntegerKey.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/AbstractTranscoder.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/svg2svg/PrettyPrinter.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/svg2svg/OutputManager.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/keys/FloatKey.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/WMFHeaderProperties.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/TranscoderException.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/print/PrintTranscoder.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/WMFConstants.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/keys/IntegerKey.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/image/ImageTranscoder.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/svg2svg/PrettyPrinter.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
jar/org/apache/batik/transcoder/AbstractTranscoder.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-jar/org/apache/batik/transcoder/wmf/tosvg/WMFTranscoder.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-jar/org/apache/batik/transcoder/wmf/tosvg/WMFPainter.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-jar/org/apache/batik/transcoder/TranscodingHints.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-jar/org/apache/batik/transcoder/keys/DOMImplementationKey.java
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-jar/org/apache/batik/transcoder/TranscoderInput.java

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-jar/org/apache/batik/transcoder/image/resources/Messages.properties

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146878592_1616810445.17/0/batik-transcoder-1-13-sources-jar/org/apache/batik/transcoder/wmf/tosvg/WMFUtilities.java

1.326 h2 1.4.197

1.327 babel-plugin-polyfill-corejs2 0.1.10

1.327.1 Available under license : MIT License

Copyright (c) 2014-present Nicol Ribaudo and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.328 to-readable-stream 1.0.0

1.328.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.329 async 2.1.2

1.329.1 Available under license:

Copyright (c) 2010-2016 Caolan McMahon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2008-2013 DataNucleus

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

========================================================================= == NOTICE file corresponding to section 4(d) of the Apache License, == == Version 2.0, in this case for the DataNucleus distribution. == ===================================================================

=================================================================== This product includes software developed by many individuals, including the following:
===================================================================
Andy Jefferson
Erik Bengtson
Joerg von Frantzius
Marco Schulze

=================================================================== This product has included contributions from some individuals, including the following:
===================================================================
Barry Haddow
Ralph Ullrich
David Ezzio
Brendan de Beer
David Eaves
Martin Taal
Tony Lai
Roland Szabo
Anton Troshin (Timesten)

=================================================================== This product also includes software developed by the TJDO project (http://tjdo.sourceforge.net/).
===================================================================

=================================================================== This product also includes software developed by the Apache Commons project (http://commons.apache.org/).

1.331 drill-metastore-api 1.19.0
1.331.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Drill : Metastore : API
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.332 commondir 1.0.1
1.332.1 Available under license :
The MIT License

Copyright (c) 2013 James Halliday (mail@substack.net)

Permission is hereby granted, free of charge,
to any person obtaining a copy of this software and
associated documentation files (the "Software"), to
deal in the Software without restriction, including
without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom
the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice
shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.333 jsreport-browser-client 2.2.2
1.333.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.334 drill-hive-exec-shaded 1.19.0
1.334.1 Available under license:

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Drill : Contrib : Storage : Hive : Exec Shaded
Copyright 2021 The Apache Software Foundation

Hive Query Language
Copyright 2019 The Apache Software Foundation

Apache Parquet Hadoop Bundle
Copyright 2018 The Apache Software Foundation

# Jackson JSON processor
Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0. To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache Commons Codec
Copyright 2002-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

1.335 ansi-styles 4.2.1

1.335.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.
1.336 raml-to-swagger 1.1.0 1.1.0
1.336.1 Available under license:
    Copyright (c) 2015,
    All rights reserved.

    Redistribution and use in source and binary forms, with or without
    modification, are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice, this
      list of conditions and the following disclaimer.

    * Redistributions in binary form must reproduce the above copyright notice,
      this list of conditions and the following disclaimer in the documentation
      and/or other materials provided with the distribution.

    * Neither the name of raml-to-swagger nor the names of its
      contributors may be used to endorse or promote products derived from
      this software without specific prior written permission.

    THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
    AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
    IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
    DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
    FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
    DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
    SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
    CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
    OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
    OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.337 pinkie-promise 2.0.1 2.0.1
1.337.1 Available under license:
    The MIT License (MIT)

    Copyright (c) Vsevolod Strukchinsky <floatdrop@gmail.com> (github.com/floatdrop)

    Permission is hereby granted, free of charge, to any person obtaining a copy
    of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.338 ecdsa-sig-formatter 1.0.10 1.0.10
1.338.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2015 D2L Corporation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.339 normalize-path 3.0.0

1.339.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software,
1.340 lucene-queries 7.4.0
1.340.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

---

Open Source Used In DNAC Platform Antelope 1277
by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

/*
 * Copyright 2001-2004 Unicode, Inc.
 *
 * Disclaimer
 *
 * This source code is provided as is by Unicode, Inc. No claims are
 * made as to fitness for any particular purpose. No warranties of any
 * kind are expressed or implied. The recipient agrees to determine
 * applicability of information provided. If this file has been
 * purchased on magnetic or optical media from Unicode, Inc., the
 * sole remedy for any claim will be exchange of defective media
 * within 90 days of receipt.
 *
 * Limitations on Rights to Redistribute This Code
 *
 * Unicode, Inc. hereby grants the right to freely use the information
 * supplied in this file in the creation of products supporting the
 * Unicode Standard, and to make copies of this file in any form

Open Source Used In DNAC Platform Antelope  1279
* for internal or external distribution as long as this notice
* remains attached.
*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/. Here is the copyright from those sources:

/*
 * Copyright (c) 2001-2009 Anders Moeller
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 *    derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton were automatically generated with the moman/finenight FSA package.

Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from ICU (http://www.icu-project.org)
The full license is available here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
 * Copyright (C) 1999-2010, International Business Machines
 * Corporation and others. All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the
 * Software, and to permit persons to whom the Software is furnished to do so,
 * provided that the above copyright notice(s) and this permission notice appear
 * in all copies of the Software and that both the above copyright notice(s) and
 * this permission notice appear in supporting documentation.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
 */
```
The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

---

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski, Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Lucene

Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such as the Unicode Character Database. See http://unicode.org/copyright.html for more details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were automatically generated with the moman/finenight FSA library, created by Jean-Philippe Barrette-LaPierre. This library is available under an MIT license, see http://sites.google.com/site/rettessite/moman and http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/ License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)
The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic, Persian, Romanian, Bulgarian, Hindi and Bengali analyzers (common) come with a default stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt

The German, Spanish, Finnish, French, Hungarian, Italian, Portuguese, Russian and Swedish light stemmers (common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil, and Edmond Nolan.

The Polish analyzer (stempel) comes with a default stopword list that is BSD-licensed created by the Carrot2 project. The file resides in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.
The SmartChineseAnalyzer source code (smartcn) was provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/) is derived from Unicode data such as the Unicode Character Database. See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary (http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP) (http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===========================================================================
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from mecab-ipadic-2.7.0-20070801

which can be obtained from http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===========================================================================
mecab-ipadic-2.7.0-20070801 Notice
===========================================================================

Nara Institute of Science and Technology (NAIST), the copyright holders, disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall NAIST be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.
In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

===========================================================================
Nori Korean Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from

mecab-ko-dic-2.0.3-20170922

which can be obtained from

https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.0.3-20170922.tar.gz

1.341 debug 2.3.3

1.341.1 Available under license:
Copyright (c) 2004-2013 Fabien Potencier

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.342.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.343 error_prone_annotations 2.3.4

1.343.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Error Prone Authors.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-
/*
 * Copyright 2014 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/concurrent/LockMethod.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/NoAllocation.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/concurrent/UnlockMethod.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Error Prone Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/concurrent/GuardedBy.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/CheckReturnValue.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/DoNotCall.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/OverridingMethodsMustInvokeSuper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 The Error Prone Authors.
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/RequiredModifiers.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/Var.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/CanIgnoreReturnValue.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/ForOverride.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/concurrent/LazyInit.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/CompileTimeConstant.java
* /opt/cola/permits/1177939649_1625570689.35/0/error-prone-annotations-2-3-4-sources-8-jar/com/google/errorprone/annotations/Immutable.java
1.344 nssocket 0.6.0

1.344.1 Available under license:

Copyright (C) 2011 Charlie Robbins, Paolo Fragomeni, & the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.345 mongodb-driver-sync 4.2.3

1.345.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/DistinctIterableImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/ChangeStreamIterableImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/KeyManagementService.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MongoCursor.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/ListIndexesIterableImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MongoClientFactory.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/package-info.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/OperationExecutor.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/gridfs/GridFSUploadStreamImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MongoDatabase.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MappingIterable.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MapReduceIterableImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MongoCollection.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/ListDatabasesIterable.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MapReduceIterable.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/KeyManagementService.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MongoClients.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MongoChangeStreamCursorImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MongoClientDelegate.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MongoBatchCursorAdapter.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MongoDatabaseImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MongoClientImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MongoIterable.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/vault/ClientEncryptions.java
* /opt/ cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/package-info.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/ListDatabasesIterableImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/KeyRetriever.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/package-info.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/ClientSession.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/package-info.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MongoClientImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MongoClient.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/ClientSession.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/package-info.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MongoDatabaseImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/client/MongoIterable.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/vault/ClientEncryptions.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/package-info.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/ListDatabasesIterableImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/KeyRetriever.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/package-info.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/internal/MongoClientImpl.java
* /opt/cola/permits/1202446882_1631568039.69/0/mongodb-driver-sync-4-2-3-sources-jar/com/mongodb/client/MongoClient.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
1.346 aggregate-error 3.1.0

1.346.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.347 asap 2.0.6

1.347.1 Available under license:

Copyright 2009-2014 Contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
1.348 stax 3.1.4

1.348.1 Available under license:

No license file was found, but licenses were detected in source scan.

/* Stax2 API extension for Streaming API for XML processing (StAX).
 *
 * Copyright (c) 2006- Tatu Saloranta, tatu.saloranta@iki.fi
 *
 * Licensed under the License specified in the file LICENSE which is
 * included with the source code.
 * You may not use this file except in compliance with the License.
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
 */opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/dom/DOMWrappingReader.java
No license file was found, but licenses were detected in source scan.

/* Stax2 API extension for Streaming Api for Xml processing (StAX).
 *
 * Copyright (c) 2006- Tatu Saloranta, tatu.saloranta@iki.fi
 *
 * Licensed under the License specified in the file LICENSE which is
 * included with the source code.
 * You may not use this file except in compliance with the License.
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
No license file was found, but licenses were detected in source scan.

/* Stax2 API extension for Streaming Api for Xml processing (StAX).
 *  
 * Copyright (c) 2006- Tatu Saloranta, tatu.saloranta@iki.fi
 *  
 * Licensed under the License specified in file LICENSE, included with
 * the source code.
 * You may not use this file except in compliance with the License.
 *  
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):

* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/Stax2WriterAdapter.java
* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/Stax2WriterImpl.java
* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/Stax2ReaderImpl.java

No license file was found, but licenses were detected in source scan.

/* StAX2 extension for StAX API (JSR-173).
 *  
 * Copyright (c) 2005- Tatu Saloranta, tatu.salaranta@iki.fi
 *  
 * Licensed under the License specified in file LICENSE, included with
 * the source code.
 * You may not use this file except in compliance with the License.
 *  
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):

* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/Base64DecoderBase.java
* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/Base64DecoderBase.java
jar/org/codehaus/stax2/typed/CharArrayBase64Decoder.java
* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-
jar/org/codehaus/stax2/typed/StringBase64Decoder.java
* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-
jar/org/codehaus/stax2/util/StreamWriterDelegate.java
* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-
jar/org/codehaus/stax2/typed/NumberUtil.java
No license file was found, but licenses were detected in source scan.

/* Reference Implementation of
* Stax2 extension API (for basic Stax API, JSR-173)
* Copyright (c) 2005- Tatu Saloranta, tatu.saloranta@iki.fi
* Licensed under the License specified in file LICENSE, included with
* the source code.
* You may not use this file except in compliance with the License.
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-
jar/org/codehaus/stax2/typed/ValueEncoderFactory.java
No license file was found, but licenses were detected in source scan.

/* Stax2 extension for basic Stax API (JSR-173).
* Copyright (c) 2005- Tatu Saloranta, tatu.saloranta@iki.fi
* Licensed under the License specified in file LICENSE, included with
* the source code.
* You may not use this file except in compliance with the License.
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1129090477_1612176217.67/0/stax2-api-3-1-4-sources-
jar/org/codehaus/stax2/typed/Base64Variant.java
Reference Implementation of
Stax2 extension API (for basic Stax API, JSR-173)

Copyright (c) 2008- Tatu Saloranta, tatu.salaranta@iki.fi

Licensed under the License specified in file LICENSE, included with
the source code.
You may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Woodstox XML processor

Copyright (c) 2004- Tatu Saloranta, tatu.salaranta@iki.fi

Licensed under the License specified in the file LICENSE which is
included with the source code.
You may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

No license file was found, but licenses were detected in source scan.
1.349 media-typer 0.3.0

1.349.1 Available under license:
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.350 range-parser 1.2.1

1.350.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015-2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.351 hadoop-yarn-api 3.2.2

1.351.1 Available under license:

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of org.iq80.leveledb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of Gson 2.2.4,

which has the following notices:

The Netty Project

Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:
This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn’s ASL2 AbstractNodeQueue, which is based on Dmitriy Vykov’s non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart’s 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in
This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate
a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/
This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===============================================================================

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.
The binary distribution of this product bundles binaries of
Commons Lang 2.6,
which has the following notices:
* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of
Apache Log4j 1.2.17,
which has the following notices:
* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of
"Java Concurrency in Practice" book annotations 1.0,
which has the following notices:
* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,
which has the following notices:
* ==============================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
 ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both
* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html
OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common http://hadoop.apache.org/ (Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by "GCC Runtime Library Exception"

== Contributors ==
* Tatu Saloranta
  * Providing benchmark suite
* Alec Wysoker
  * Performance and memory usage improvement

The binary distribution of this product bundles binaries of Xerces2 Java Parser 2.9.1, which has the following notices:
* =================================================================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xerces Java distribution. ==
================================================================================================================

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

The binary distribution of this product bundles binaries of
Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0

--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting
it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was
relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

This product includes portions of the Guava project v14, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of
Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library
distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5
(https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann
The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
  which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1,
which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9
which has the following notices:
* =---------------------------------------------------------------------------------
  = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
  =---------------------------------------------------------------------------------
  This product includes software developed by
  Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64,
LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 * 
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL
 * nor the names of its contributors may be used to endorse or
 * promote products derived from this software without specific prior
 * written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, *
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, *
* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; *
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER *
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT *
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN *
* ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE *
* POSSIBILITY OF SUCH DAMAGE.
*/

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/
/*++
 *
* Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
* This software program is licensed subject to the BSD License,
* available at http://www.opensource.org/licenses/bsd-license.html
*
* Abstract: The main routine
*
--*/

For src/main/native/src/org/apache/hadoop/io/compress/lz4/{lz4.h,lz4.c,lz4hc.h,lz4hc.c},

/*
LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at:
- LZ4 source repository: http://code.google.com/p/lz4/
- LZ4 public forum: https://groups.google.com/forum/#!forum/lz4c
*/

For hadoop-common-project/hadoop-common/src/main/native/gtest
---------------------------------------------------------------------
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1
---------------------------------------------------------------------
(GO license)
This is a work derived from Russ Cox's RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni (https://github.com/fusesource/leveldbjni), which is available under the following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of FuseSource Corp. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

============

The binary distribution of this product bundles binaries of leveldb
(http://code.google.com/p/leveldb/), which is available under the following
license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy
(http://code.google.com/p/snappy/), which is available under the following
license:
Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
- hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
- hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
- bootstrap v3.3.6
- broccoli-asset-rev v2.4.2
- broccoli-funnel v1.0.1
- datatables v1.10.8
- em-helper v0.5.13
- em-table v0.1.6
- ember v2.2.0
- ember-array-contains-helper v1.0.2
- ember-bootstrap v0.5.1
- ember-cli v1.13.13
- ember-cli-app-version v1.0.0
- ember-cli-babel v5.1.6
- ember-cli-content-security-policy v0.4.0
- ember-cli-dependency-checker v1.2.0
- ember-cli-htmlbars v1.0.2
- ember-cli-htmlbars-inline-precompile v0.3.1
- ember-cli-ic-ajax v0.2.1
- ember-cli-inject-live-reload v1.4.0
- ember-cli-jquery-ui v0.0.20
- ember-cli-qunit v1.2.1
- ember-cli-release v0.2.8
- ember-cli-shims v0.0.6
- ember-cli-sri v1.2.1
- ember-cli-test-loader v0.2.1
- ember-cli-uglify v1.2.0
- ember-d3 v0.1.0
- ember-data v2.1.0
- ember-disable-proxy-controllers v1.0.1
- ember-export-application-global v1.0.5
- ember-load-initializers v0.1.7
- ember-qunit v0.4.16
- ember-qunit-notifications v0.1.0
- ember-resolver v2.0.3
- ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this
file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the
License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
express or implied. See the License for the specific language governing permissions and
limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js
--------------------------------------------------------------------------------
The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software
and associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT
LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE
OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js
--------------------------------------------------------------------------------
The MIT License
Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery
Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
oj! Algorithms - version 43.0

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

For:

hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js

-----------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

D3 is available under a 3-clause BSD license. For details, see:
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
- HSQLDB Database 2.3.4

(HSQL License)

"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its
contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPersonic SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.”
The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

--------------------------------------------------------------------------------
(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licenseable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work;

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this
License;

(d) Notwithstanding Section 2.1(b) above, no patent license is
granted: (1) for code that You delete from the Original Software, or (2) for
infringements caused by: (i) the modification of the Original Software, or
(ii) the combination of the Original Software with other software or
devices.

2.2. Contributor Grant. Conditioned upon Your compliance with
Section 3.1 below and subject to third party intellectual property claims, each
Contributor hereby grants You a world-wide, royalty-free, non-exclusive
license:

(a) under intellectual property rights (other than patent or
trademark) Licensable by Contributor to use, reproduce, modify, display,
perform, sublicense and distribute the Modifications created by such Contributor
(or portions thereof), either on an unmodified basis, with other Modifications,
as Covered Software and/or as part of a Larger Work; and

(b) under Patent
Claims infringed by the making, using, or selling of Modifications made by that
Contributor either alone and/or in combination with its Contributor Version (or
portions of such combination), to make, use, sell, offer for sale, have made,
and/or otherwise dispose of: (1) Modifications made by that Contributor (or
portions thereof); and (2) the combination of Modifications made by that
Contributor with its Contributor Version (or portions of such
combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are
effective on the date Contributor first distributes or otherwise makes the
Modifications available to a third party.

(d) Notwithstanding Section 2.2(b)
above, no patent license is granted: (1) for any code that Contributor has
deleted from the Contributor Version; (2) for infringements caused by:
(i) third party modifications of Contributor Version, or (ii) the combination
of Modifications made by that Contributor with other software (except as part of
the Contributor Version) or other devices; or (3) under Patent Claims infringed
by Covered Software in the absence of Modifications made by that
Contributor.

3. Distribution Obligations.

3.1. Availability of Source
Code. Any Covered Software that You distribute or otherwise make available in
Executable form must also be made available in Source Code form and that Source
Code form must be distributed only under the terms of this License. You must
include a copy of this License with every copy of the Source Code form of the
Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are
fulfilled for the Covered Software.

4. Versions of the License.

4.1.
New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the
breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is
in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision
that addresses Government rights in computer software under this
License.

9. MISCELLANEOUS.
This License represents the complete agreement
concerning subject matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent necessary to
make it enforceable. This License shall be governed by the law of the
jurisdiction specified in a notice contained within the Original Software
(except to the extent applicable law, if any, provides otherwise), excluding
such jurisdictions conflict-of-law provisions. Any litigation relating to this
License shall be subject to the jurisdiction of the courts located in the
jurisdiction and venue specified in a notice contained within the Original
Software, with the losing party responsible for costs, including, without
limitation, court costs and reasonable attorneys fees and expenses. The
application of the United Nations Convention on Contracts for the International
Sale of Goods is expressly excluded. Any law or regulation which provides that
the language of a contract shall be construed against the drafter shall not
apply to this License. You agree that You alone are responsible for compliance
with the United States export administration regulations (and the export control
laws and regulation of any other countries) when You use, distribute or
otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is
responsible for claims and damages arising, directly or indirectly, out of its
utilization of rights under this License and You agree to work with Initial
Developer and Contributors to distribute such responsibility on an equitable
basis. Nothing herein is intended or shall be deemed to constitute any admission
of liability.

The binary distribution of this product bundles these dependencies under the
following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
gizzly-framework 2.2.21
gizzly-http 2.2.21
gizzly-http-server 2.2.21
1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than
fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2)
for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with
other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or
otherwise make available in Executable form must also be made available in
Source Code form and that Source Code form must be distributed only under the
terms of this License. You must include a copy of this License with every copy
of the Source Code form of the Covered Software You distribute or otherwise make
available. You must inform recipients of any such Covered Software in Executable
form as to how they can obtain such Covered Software in Source Code form in a
reasonable manner on or through a medium customarily used for software
exchange.

3.2. Modifications.
The Modifications that You create or to which
You contribute are governed by the terms of this License. You represent that You
believe Your Modifications are Your original creation(s) and/or You have
sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that
identifies You as the Contributor of the Modification. You may not remove or
alter any copyright, patent or trademark notices contained within the Covered
Software, or any notices of licensing or any descriptive text giving attribution
to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source
Code form that alters or restricts the applicable version of this License or the
recipients’ rights hereunder. You may choose to offer, and to charge a fee for,
warranty, support, indemnity or liability obligations to one or more recipients
of Covered Software. However, you may do so only on Your own behalf, and not on
behalf of the Initial Developer or any Contributor. You must make it absolutely
clear that any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial Developer
and every Contributor for any liability incurred by the Initial Developer or
such Contributor as a result of warranty, support, indemnity or liability terms
You offer.

3.5. Distribution of Executable Versions.
You may distribute the
Executable form of the Covered Software under the terms of this License or under
the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY
SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is
responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:

Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a
Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program. 
“Contributor” means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS
A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in
any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.
All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the
  names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the
following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.
--------------------------------------------------------------------------
(2-clause BSD)
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those
of the authors and should not be interpreted as representing official policies,
either expressed or implied, of the FreeBSD Project.
The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor
hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the
duration of the applicable copyright) license to exercise the rights in the Work
as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective
Works, and to reproduce the Work as incorporated in the Collective Works;
to create and reproduce Derivative Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and
perform publicly by means of a digital audio transmission the Work including as
incorporated in Collective Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and
perform publicly by means of a digital audio transmission Derivative Works.
For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive
right to collect, whether individually or via a performance rights society (e.g.
ASCAP, BMI, SESAC), royalties for the public performance or public digital
performance (e.g. webcast) of the Work.
Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right
to collect, whether individually or via a music rights agency or designated
agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the
Work ("cover version") and distribute, subject to the compulsory license created
by 17 USC Section 115 of the US Copyright Act (or the equivalent in other
jurisdictions).
Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the
Work is a sound recording, Licensor waives the exclusive right to collect,
whether individually or via a performance-rights society (e.g. SoundExchange),
royalties for the public digital performance (e.g. webcast) of the Work, subject
to the compulsory license created by 17 USC Section 114 of the US Copyright Act
(or the equivalent in other jurisdictions).
The above rights may be exercised in all media and formats whether now known or
hereafter devised. The above rights include the right to make such modifications
as are technically necessary to exercise the rights in other media and formats.
All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject
to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally
perform the Work only under the terms of this License, and You must include a
copy of, or the Uniform Resource Identifier for, this License with every copy or
phonorecord of the Work You distribute, publicly display, publicly perform, or
publicly digitally perform. You may not offer or impose any terms on the Work
that alter or restrict the terms of this License or the recipients' exercise of
the rights granted hereunder. You may not sublicense the Work. You must keep
intact all notices that refer to this License and to the disclaimer of
warranties. You may not distribute, publicly display, publicly perform, or
publicly digitally perform the Work with any technological measures that control
access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor’s copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., “French translation of the Work by Original Author,” or “Screenplay based on original Work by Original Author”). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon
any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

jamon-runtime 2.4.1

(MPL 2.0)

Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation
of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a
Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice
in Exhibit A, the Executable Form of such Source Code Form, and Modifications of
such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to
the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or
earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate
file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the
time of the initial grant or subsequently, any and all of the rights conveyed by
this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or
modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and
apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution
or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the
recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.
4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of
any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party’s ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the
10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

/*--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

/"--
Open Source Used In DNAC Platform Antelope 1366
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is
available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a
collective whole. A work that constitutes a Collection will not be
considered an Adaptation (as defined above) for the purposes of this
License.

c. "Distribute" means to make available to the public the original and
copies of the Work or Adaptation, as appropriate, through sale or
other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that
offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work,
the individual, individuals, entity or entities who created the Work
or if no individual or entity can be identified, the publisher; and in
addition (i) in the case of a performance the actors, singers,
musicians, dancers, and other persons who act, sing, deliver, declaim,
play in, interpret or otherwise perform literary or artistic works or
expressions of folklore; (ii) in the case of a phonogram the producer
being the person or legal entity who first fixes the sounds of a
performance or other sounds; and, (iii) in the case of broadcasts, the
organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms
of this License including without limitation any production in the
literary, scientific and artistic domain, whatever may be the mode or
form of its expression including digital form, such as a book,
pamphlet and other writing; a lecture, address, sermon or other work
of the same nature; a dramatic or dramatico-musical work; a
choreographic work or entertainment in dumb show; a musical
composition with or without words; a cinematographic work to which are
assimilated works expressed by a process analogous to cinematography;
a work of drawing, painting, architecture, sculpture, engraving or
lithography; a photographic work to which are assimilated works
expressed by a process analogous to photography; a work of applied
art; an illustration, map, plan, sketch or three-dimensional work
relative to geography, topography, architecture or science; a
performance; a broadcast; a phonogram; a compilation of data to the
extent it is protected as a copyrightable work; or a work performed by
a variety or circus performer to the extent it is not otherwise
considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this
License who has not previously violated the terms of this License with
respect to the Work, or who has received express permission from the
Licensor to exercise rights under this License despite a previous
violation.

h. "Publicly Perform" means to perform public recitations of the Work and
to communicate to the public those public recitations, by any means or
process, including by wire or wireless means or public digital
performances; to make available to the public Works in such a way that
members of the public may access these Works from a place and at a
place individually chosen by them; to perform the Work to the public
by any means or process and the communication to the public of the
performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.
The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URL if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay
based on original Work by Original Author”). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author’s honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author’s honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971).
These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.


Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under
the License.

1.352 parquet-format-structures 1.12.0
1.352.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/org/apache/parquet/format/event/TypedConsumer.java
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/org/apache/parquet/format/InterningProtocol.java
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/org/apache/parquet/format/event/EventBasedThriftReader.java
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/org/apache/parquet/format/BlockCipher.java
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/org/apache/parquet/format/event/FieldConsumer.java
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/org/apache/parquet/format/Util.java
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/org/apache/parquet/format/event/Consumers.java
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/org/apache/parquet/format/LogicalTypes.java
No license file was found, but licenses were detected in source scan.

/**
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446978_1631568294.17/0/parquet-format-structures-1-12-0-sources-jar/parquet.thrift

1.353 tough-cookie 2.5.0

1.353.1 Available under license:

Copyright (c) 2015, Salesforce.com, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of Salesforce.com nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
Open Source Used In DNAC Platform Antelope 1377

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.354 babel-helper-optimise-call-expression

6.24.1 6.24.1

1.354.1 Available under license :

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.355 xtend 2.1.2 2.1.2

1.355.1 Available under license :

Copyright (c) 2012 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.356 jersey-container-servlet-core 2.25.1
1.356.1 Available under license:
   CDDL+GPL License
   http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.357 oas-kit-common 1.0.8 1.0.8
1.357.1 Available under license:
   BSD 3-Clause License

   Copyright (c) 2016, Mermade Software
   All rights reserved.

   Redistribution and use in source and binary forms, with or without
   modification, are permitted provided that the following conditions are met:

   * Redistributions of source code must retain the above copyright notice, this
     list of conditions and the following disclaimer.

   * Redistributions in binary form must reproduce the above copyright notice,
     this list of conditions and the following disclaimer in the documentation
     and/or other materials provided with the distribution.

   * Neither the name of the copyright holder nor the names of its
     contributors may be used to endorse or promote products derived from
     this software without specific prior written permission.

   THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
   AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
   IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
   DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
   FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
   DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
   SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
   CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
   OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
   OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.358 is-buffer 2.0.4 2.0.4

1.358.1 Available under license :

The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.359 jersey-container-servlet-core 2.32

1.359.1 Available under license :

# Notice for Jersey
This content is produced and maintained by the Eclipse Jersey project.

* Project home: https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright
All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms
of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

Angular JS, v1.6.6
- License: MIT (http://www.opensource.org/licenses/mit-license.php)
- Project: http://angularjs.org
- Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1
- License: all the source code provided by AOP Alliance is Public Domain.
- Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2
- License: Apache License, 2.0
- Project: http://beanvalidation.org/1.1/
- Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
- by the @authors tag.

Hibernate Validator CDI, 6.1.2.Final
- License: Apache License, 2.0
- Project: https://beanvalidation.org/
- Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

Bootstrap v3.3.7
- License: MIT license (https://github.com/twbs/bootstrap/blob/master/LICENSE)
- Project: http://getbootstrap.com
- Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0
- License: Apache License, 2.0
- Copyright (C) 2009 The Guava Authors

javax.inject Version: 1
- License: Apache License, 2.0
* Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA
* License: Apache License, 2.0
* Project: http://www.javassist.org/
* Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.10.1
* License: Apache License, 2.0
* Project: https://github.com/FasterXML/jackson-jaxrs-providers
* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4
* License: jquery.org/license
* Project: jquery.org
* Copyright: (c) jQuery Foundation

jQuery Barcode plugin 0.3
* Project: http://www.pasella.it/projects/jquery/barcode
* Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266
* License: CC0
* No copyright
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

KineticJS, v4.7.1
* License: MIT license (http://www.opensource.org/licenses/mit-license.php)
* Copyright: Eric Rowell

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0
* License: Apache License, 2.0
* Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
   Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
   where such changes and/or additions to the Program originate from
   and are Distributed by that particular Contributor. A Contribution
   "originates" from a Contributor if it was added to the Program by
   such Contributor itself or anyone acting on such Contributor's behalf.
   Contributions do not include changes or additions to the Program that
   are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which
are necessarily infringed by the use or sale of its Contribution alone
or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this
Agreement.

"Recipient" means anyone who receives the Program under this Agreement
or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other
form, that is based on (or derived from) the Program and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that
results from an addition to, deletion from, or modification of the
contents of the Program, including, for purposes of clarity any new file
in Source Code form that contains any contents of the Program. Modified
Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

   iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial
Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.
5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses
granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor

---
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program
proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed
only for noncommercial distribution and only if you received the
program in object code or executable form with such an offer, in
accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source code
means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to control
compilation and installation of the executable. However, as a special
exception, the source code distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies the
executable.

If distribution of executable or object code is made by offering access
to copy from a designated place, then offering equivalent access to copy
the source code from the same place counts as distribution of the source
code, even though third parties are not compelled to copy the source
along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt otherwise
to copy, modify, sublicense or distribute the Program is void, and will
automatically terminate your rights under this License. However, parties
who have received copies, or rights, from you under this License will
not have their licenses terminated so long as such parties remain in
full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and all
its terms and conditions for copying, distributing or modifying the
Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further restrictions
on the recipients' exercise of the rights granted herein. You are not
responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
    `show w'. This is free software, and you are welcome to redistribute
    it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

    Yoyodyne, Inc., hereby disclaims all copyright interest in the
    program `Gnomovision' (which makes passes at compilers) written by
    James Hacker.

    signature of Ty Coon, 1 April 1989
    Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

---

## CLASSPATH EXCEPTION
Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.360 xtend 2.1.2
1.360.1 Available under license:
Copyright (c) 2012 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.361 type-check 0.3.2
1.361.1 Available under license:
Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.362 assign-symbols 1.0.0
1.362.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
THE SOFTWARE.

1.363 @babel/preset-modules 0.1.4
1.363.1 Available under license :
MIT License

Copyright (c) 2020 Babel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.364 component-emitter 1.1.2 1.1.2
1.364.1 Available under license :
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.365 configstore 5.0.1
1.365.1 Available under license:

BSD 2-Clause License

Copyright (c) Google
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
 CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.366 get-caller-file 2.0.5
1.366.1 Available under license:

ISC License (ISC)
Copyright 2018 Stefan Penner

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby
granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
1.367 grpc-core 1.27.1

1.367.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic
mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for
inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.368 jetty-http 9.3.27.v20190418

1.368.1 Available under license:
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are
separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and
fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS
IS’ BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient’s rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.
This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.369 append-field 1.0.0

1.369.1 Available under license:

MIT License

Copyright (c) 2019 Art Deco Code Limited

+ ES6 Code
+ Zoroaster Mask Tests
+ Depack compilation

Original Work

Copyright (c) 2015 Linus Unnebeck

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version
(or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that
identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.
4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.
6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT
LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.
10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.
1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modifications;

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.
3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

[-Sun Microsystems, Inc.-]

{+Oracle+} is the initial license steward and may publish revised
and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate
automatically if You fail to comply with terms herein and fail to
cure such breach within 30 days of becoming aware of the breach.
Provisions which, by their nature, must remain in effect beyond
the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding
declaratory judgment actions) against Initial Developer or a
Contributor (the Initial Developer or Contributor against whom You
assert such claim is referred to as "Participant") alleging that
the Participant Software (meaning the Contributor Version where
the Participant is a Contributor or the Original Software where
the Participant is the Initial Developer) directly or indirectly
infringes any patent, then any and all rights granted directly or
indirectly to You by such Participant, the Initial Developer (if
the Initial Developer is not the Participant) and all Contributors
under Sections 2.1 and/or 2.2 of this License shall, upon 60 days
notice from Participant terminate prospectively and automatically
at the expiration of such 60 day notice period, unless if within
such 60 day period You withdraw Your claim with respect to the
Participant Software against such Participant either unilaterally
or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant
alleging that the Participant Software directly or indirectly
infringes any patent where such claim is resolved (such as by
license or settlement) prior to the initiation of patent
infringement litigation, then the reasonable value of the licenses
granted by such Participant under Sections 2.1 or 2.2 shall be
taken into account in determining the amount or value of any
payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above,
all end user licenses that have been validly granted by You or any
distributor hereunder prior to termination (excluding licenses
granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT
(INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE
INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF
COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE
LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR
CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT
LIMITATION, DAMAGES FOR [-LOST PROFITS,-] LOSS OF GOODWILL, WORK STOPPAGE,
COMPUTER
FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR
LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE
POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES. SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is
responsible for claims and damages arising, directly or
indirectly, out of its utilization of rights under this License
and You agree to work with Initial Developer and Contributors to
distribute such responsibility on an equitable basis. Nothing
herein is intended or shall be deemed to constitute any admission
of liability.

cddl.1.0.txt: 2537 words  2530 100% common  2 0% deleted  5 0% changed
cddl.1.1.txt: 2607 words  2530 97% common  74 3% inserted  3 0% changed

Select All

1.371 component-emitter 1.3.0
1.371.1 Available under license :
(The MIT License)

Copyright (c) 2014 Component contributors <dev@component.io>

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

1.372 core-util-is 1.0.2
1.372.1 Available under license :
No license file was found, but licenses were detected in source scan.

    // Permission is hereby granted, free of charge, to any person obtaining a
    // copy of this software and associated documentation files (the
    // "Software"), to deal in the Software without restriction, including
// without limitation the rights to use, copy, modify, merge, publish,
// distribute, sublicense, and/or sell copies of the Software, and to permit
// The above copyright notice and this permission notice shall be included
// in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1110639221_1606842346.93/0/fis-components-core-util-is-1-0-2-0-gb94b22f-tar-gz/fis-
components-core-util-is-5917000/lib/util.js

1.373 commons-collections 4.4
1.373.1 Available under license:
Apache Commons Collections
Copyright 2001-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work.
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.374 jetty-http-spi 9.4.17.v20190418

1.374.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS
"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor
disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program. Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:
a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==============================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish
---

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

---

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following
artifacts may be included in the distribution or downloaded when ALPN
module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications
are hosted at github and both modified and original are under GPL v2 with
classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

---

OW2

The following artifacts are licensed by the OW2 Foundation according to the
terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

---

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

---

MortBay
The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.375 de.huxhorn.lilith.logback.classic 0.9.44
1.375.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:
a) under this License, provided that you make a good faith effort
to ensure that, in the event an Application does not supply the
function or data, the facility still operates, and performs
whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of
this License applicable to that copy.


The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the
Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this
license document.


You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this
license document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of
this License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

/*
 * Lilith - a log event viewer.
 * Copyright (C) 2007-2011 Joern Huxhorn
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Lesser General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 * *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Lesser General Public License for more details.
 * *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

/*
 * Copyright 2007-2011 Joern Huxhorn
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
1.376

de.huxhorn.lilith.logback.appender.multiplex-core 0.9.44

1.376.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:
a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

/*
 * Lilith - a log event viewer.
 * Copyright (C) 2007-2011 Joern Huxhorn
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Lesser General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
1.377 is-extendable 0.1.1

1.377.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.378 debug 3.2.7 3.2.7
1.378.1 Available under license:

(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.379 swagger 1.5.0

1.379.1 Available under license:

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1433634398680
Build-Jdk: 1.7.0_72
Built-By: tony
Bundle-Description: Sonatype helps open source projects to set up Maven repositories on https://oss.sonatype.org/
Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.html
Bundle-ManifestVersion: 2
Bundle-Name: swagger-models
Bundle-SymbolicName: io.swagger.models
Bundle-Version: 1.5.0
Created-By: Apache Maven Bundle Plugin
Export-Package: io.swagger.models.parameters;uses:="com.fasterxml.jackson.annotation,io.swagger.models,io.swagger.models.properties";version="1.5.0",io.swagger.models.properties;uses:="com.fasterxml.jackson.annotation,io.swagger.models.parameters,io.swagger.models.properties,javax.xml.bind.annotation";version="1.5.0"
implementation-version: 1.5.0
Import-Package: com.fasterxml.jackson.annotation;version="[2.4,3)
javax.xml.bind.annotation,org.slf4j;version="[1.6,2)
mode: development
package: io.swagger
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.6))"
Tool: Bnd-2.3.0.201405100607
url: https://github.com/swagger-api/swagger-core/modules/swagger-models

Found in path(s):
* /opt/cola/permits/1001159702_1607016290.96/0/swagger-models-1-5-0-jar/META-INF/MANIFEST.MF

1.380 xregexp 4.0.0

1.380.1 Available under license :
The MIT License

Copyright (c) 2007-2017 Steven Levithan <http://xregexp.com/>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.381 schema-utils 0.4.7 0.4.7

1.381.1 Available under license :
Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.382 is-accessor-descriptor 0.1.6

1.382.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.383 for-own 1.0.0

1.383.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2015, 2017, Jon Schlinkert
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.384 delayed-stream 1.0.0
1.384.1 Available under license:
Copyright (c) 2011 Debuggable Limited <felix@debuggable.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.385 @babel/plugin-transform-function-name 7.12.13
1.385.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.386 hawtjni-runtime 1.17

1.386.1 Available under license:
Found license 'Eclipse Public License 1.0' in 'All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0'

1.387 commons-lang3 3.11

1.387.1 Available under license:
Apache Commons Lang
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.388 event-stream 3.3.4 3.3.4

1.388.1 Available under license:
The MIT License (MIT)

Copyright (c) 2011 Dominic Tarr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.389 yargs-parser 18.1.3 18.1.3

1.389.1 Available under license:
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS.
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.390 assert-plus 1.0.0 1.0.0

1.390.1 Available under license:
The MIT License (MIT)
Copyright (c) 2012 Mark Cavage

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.391 color-convert 1.9.2

1.391.1 Available under license:
Copyright (c) 2011-2016 Heather Arthur <fayearthur@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
Merchandability, Fitness for a particular purpose and Noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

1.392 depd 1.1.1 1.1.1

1.392.1 Available under license:
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software is provided 'as is', without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

1.393 binary-extensions 1.13.1

1.393.1 Available under license:
MIT License

Copyright (c) 2019 Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com), Paul Miller (https://paulmillr.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.394 repeat-string 1.6.1

1.394.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.395 pdfbox 2.0.22

1.395.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/  

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
鑫联信用于DNAC平台的开源软件

EXTERNAL COMPONENTS

Apache PDFBox includes a number of components with separate copyright notices and license terms. Your use of these components is subject to the terms and conditions of the following licenses.

Contributions made to the original PDFBox project:

Copyright (c) 2002-2007, www.pdfbox.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of pdfbox; nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Adobe Font Metrics (AFM) for PDF Core 14 Fonts

This file and the 14 PostScript(R) AFM files it accompanies may be used, copied, and distributed for any purpose and without charge, with or without modification, provided that all copyright notices are retained; that the AFM files are not distributed without this file; that all modifications to this file or any of the AFM files are prominently noted in the modified file(s); and that this paragraph is not modified. Adobe Systems has no responsibility or obligation to support the use of the AFM files.
CMaps for PDF Fonts (http://opensource.adobe.com/wiki/display/cmap/Downloads)

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Adobe Systems Incorporated nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.


Permission is hereby granted, free of charge, to any person obtaining a copy of this documentation file to use, copy, publish, distribute, sublicense, and/or sell copies of the documentation, and to permit others to do the same, provided that:
- No modification, editing or other alteration of this document is allowed; and
- The above copyright notice and this permission notice shall be included in all copies of the documentation.

Permission is hereby granted, free of charge, to any person obtaining a copy of this documentation file, to create their own derivative works.
from the content of this document to use, copy, publish, distribute, sublicense, and/or sell the derivative works, and to permit others to do the same, provided that the derived work is not represented as being a copy or version of this document.

Adobe shall not be liable to any party for any loss of revenue or profit or for indirect, incidental, special, consequential, or other similar damages, whether based on tort (including without limitation negligence or strict liability), contract or other legal or equitable grounds even if Adobe has been advised or had reason to know of the possibility of such damages. The Adobe materials are provided on an "AS IS" basis. Adobe specifically disclaims all express, statutory, or implied warranties relating to the Adobe materials, including but not limited to those concerning merchantability or fitness for a particular purpose or non-infringement of any third party rights regarding the Adobe materials.

Liberation Fonts (https://fedorahosted.org/liberation-fonts)

Digitized data copyright (c) 2010 Google Corporation with Reserved Font Arimo, Tinos and Cousine.
Copyright (c) 2012 Red Hat, Inc. with Reserved Font Name Liberation.

This Font Software is licensed under the SIL Open Font License, Version 1.1.

This license is copied below, and is also available with a FAQ at: http://scripts.sil.org/OFL

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS
"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed
under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Twelvemonkeys (https://github.com/haraldk/TwelveMonkeys/)

Copyright (c) 2008-2016, Harald Kuhr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

○ Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

○ Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

○ Neither the name "TwelveMonkeys" nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache PDFBox
Copyright 2002-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Based on source code contributed to the original PDFBox project.
Copyright (c) 2002-2007, www.pdfbox.org

Includes the Adobe Glyph List

Includes the Zapf Dingbats Glyph List

Includes the Bidi Mirroring Glyph Property (BidiMirroring-8.0.0.txt)
Copyright 1991-2015 Unicode, Inc.

Includes parts of TwelveMonkeys ImageIO
Copyright 2008-2016 Harald Kuhr

1.396 chokidar 2.1.8
1.396.1 Available under license :
No license file was found, but licenses were detected in source scan.

The MIT License (MIT)
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1110634227_1606843214.31/0/chokidar-2-1-8-5-tgz/package/README.md

1.397 babel-code-frame 6.26.0
1.397.1 Available under license :
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The "Software identified by reference to the Ecma Standard* ("Software")" is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
**1.398 acorn 3.3.0**

**1.398.1 Available under license:**

Copyright (C) 2012-2016 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**1.399 jszip 3.5.0**

**1.399.1 Available under license:**

JSZip is dual licensed. You may use it under the MIT license *or* the GPLv3 license.

The MIT License
===============

Copyright (c) 2009-2016 Stuart Knightley, David Duponchel, Franz Buchinger, Antnio Afonso

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GPL version 3

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.
Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display
Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the
Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has
been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.
11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered
work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

1.400 log4j-slf4j-impl 2.8.2
1.400.1 Available under license :

Apache Log4j SLF4J Binding
Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.401 javax-servlet-jsp-api 2.1

1.401.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common
control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or
trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of:
(1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients’ rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,
INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses
7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and
reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

1.402 jline 2.14.6
1.402.1 Available under license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.403 annotations 4.1.1.4
1.403.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Android Open Source Project
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135864606_1613618178.74/0/annotations-4-1-1-4-sources-1-
jar/android/annotation/SuppressLint.java
* /opt/cola/permits/1135864606_1613618178.74/0/annotations-4-1-1-4-sources-1-
jar/android/annotation/TargetApi.java

1.404 map-age-cleaner 0.1.3
1.404.1 Available under license:
MIT License

Copyright (c) Sam Verschueren <sam.verschueren@gmail.com> (github.com/SamVerschueren)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.405 convert-source-map 1.7.0

1.405.1 Available under license:

Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.406 hadoop-client 3.2.2

1.406.1 Available under license:

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of Gson 2.2.4,
which has the following notices:

The Netty Project
=================
Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

-------------------------------------------------------------------------------
This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.

Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:

* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices:

* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
  Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
  TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of "Java Concurrency in Practice" book annotations 1.0, which has the following notices:

* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5) Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,
which has the following notices:

* ==============================================================

Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
  ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
Open Source Used In DNAC Platform Antelope

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies./

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common http://hadoop.apache.org/ (Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by “GCC Runtime Library Exception”

== Contributors ==
* Tatu Saloranta
* Providing benchmark suite
* Alec Wysoker
* Performance and memory usage improvement

The binary distribution of this product bundles binaries of
Xerces2 Java Parser 2.9.1,
which has the following notices:

* =========================================================================
==  NOTICE file corresponding to section 4(d) of the Apache License,  ==
==  Version 2.0, in this case for the Apache Xerces Java distribution. ==
=========================================================================

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

The binary distribution of this product bundles binaries of
Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0

--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically 'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation
This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5 (https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of Plexus Cipher: encryption/decryption Component 1.4, which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
  which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of software.amazon.ion:ion-java 1.0.1, which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of joda-time:joda-time:2.9.9, which has the following notices:
* =---------------------------------------------------------------------
  = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
  =---------------------------------------------------------------------

This product includes software developed by Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of Ehcache 3.3.1, which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64,
LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 *
* Copyright (c) 2005, European Commission project OneLab under contract 034819 (http://www.one-lab.org)
* All rights reserved.
* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the European Commission nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008, 2009, 2010 Massachusetts Institute of Technology.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*++
 *
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 *
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 *
 * Abstract: The main routine
 *
 */

For src/main/native/src/org/apache/hadoop/io/compress/lz4/\{lz4.h,lz4.c,lz4hc.h,lz4hc.c\},

/*
 LZ4 - Fast LZ compression algorithm
 Header File
 Copyright (C) 2011-2014, Yann Collet.
 BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at:
- LZ4 source repository: http://code.google.com/p/lz4/
- LZ4 public forum: https://groups.google.com/forum/#!forum/lz4c

*/

For hadoop-common-project/hadoop-common/src/main/native/gtest

----------------------------------------
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1

---------------------------------------------------------------------
(GO license)
This is a work derived from Russ Cox’s RE2 in Go, whose license
http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h
---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni (https://github.com/fusesource/leveldbjni), which is available under the following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of FuseSource Corp. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

============= 

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

--------------------------------------------------------------------------------
Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

--------------------------------------------------------------------------------
Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js
--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
em-helpers v0.5.13
em-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

--------------------------------------------------------------------------------
Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js

--------------------------------------------------------------------------------
The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js

The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery

Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
oj! Algorithms - version 43.0

--------------------------------------------------------------------------------


This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

All files located in the node_modules and external directories are
externally maintained libraries used by this software which have their
own licenses; we recommend you read them, as their terms may differ from
the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js

-----------------------------------------------------------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

-----------------------------------------------------------------------------------------------

D3 is available under a 3-clause BSD license. For details, see:
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the
following license:
HSQldb Database 2.3.4
COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation.
and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group."

The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. **Executable** means the Covered Software in any form other than Source Code.

1.5. **Initial Developer** means the individual or entity that first makes Original Software available under this License.

1.6. **Larger Work** means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. **License** means this document.

1.8. **Licensable** means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. **Modifications** means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. **Original Software** means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. **Patent Claims** means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. **Source Code** means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. **You** (or **Your**) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. **License Grants.**
2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or
such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the
5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF
COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial
The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
gizzly-http 2.2.21
gizzly-http-server 2.2.21
gizzly-http-servlet 2.2.21
gizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.
2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted:

(1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer
includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant.
Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_Generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04
--------------------------------------------------------------------------------

By obtaining, using and/or copying this work, you (the licensee) agree that you
have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation,
with or without modification, for any purpose and without fee or royalty is
hereby granted, provided that you include the following on ALL copies of the
software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the
  redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and
  conditions. If none exist, the W3C Software Short Notice should be included
  (hypertext is preferred, text is permitted) within the body of any redistributed
  or derivative code.
- Notice of any changes or modifications to the files, including the date changes
  were made. (We recommend you provide URIs to the location from which the code is
derived.)

The binary distribution of this product bundles these dependencies under the
following license:
JUnit 4.11
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the
Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.
4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement , including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in
any resulting litigation.

The binary distribution of this product bundles these dependencies under the
following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2
--------------------------------------------------------------------------------
(3-clause BSD)
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the
names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the
following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.
--------------------------------------------------------------------------------
(2-clause BSD)
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work
will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject
to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer
UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the
party to be charged with such waiver or consent. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:
jamon-runtime 2.4.1

(MPL 2.0)
Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate
file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the
time of the initial grant or subsequently, any and all of the rights conveyed by
this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or
modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and
apparatus claims, in any patent Licensable by such Contributor that would be
infringed, but for the grant of the License, by the making, using, selling,
offering for sale, having made, import, or transfer of either its Contributions
or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General
Public License, Version 2.1, the GNU Affero General Public License, Version 3.0,
or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For
legal entities, You includes any entity that controls, is controlled by,
or is under common control with You. For purposes of this definition,
control means (a) the power, direct or indirect, to cause the direction or
management of such entity, whether by contract or otherwise, or (b) ownership of
more than fifty percent (50%) of the outstanding shares or beneficial ownership
of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive
license:
under intellectual property rights (other than patent or trademark) Licensable
by such Contributor to use, reproduce, make available, modify, display, perform,
distribute, and otherwise exploit its Contributions, either on an unmodified
basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have
made, import, and otherwise transfer either its Contributions or its Contributor
Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become
effective for each Contribution on the date the Contributor first distributes
such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this
License. No additional rights or licenses will be implied from the distribution
or licensing of Covered Software under this License. Notwithstanding Section
2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third partys
modifications of Covered Software, or (ii) the combination of its Contributions
with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its
Contributions.

This License does not grant any rights in the trademarks, service marks, or
logos of any Contributor (except as may be necessary to comply with the notice
requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute
the Covered Software under a subsequent version of this License (see Section
10.2) or under the terms of a Secondary License (if permitted under the terms of
Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are
its original creation(s) or it has sufficient rights to grant the rights to its
Contributions conveyed by this License.
2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).
3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent
infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous
This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.
Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

/*--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the
following license:
Hbase Server 1.2.4
--------------------------------------------------------------------------------
This project bundles a derivative image for our Orca Logo. This image is
available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE
LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN
ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS
INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES
REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR
DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE
COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY
COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS
AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE
to be bound by the terms of this license. To the extent this license may
be considered to be a contract, the licensor grants you the rights
contained here in consideration of your acceptance of such terms and
conditions.

1. Definitions
a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or
lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit
as required by Section 4(b), as requested. If You create an
Adaptation, upon notice from any Licensor You must, to the extent
practicable, remove from the Adaptation any credit as required by
Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or
Collections, You must, unless a request has been made pursuant to
Section 4(a), keep intact all copyright notices for the Work and
provide, reasonable to the medium or means You are utilizing: (i) the
name of the Original Author (or pseudonym, if applicable) if supplied,
and/or if the Original Author and/or Licensor designate another party
or parties (e.g., a sponsor institute, publishing entity, journal) for
attribution ("Attribution Parties") in Licensor's copyright notice,
terms of service or by other reasonable means, the name of such party
or parties; (ii) the title of the Work if supplied; (iii) to the
extent reasonably practicable, the URL if any, that Licensor
specifies to be associated with the Work, unless such URI does not
refer to the copyright notice or licensing information for the Work;
and (iv), consistent with Section 3(b), in the case of an Adaptation,
a credit identifying the use of the Work in the Adaptation (e.g.,
"French translation of the Work by Original Author," or "Screenplay
based on original Work by Original Author"). The credit required by
this Section 4(b) may be implemented in any reasonable manner;
provided, however, that in the case of a Adaptation or Collection, at
a minimum such credit will appear, if a credit for all contributing
authors of the Adaptation or Collection appears, then as part of these
credits and in a manner at least as prominent as the credits for the
other contributing authors. For the avoidance of doubt, You may only
use the credit required by this Section for the purpose of attribution
in the manner set out above and, by exercising Your rights under this
License, You may not implicitly or explicitly assert or imply any
connection with, sponsorship or endorsement by the Original Author,
Licensor and/or Attribution Parties, as appropriate, of You or Your
use of the Work, without the separate, express prior written
permission of the Original Author, Licensor and/or Attribution
Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be
otherwise permitted by applicable law, if You Reproduce, Distribute or
Publicly Perform the Work either by itself or as part of any
Adaptations or Collections, You must not distort, mutilate, modify or
take other derogatory action in relation to the Work which would be
prejudicial to the Original Author's honor or reputation. Licensor
agrees that in those jurisdictions (e.g. Japan), in which any exercise
of the right granted in Section 3(b) of this License (the right to
make Adaptations) would be deemed to be a distortion, mutilation,
modification or other derogatory action prejudicial to the Original
Author's honor and reputation, the Licensor will waive or not assert,
as appropriate, this Section, to the fullest extent permitted by the
applicable national law, to enable You to reasonably exercise Your
right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONcerning THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark “Creative Commons” or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons’ then-current trademark usage
guidelines, as may be published on its website or otherwise made
available upon request from time to time. For the avoidance of doubt,
this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.

For: hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs
/server/datanode/checker/AbstractFuture.java and
hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs
/server/datanode/checker/TimeoutFuture.java

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License. You may obtain a copy of
the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License
for the specific language governing permissions and limitations under
the License.

1.407 ienoopen 1.0.0
1.407.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014, 2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.408 libtiff 4.0.6 1ubuntu0.7
1.408.1 Available under license :
Copyright (c) 1988-1997 Sam Leffler
Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.409 kafka-raft 2.8.0
1.409.1 Available under license :
Apache Kafka
Copyright 2021 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

This distribution has a binary dependency on jersey, which is available under the CDDL License. The source code of jersey can be found at https://github.com/jersey/jersey/.

The streams-scala (streams/streams-scala) module was donated by Lightbend and the original code was copyrighted by them:
Copyright (C) 2018 Lightbend Inc. <https://www.lightbend.com>
Copyright (C) 2017-2018 Alexis Seigneurin.
This project contains the following code copied from Apache Hadoop:
clients/src/main/java/org/apache/kafka/common/utils/PureJavaCrc32C.java
Some portions of this file Copyright (c) 2004-2006 Intel Corporation and licensed under the BSD license.

This project contains the following code copied from Apache Hive:
streams/src/main/java/org/apache/kafka/streams/state/internals/Murmur3.java

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.410 d-bus 1.10.6-1ubuntu3.5

1.410.1 Available under license:

D-Bus is licensed to you under your choice of the Academic Free License version 2.1, or the GNU General Public License version 2 (or, at your option any later version).

Both licenses are included here. Some of the standalone binaries are under the GPL only; in particular, but not limited to, tools/dbus-cleanup-sockets.c and test/decode-gcov.c. Each source code file is marked with the proper copyright information - if you find a file that isn't marked please bring it to our attention.

The Academic Free License
v. 2.1

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

a) to reproduce the Original Work in copies;

b) to prepare derivative works ("Derivative Works") based upon the Original Work;
c) to distribute copies of the Original Work and Derivative Works to the public;

d) to perform the Original Work publicly; and

e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution
Notice reasonably calculated to inform recipients that You have
modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor
warrants that the copyright in and to the Original Work and the patent
rights granted herein by Licensor are owned by the Licensor or are
sublicensed to You under the terms of this License with the permission
of the contributor(s) of those copyrights and patent rights. Except as
expressly stated in the immediately proceeding sentence, the Original
Work is provided under this License on an "AS IS" BASIS and WITHOUT
WARRANTY, either express or implied, including, without limitation,
the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL
WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential
part of this License. No license to Original Work is granted hereunder
except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal
time, whether in tort (including negligence), contract, or
otherwise, shall the Licensor be liable to any person for any direct,
indirect, special, incidental, or consequential damages of any
character arising as a result of this License or the use of the
Original Work including, without limitation, damages for loss of
goodwill, work stoppage, computer failure or malfunction, or any and
all other commercial damages or losses. This limitation of liability
shall not apply to liability for death or personal injury resulting
from Licensor's negligence to the extent applicable law prohibits such
limitation. Some jurisdictions do not allow the exclusion or
limitation of incidental or consequential damages, so this exclusion
and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the
Original Work or a Derivative Work, You must make a reasonable effort
under the circumstances to obtain the express assent of recipients to
the terms of this License. Nothing else but this License (or another
written agreement between Licensor and You) grants You permission to
create Derivative Works based upon the Original Work or to exercise
any of the rights granted in Section 1 herein, and any attempt to do
so except under the terms of this License (or another written
agreement between Licensor and You) is expressly prohibited by
U.S. copyright law, the equivalent laws of other countries, and by
international treaty. Therefore, by exercising any of the rights
granted to You in Section 1 herein, You indicate Your acceptance of
this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate
automatically and You may no longer exercise any of the rights granted
to You by this License as of the date You commence an action,
11) Jurisdiction, Venue and Governing Law. Any action or suit relating
to this License may be brought only in the courts of a jurisdiction
wherein the Licensor resides or in which Licensor conducts its primary
business, and under the laws of that jurisdiction excluding its
conflict-of-law provisions. The application of the United Nations
Convention on Contracts for the International Sale of Goods is
expressly excluded. Any use of the Original Work outside the scope of
this License or after its termination shall be subject to the
requirements and penalties of the U.S. Copyright Act, 17 U.S.C. 101
et seq., the equivalent laws of other countries, and international
treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License
or seeking damages relating thereto, the prevailing party shall be
entitled to recover its costs and expenses, including, without
limitation, reasonable attorneys' fees and costs incurred in
connection with such action, including any appeal of such action. This
section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement
concerning the subject matter hereof. If any provision of this License
is held to be unenforceable, such provision shall be reformed only to
the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this
License, whether in upper or lower case, means an individual or a
legal entity exercising rights under, and complying with all of the
terms of, this License. For legal entities, "You" includes any entity
that controls, is controlled by, or is under common control with
you. For purposes of this definition, "control" means (i) the power,
direct or indirect, to cause the direction or management of such
entity, whether by contract or otherwise, or (ii) ownership of fifty
percent (50%) or more of the outstanding shares, or (iii) beneficial
ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not
otherwise restricted or conditioned by this License or by law, and
Licensor promises not to interfere with or be responsible for such
uses by You.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights
reserved. Permission is hereby granted to copy and distribute this
license without modification. This license may not be modified without
the express written permission of its copyright owner.

---

END OF ACADEMIC FREE LICENSE. The following is intended to describe the essential
differences between the Academic Free License (AFL) version 1.0 and other
open source licenses:

The Academic Free License is similar to the BSD, MIT, UoI/NCSA and Apache
licenses in many respects but it is intended to solve a few problems with
those licenses.

* The AFL is written so as to make it clear what software is being
  licensed (by the inclusion of a statement following the copyright notice
  in the software). This way, the license functions better than a template
  license. The BSD, MIT and UoI/NCSA licenses apply to unidentified software.

* The AFL contains a complete copyright grant to the software. The BSD
  and Apache licenses are vague and incomplete in that respect.

* The AFL contains a complete patent grant to the software. The BSD, MIT,
  UoI/NCSA and Apache licenses rely on an implied patent license and contain
  no explicit patent grant.

* The AFL makes it clear that no trademark rights are granted to the
  licensor's trademarks. The Apache license contains such a provision, but the
  BSD, MIT and UoI/NCSA licenses do not.

* The AFL includes the warranty by the licensor that it either owns the
  copyright or that it is distributing the software under a license. None of
  the other licenses contain that warranty. All other warranties are disclaimed,
  as is the case for the other licenses.

* The AFL is itself copyrighted (with the right granted to copy and distribute
  without modification). This ensures that the owner of the copyright to the
  license will control changes. The Apache license contains a copyright notice,
  but the BSD, MIT and UoI/NCSA licenses do not.

---

START OF GNU GENERAL PUBLIC LICENSE

---

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any
The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and “any
later version”, you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.411 dom-serializer 0.1.1

1.411.1 Available under license:
License

(The MIT License)

Copyright (c) 2014 The cheeriojs contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.412 safe-buffer 5.1.1 5.1.1

1.412.1 Available under license:

The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.413 p-limit 1.3.0

1.413.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.414 slice-ansi 0.0.4

1.414.1 Available under license:
(The MIT License)

Copyright (c) 2015 DC <threedeecee@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
 TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.415 depd 1.1.0 1.1.0

1.415.1 Available under license:
(The MIT License)

Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.416 jaxb-api 2.2.11
1.416.1 Available under license:

Found license 'General Public License 2.0' in
* Copyright (c) 2004-2013 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 2003-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in 'Copyright (c) 2004-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in 'Copyright (c) 2006-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in '# Copyright (c) 2003-2013 Oracle and/or its affiliates. All rights reserved. # The contents of this file are subject to the terms of either the GNU # General Public License Version 2 only ("GPL") or the Common Development # and Distribution License("CDDL") (collectively, the "License"). You # may not use this file except in compliance with the License. You can # Oracle designates this particular file as subject to the "Classpath" # exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 2005-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in 'Copyright (c) 2005-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to
the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'
Found license 'General Public License 2.0' in /* Copyright (c) 2003-2013 Oracle and/or its affiliates. All rights
reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2
only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'
Found license 'General Public License 2.0' in /* Copyright (c) 2003-2013 Oracle and/or its affiliates. All rights
reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2
only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'
Found license 'General Public License 2.0' in /* Copyright (c) 2003-2013 Oracle and/or its affiliates. All rights
reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2
only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'
Found license 'General Public License 2.0' in /* Copyright (c) 2003-2013 Oracle and/or its affiliates. All rights
reserved. The contents of this file are subject to the terms of either the GNU General
Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL")
(collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle
designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2
section of the License'
Found license 'General Public License 2.0' in 'Copyright (c) 2003-2013 Oracle and/or its affiliates. All rights
reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only
("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may
not use this file except in compliance with the License. You can Oracle designates this particular file as subject to
the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

1.417 jsbn 0.1.1
1.417.1 Available under license:
Licensing
-------

This software is covered under the following copyright:

/*
 * Copyright (c) 2003-2005 Tom Wu
 * All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL TOM WU BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

In addition, the following condition applies:

All redistributions must retain an intact copy of this copyright notice and disclaimer.

Address all questions regarding this license to:

Tom Wu
tjw@cs.Stanford.EDU

1.418 uuid 3.3.3
1.418.1 Available under license:

The MIT License (MIT)

Copyright (c) 2010-2016 Robert Kieffer and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.419 is-glob 3.1.0

1.419.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.420 serve-static 1.10.3 1.10.3

1.420.1 Available under license:
(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.421 fast-levenshtein 2.0.6
1.421.1 Available under license:
(MIT License)

Copyright (c) 2013 [Ramesh Nair](http://www.hiddentao.com/)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.422 jwa 1.4.1
1.422.1 Available under license:
Copyright (c) 2013 Brian J. Brennan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.423 babel-helper-explode-class 6.24.1

1.423.1 Available under license:

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

MIT License

Copyright (c) 2014-2017 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2015 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
1.424 is-ip 1.0.0
1.424.1 Available under license :
MIT

1.425 kerb-crypto 1.0.0
1.425.1 Available under license :

Kerby-kerb Crypto
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.426 parsejson 0.0.3 0.0.3

1.426.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.427 commons-jexl 2.1.1

1.427.1 Available under license:

Apache Commons JEXL
Copyright 2001-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.428 finalhandler 1.1.2

1.428.1 Available under license :

(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.429 slf4j-log4j 1.7.32
1.429.1 Available under license :

/**
 * Copyright (c) 2004-2011 QOS.ch
 * All rights reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
 * LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
 * OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
 * WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 *
*/

1.430 faker 4.1.0 4.1.0
1.430.1 Available under license :
faker.js - Copyright (c) 2010-2015
Matthew Bergman & Marak Squires
http://github.com/marak/faker.js/

faker.js was inspired by and has used data definitions from:

* https://github.com/stympy/faker/ - Copyright (c) 2007-2010 Benjamin Curtis
* http://search.cpan.org/~jasonk/Data-Faker-0.07/ - Copyright 2004-2005 by Jason Kohles
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.431 snapdragon-node 2.1.1

1.431.1 Available under license:

The MIT License (MIT)

Copyright (c) 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.432 jsreport-templates 2.4.0

1.432.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library.

Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:
a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.433 kerb-core 1.0.0

1.433.1 Available under license:

Kerby-kerb core
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.434 hk2-utils 2.6.1

1.434.1 Available under license:

# Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
   Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims
brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

   iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.
3.2 When the Program is Distributed as Source Code:

   a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

   b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility
alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it
fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.
The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so
Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to
distribute corresponding source code. (This alternative is allowed
only for noncommercial distribution and only if you received the
program in object code or executable form with such an offer, in
accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source code
means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to control
compilation and installation of the executable. However, as a special
exception, the source code distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies the
executable.

If distribution of executable or object code is made by offering access
to copy from a designated place, then offering equivalent access to copy
the source code from the same place counts as distribution of the source
code, even though third parties are not compelled to copy the source
along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt otherwise
to copy, modify, sublicense or distribute the Program is void, and will
automatically terminate your rights under this License. However, parties
who have received copies, or rights, from you under this License will
not have their licenses terminated so long as such parties remain in
full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and all
its terms and conditions for copying, distributing or modifying the
Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further restrictions
on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications...
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

---

## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is
making a combined work based on this library. Thus, the terms and
conditions of the GNU General Public License version 2 cover the whole
combination.

As a special exception, the copyright holders of this library give you
permission to link this library with independent modules to produce an
executable, regardless of the license terms of these independent
modules, and to copy and distribute the resulting executable under
terms of your choice, provided that you also meet, for each linked
independent module, the terms and conditions of the license of that
module. An independent module is a module which is not derived from or
based on this library. If you modify this library, you may extend this
exception to your version of the library, but you are not obligated to
do so. If you do not wish to do so, delete this exception statement
from your version.

# Notices for Eclipse GlassFish

This content is produced and maintained by the Eclipse GlassFish project.

* Project home: https://projects.eclipse.org/projects/ee4j.glassfish

## Trademarks

Eclipse GlassFish, and GlassFish are trademarks of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For
more information regarding authorship of content, please consult the listed
source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms
of the Eclipse Public License v. 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made
available under the following Secondary Licenses when the conditions for such
availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU
General Public License, version 2 with the GNU Classpath Exception which is
SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/glassfish-ha-api
* https://github.com/eclipse-ee4j/glassfish-logging-annotation-processor
* https://github.com/eclipse-ee4j/glassfish-shoal
* https://github.com/eclipse-ee4j/glassfish-cdi-porting-tck
* https://github.com/eclipse-ee4j/glassfish-jsftemplating
* https://github.com/eclipse-ee4j/glassfish-hk2-extra
* https://github.com/eclipse-ee4j/glassfish-hk2
* https://github.com/eclipse-ee4j/glassfish-fighterfish

## Third-party Content

This project leverages the following third party content.

None

## Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

---

### 1.435 glib-networking 2.48.2 1~ubuntu16.04.1

1.435.1 Available under license:

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributer's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.
Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don’t assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".
A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.
Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).
To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.
The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.
In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/

Files: *
Copyright: 2009-2010 Red Hat, Inc.
    2010 Collabora, Ltd.
License: LGPL-2+

Files: po/bg.po
    po/eo.po
    po/sl.po
    po/sv.po
    po/th.po
    po/tr.po
Copyright: 2011-2013 Free Software Foundation
License: LGPL-2+

Files: po/fi.po
Copyright: 2011 Tommi Vainikainen
License: LGPL-2+

Files: po/te.po
Copyright: 2011-2012 Swecha Telugu Localisation team
License: LGPL-2+

Files: po/vi.po
Copyright: 2011 Free Software Foundation
License: LGPL-2+

Files: tls/tests/mock-pkcs11.*
Copyright: 2010 Stefan Walter
    2011 Collabora Ltd.
License: LGPL-2.1+

License: LGPL-2+
This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library. If not, see <http://www.gnu.org/licenses/>

On Debian systems, the complete text of the GNU General
Public License version 2 can be found in "/usr/share/common-licenses/
LGPL-2".

License: LGPL-2.1+
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library. If not, see <http://www.gnu.org/licenses/>

On Debian systems, the complete text of the GNU General
Public License version 2 can be found in "/usr/share/common-licenses/
LGPL-2.1".

1.436 nspr 4.13.1 0ubuntu0.16.04.1
1.436.1 Available under license :
Mozilla Public License Version 2.0
==================================
1. Definitions
------------------
1.1. "Contributor"
    means each individual or legal entity that creates, contributes to
    the creation of, or owns Covered Software.

1.2. "Contributor Version"
    means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached
the notice in Exhibit A, the Executable Form of such Source Code
Form, and Modifications of such Source Code Form, in each case
including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described
in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of
version 1.1 or earlier of the License, but not also under the
terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in
a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible,
whether at the time of the initial grant or subsequently, any and
all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to,
deletion from, or modification of the contents of Covered
Software; or

(b) any new file in Source Code Form that contains any Covered
Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method,
process, and apparatus claims, in any patent Licensable by such
Contributor that would be infringed, but for the grant of the
License, by the making, using, selling, offering for sale, having
made, import, or transfer of either its Contributions or its
Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU
Lesser General Public License, Version 2.1, the GNU Affero General
Public License, Version 3.0, or any later versions of those
licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this
License. For legal entities, "You" includes any entity that
controls, is controlled by, or is under common control with You. For
purposes of this definition, "control" means (a) the power, direct
or indirect, to cause the direction or management of such entity,
whether by contract or otherwise, or (b) ownership of more than
fifty percent (50%) of the outstanding shares or beneficial
ownership of such entity.

2. License Grants and Conditions
---------------------------------

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free,
non-exclusive license:

(a) under intellectual property rights (other than patent or trademark)
Licensable by such Contributor to use, reproduce, make available,
modify, display, perform, distribute, and otherwise exploit its
Contributions, either on an unmodified basis, with Modifications, or
as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer
for sale, have made, import, and otherwise transfer either its
Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution
become effective for each Contribution on the date the Contributor first
distributes such Contribution.
2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under
this License. No additional rights or licenses will be implied from the
distribution or licensing of Covered Software under this License.
Notwithstanding Section 2.1(b) above, no patent license is granted by a
Contributor:

(a) for any code that a Contributor has removed from Covered Software;
or

(b) for infringements caused by: (i) Your and any other third party’s
modifications of Covered Software, or (ii) the combination of its
Contributions with other software (except as part of its Contributor
Version); or

(c) under Patent Claims infringed by Covered Software in the absence of
its Contributions.

This License does not grant any rights in the trademarks, service marks,
or logos of any Contributor (except as may be necessary to comply with
the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to
distribute the Covered Software under a subsequent version of this
License (see Section 10.2) or under the terms of a Secondary License (if
permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its
Contributions are its original creation(s) or it has sufficient rights
to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under
applicable copyright doctrines of fair use, fair dealing, or other
equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted
in Section 2.1.
3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.
3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to
You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is" basis, *
*  without warranty of any kind, either expressed, implied, or statutory, *
*  including, without limitation, warranties that the Covered Software *
*  is free of defects, merchantable, fit for a particular purpose or *
*  non-infringing. The entire risk as to the quality and performance of *
*  the Covered Software is with You. Should any Covered Software prove *
*  defective in any respect, You (not any Contributor) assume the cost *
*  of any necessary servicing, repair, or correction. This disclaimer *
*  of warranty constitutes an essential part of this License. No use of *
*  any Covered Software is authorized under this License except under *
*  this disclaimer.                                                    *
*                                                                      *
************************************************************************
************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*                                                                      *
*  Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.  

*
8. Litigation
-------------

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous
----------------

This License represents the complete agreement concerning the subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. Any law or regulation which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License
----------------------------

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section
10.3, no one other than the license steward has the right to modify or
publish new versions of this License. Each version will be given a
distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version
of the License under which You originally received the Covered Software,
or under the terms of any subsequent version published by the license
steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to
create a new license for such software, you may create and use a
modified version of this License if you rename the license and remove
any references to the name of the license steward (except to note that
such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary
Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

1.437 encoding 0.1.12 0.1.12

1.437.1 Available under license:

Copyright (c) 2012-2014 Andris Reinman

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.438 snake-yaml 1.27
1.438.1 Available under license:

Found license 'GNU Lesser General Public License' in 'This module is multi-licensed and may be used under the terms EPL, Eclipse Public License, V1.0 or later, http://www.eclipse.org/legal LGPL, GNU Lesser General Public License, V2.1 or later, http://www.gnu.org/licenses/lgpl.html GPL, GNU General Public License, V2 or later, http://www.gnu.org/licenses/gpl.html AL, Apache License, V2.0 or later, http://www.apache.org/licenses BSD, BSD License, http://www.opensource.org/licenses/bsd-license.php * Multi-licensed: EPL / LGPL / GPL / AL / BSD.'

Found license 'Eclipse Public License 1.0' in 'This module is multi-licensed and may be used under the terms EPL, Eclipse Public License, V1.0 or later, http://www.eclipse.org/legal LGPL, GNU Lesser General Public License, V2.1 or later, http://www.gnu.org/licenses/lgpl.html GPL, GNU General Public License, V2 or later, http://www.gnu.org/licenses/gpl.html AL, Apache License, V2.0 or later, http://www.apache.org/licenses BSD, BSD License, http://www.opensource.org/licenses/bsd-license.php * Multi-licensed: EPL / LGPL / GPL / AL / BSD.'

1.439 avro-ipc 1.9.1
1.439.1 Available under license:

Apache Avro IPC
Copyright 2009-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control"
means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and

Open Source Used In DNAC Platform Antelope 1677
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

---

Open Source Used In DNAC Platform Antelope 1678
as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.440.1 Available under license:
MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.441 lodash-groupby 4.6.0

1.441.1 Available under license:
Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.442 @babel/plugin-proposal-unicode-property-regex 7.12.13

1.442.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.443 jackson-xc 2.10.4

1.443.1 Available under license:

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
This copy of Jackson JSON processor `jackson-module-jaxb-annotations` module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.444 ansicolors 0.2.1

1.444.1 Available under license:

Copyright 2013 Thorsten Lorenz.
All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.445 commons-cli 1.2
1.445.1 Available under license:

Apache Commons CLI
Copyright 2001-2009 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the

Open Source Used In DNAC Platform Antelope 1684
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.446 decode-uri-component 0.2.0

1.446.1 Available under license :

The MIT License (MIT)

Copyright (c) Sam Verschueren <sam.verschueren@gmail.com> (github.com/SamVerschueren)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.447 de.huxhorn.sulky.formatting 0.9.17

1.447.1 Available under license:
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.
A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding
Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy’s public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.448 log-symbols 3.0.0
1.448.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.449 object.getownpropertydescriptors 2.1.2

1.449.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.450 twill-zookeeper 0.6.0-incubating

1.450.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2-jar/org/apache/twill/zookeeper/AbstractZKClient.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2-jar/org/apache/twill/zookeeper/ACLDATA.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2-jar/org/apache/twill/permits/BasicACLData.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2-jar/org/apache/twill/permits/ReentrantDistributedLock.java

No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/zookeeper/ForwardingZKClient.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/internal/zookeeper/BasicNodeChildren.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/zookeeper/OperationFuture.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/forwarding/ZKClient.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/forwarding/ZKClientService.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/forwarding/ZKClientServices.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/internal/zookeeper/package-info.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/internal/zookeeper/BasicNodeData.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/internal/zookeeper/KillZKSession.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/internal/zookeeper/LeaderElection.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/internal/zookeeper/LeaderElection.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/intern/zookeeper/NodeData.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/Internal/zookeeper/RetryUtil.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/zookeeper/package-info.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/internal/zookeeper/RewatchOnExpireWatcher.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/Internal/zookeeper/FailureRetryZKClient.java
* /opt/cola/permits/1137746061_1614121450.21/0/twill-zookeeper-0-6-0-incubating-sources-2.jar/org/apache/twill/zookeeper/RetryStrategies.java
1.451 engine.io-parser 2.2.1

1.451.1 Available under license:

(The MIT License)

Copyright (c) 2016 Guillermo Rauch (@rauchg)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.452 parquet-column 1.8.1

1.452.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

------------------------------------------------------------------------

This product depends on fastutil and includes it in this binary artifact.
Fastutil provides type-specific collection implementations.

Copyright: 2002-2014 Sebastiano Vigna
Home page: http://fasutil.di.unimi.it/
License: http://www.apache.org/licenses/LICENSE-2.0.html

Apache Parquet Column
Copyright 2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.453 istanbuljs-schema 0.1.3
1.453.1 Available under license:

MIT License

Copyright (c) 2019 CFWare, LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.454 prop-types 15.6.2 15.6.2

1.454.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.455 jcl-over-slf4j 1.7.26

1.456 utf-8-validate 1.2.1
1.456.1 Available under license :
   No license file was found, but licenses were detected in source scan.

    /*! 
    * UTF-8 validate: UTF-8 validation for WebSockets.
    * Copyright(c) 2015 Einar Otto Stangvik <einaros@gmail.com>
    * MIT Licensed
    */

   Found in path(s):
   * /opt/cola/permits/1110633896_1606843999.83/0(utf-8-validate-1-2-1-1-tgz/package/fallback.js
   * /opt/cola/permits/1110633896_1606843999.83/0(utf-8-validate-1-2-1-1-tgz/package/src/validation.cc

1.457 intl-messageformat 2.2.0 2.2.0
1.457.1 Available under license :
   Copyright 2013 Yahoo! Inc.
   All rights reserved.

   Redistribution and use in source and binary forms, with or without
   modification, are permitted provided that the following conditions are met:

   * Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.

   * Redistributions in binary form must reproduce the above copyright
     notice, this list of conditions and the following disclaimer in the
     documentation and/or other materials provided with the distribution.

   * Neither the name of the Yahoo! Inc. nor the
     names of its contributors may be used to endorse or promote products
     derived from this software without specific prior written permission.

   THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
   ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
   WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Pluralization rules built from
https://github.com/papandreou/node-cldr
which is licensed under the BSD license and has the following license:

Copyright (c) 2012, Andreas Lind Petersen
All rights reserved.

See the following for more details:
https://github.com/papandreou/node-cldr/blob/master/LICENSE

1.458 amcharts3-export 1.4.73 1.4.73

1.458.1 Available under license:
Copyright (C) 2012-2015  SheetJS

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Except where noted, this license applies to any and all software programs and associated documentation files created
by the Original Author and distributed with the Software:

'jszip.js' is a modified version of JSZip, Copyright (c) Stuart Knightley, David Duponchel, Franz Buchinger,
Antonio Afonso.  JSZip is dual licensed and is used according to the terms of the MIT License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a counterclaim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.459 write 0.2.1

1.459.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.460 source-map 0.6.1 0.6.1

1.460.1 Available under license:

Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.
* Neither the names of the Mozilla Foundation nor the names of project contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.461 swagger-annotations 1.5.16

1.461.1 Available under license:

```java
/**
 * Copyright 2016 SmartBear Software
 * <p>
 * Licensed under the Apache License, Version 2.0 (the "License);
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * <p>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p>
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and limitations under the License.
 */

package io.swagger.annotations;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
 * License metadata available within the info section of a Swagger definition, see
 * https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md#licenseObject
 * @since 1.5.0
 */```
@Target(ElementType.ANNOTATION_TYPE)
@Retention(RetentionPolicy.RUNTIME)
public @interface License {

/**
 * The name of the license.
 *
 * @return the name of the license
 */
String name();

/**
 * An optional URL for the license.
 *
 * @return an optional URL for the license.
 */
String url() default "";
}

1.462 is-utf8 0.2.1

1.462.1 Available under license:

The MIT License (MIT)

Copyright (C) 2014 Wei Fanzhe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.463 drill-format-ltsv 1.19.0
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object...
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.464 has-ansi 2.0.0

1.464.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.465 @babel/code-frame 7.10.4
1.465.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.466 vary 1.1.2 1.1.2
1.466.1 Available under license:
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
1.467 **mime-db 1.44.0**

1.467.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.468 **drill-storage-cassandra 1.19.0**

1.468.1 Available under license:

Drill : Contrib : Storage : Cassandra

Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.469 jffi 1.2.15

1.469.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.
1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.470 restore-cursor 2.0.0
1.470.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.471 jsreport-studio-theme-dark 0.3.0

1.471.1 Available under license:
MIT License

Copyright (c) 2019 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.


1.472.1 Available under license:
MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.473 arr-flatten 1.1.0
1.473.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.474 mail 1.5.6
1.474.1 Available under license :
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a
Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

   B. Any new file that contains any part of the Original Software or previous Modification; or

   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a
medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under
any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF
SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the
Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY
COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE
PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL,
SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY
TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER
PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to
achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to
most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a
pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public
License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later
version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the
Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and
you are welcome to redistribute it under certain conditions; type `show c' for details.
The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.475 jnr-posix 3.0.41
1.475.1 Available under license:

Found license 'GNU Lesser General Public License' in 'License Version 1.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of Software distributed under the License is distributed on an "AS Alternatively, the contents of this file may be used under the terms of either of the GNU General Public License Version 2 or later (the "GPL"), or the GNU Lesser General Public License Version 2.1 or later (the "LGPL"), use your version of this file under the terms of the CPL, indicate your'
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein,
no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual
property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any
other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights
and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual
property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute
the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any,
to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided
that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including
warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and
fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special,
incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by
any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to
obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows
subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners
and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner which does not create potential
liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient
institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.476 source-map 0.6.0 0.6.0

1.476.1 Available under license :

Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the names of the Mozilla Foundation nor the names of project contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.477 nyc 15.1.0
1.477.1 Available under license:
ISC License

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.478 estraverse 4.2.0
1.478.1 Available under license:
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.479 swagger 2.0.7
1.479.1 Available under license :
Copyright 2018 SmartBear Software

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at [apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0)

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.480 parse-glob 3.0.4
1.480.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.481 array-flatten 1.1.1 1.1.1

1.481.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.482 install 0.12.2 0.12.2

1.482.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Benjamin Newman <ben@benjamn.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.483 is-stream 1.1.0 1.1.0
1.483.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.484 asm 3.3.1
1.484.1 Available under license:

No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2007 INRIA, France Telecom
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
* 1. Redistributions of source code must retain the above copyright
   * notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
   * notice, this list of conditions and the following disclaimer in the
   * documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
   * contributors may be used to endorse or promote products derived from
   * this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/commons/JSRInlinerAdapter.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/MethodNode.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/analysis/SmallSet.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/ClassReader.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/TableSwitchInsnNode.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/MethodAdapter.java
jar/org/objectweb/asm/tree/TryCatchBlockNode.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/analysis/BasicInterpreter.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/MultiANewArrayInsnNode.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/MethodVisitor.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/ByteVector.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/IntInsnNode.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/signature/SignatureVisitor.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/commons/AnalyzerAdapter.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/FieldInsnNode.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/commons/AdviceAdapter.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/AbstractInsnNode.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/ClassNode.java
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/InnerClassNode.java

Open Source Used In DNAC Platform Antelope 1750
No license file was found, but licenses were detected in source scan.

/**
 * ASM XML Adapter
 * Copyright (c) 2004, Eugene Kuleshov
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */

Found in path(s):
  * /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/xml/SAXCodeAdapter.java
  * /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/xml/Processor.java
  * /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/xml/SAXClassAdapter.java
  * /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/xml/SAXAdapter.java
No license file was found, but licenses were detected in source scan.

* ASM XML Adapter
  * Copyright (c) 2004, Eugene Kuleshov
  * All rights reserved.
  *
  * Redistribution and use in source and binary forms, with or without
  * modification, are permitted provided that the following conditions
  * are met:
  * 1. Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.
  * 2. Redistributions in binary form must reproduce the above copyright
     notice, this list of conditions and the following disclaimer in the
     documentation and/or other materials provided with the distribution.
  * 3. Neither the name of the copyright holders nor the names of its
     contributors may be used to endorse or promote products derived from
     this software without specific prior written permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
  * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
  * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
  * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
  * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
  * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
  * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
  * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
  * THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/xml/package.html

No license file was found, but licenses were detected in source scan.

2005 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/package.html
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/commons/package.html
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/util/package.html
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/attrs/package.html
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/tree/analysis/package.html
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/package.html
* /opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/signature/package.html

No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2005 INRIA, France Telecom
* All rights reserved.
* Redistributions and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
***/
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
/opt/cola/permits/1135935623_1613639042.9/0/asm-all-3-3-1-sources-6-jar/org/objectweb/asm/commons/InstructionAdapter.java
No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2007 INRIA, France Telecom
 * All rights reserved.
 * 
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 * contributors may be used to endorse or promote products derived from
 * this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */
ASM XML Adapter
Copyright (c) 2004, Eugene Kuleshov
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Curator Framework
Copyright 2011-2020 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.486 inherits 2.0.3

1.486.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

1.487 mime 2.6.0

1.487.1 Available under license:
The MIT License (MIT)

Copyright (c) 2017 Ronomon

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.488 lodash-reject 4.6.0

1.488.1 Available under license :

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====
Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.489 mimic-response 1.0.1

1.489.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.490 object-assign 4.1.1 4.1.1

1.490.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.491 dashdash 1.14.1
1.491.1 Available under license:

# This is the MIT license

Copyright (c) 2013 Trent Mick. All rights reserved.
Copyright (c) 2013 Joyent Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.492 st4 4.1
1.492.1 Available under license:

/*
 * [The "BSD license"]
 * Copyright (c) 2011 Terence Parr
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 *    derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

1.493 okhttp 2.7.5

1.493.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by its Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date
such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.494 common-utils 6.1.1

1.494.1 Available under license:

The following libraries are included in packaged versions of this project:

* Apache ZooKeeper
  * COPYRIGHT: Copyright 2009-2014 The Apache Software Foundation
  * LICENSE: licenses/LICENSE.apache2.txt
  * NOTICE: licenses/NOTICE.zookeeper.txt
  * HOMEPAGE: http://zookeeper.apache.org/

* jline
  * COPYRIGHT: Copyright (c) 2002-2006, Marc Prud'hommeaux <mwp1@cornell.edu>
  * LICENSE: licenses/LICENSE.bsd.txt
  * HOMEPAGE: http://jline.sourceforge.net/

* SLF4J
  * COPYRIGHT: Copyright (c) 2004-2013 QOS.ch
  * LICENSE: licenses/LICENSE.mit.txt
  * HOMEPAGE: http://www.slf4j.org/

* ZkClient
  * LICENSE: licenses/LICENSE.apache2.txt
  * HOMEPAGE: https://github.com/sgroschupf/zkclient

Apache ZooKeeper
Copyright 2009-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.495 format-util 1.0.5 1.0.5

1.495.1 Available under license:

Copyright 2015 Freeform Systems and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

NO WARRANTY OR LIABILITY

YOU EXPRESSLY AGREE THAT DOWNLOADING THE SOFTWARE AND ANY USE OF THE SOFTWARE IS AT YOUR OWN RISK. NO WARRANTY, REPRESENTATION, CONDITION, UNDERTAKING OR TERM - EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE - INCLUDING BUT NOT LIMITED TO THE CONDITION, QUALITY, DURABILITY, PERFORMANCE, ACCURACY, STABILITY, RELIABILITY, NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE OR USE OF THE SOFTWARE IS GIVEN OR ASSUMED BY FREEFORM SYSTEMS LTD. ALL SUCH WARRANTIES, REPRESENTATIONS, CONDITIONS, UNDERTAKINGS AND TERMS ARE HEREBY EXCLUDED. FREEFORM SYSTEMS LTD MAKES NO WARRANTY THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT IT WILL BE UNINTERRUPTED, TIMELY, SECURE, OR ERROR FREE; IN NO EVENT SHALL FREEFORM SYSTEMS LTD BE LIABLE TO ANY PARTY FOR ANY DAMAGES INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR INFORMATION, LOSS OF PROFITS AND SAVINGS AND THE LIKE), OR ANY OTHER DAMAGES ARISING - IN ANY WAY, SHAPE OR FORM - OUT OF THE AVAILABILITY, USE, RELIANCE ON, INABILITY TO UTILIZE OR IMPROPER USE OF THE SOFTWARE EVEN IF FREEFORM
SYSTEMS LTD SHALL HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND REGARDLESS OF THE FORM OF ACTION, WHETHER IN CONTRACT, TORT, OR OTHERWISE. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, THE ABOVE EXCLUSIONS OF INCIDENTAL AND CONSEQUENTIAL DAMAGES MAY NOT APPLY TO YOU.

1.496 string-width 3.1.0
1.496.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.497 ws 1.1.5 1.1.5
1.497.1 Available under license :
## License

(The MIT License)

Copyright (c) 2011 Einar Otto Stangvik &lt;einaros@gmail.com&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
 included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.498 x-xss-protection 1.0.0

1.498.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.499 object-assign 2.1.1 2.1.1

1.499.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights

Open Source Used In DNAC Platform Antelope 1779
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.500 diff 3.5.0
1.500.1 Available under license :
Software License Agreement (BSD License)

Copyright (c) 2009-2015, Kevin Decker <kpdecker@gmail.com>

All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Kevin Decker nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.501 jmespath-java 1.11.375

1.501.1 Available under license:
No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>

Found in path(s):
* /opt/cola/permits/1135892278_1613627038.25/0/jmespath-java-1-11-375-sources-1-jar/META-INF/maven/com.amazonaws/jmespath-java/pom.xml

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892278_1613627038.25/0/jmespath-java-1-11-375-sources-1-jar/com/amazonaws/jmespath/JmesPathLiteral.java
* /opt/cola/permits/1135892278_1613627038.25/0/jmespath-java-1-11-375-sources-1-jar/com/amazonaws/jmespath/JmesPathSubExpression.java
* /opt/cola/permits/1135892278_1613627038.25/0/jmespath-java-1-11-375-sources-1-jar/com/amazonaws/jmespath/OpGreaterThan.java
* /opt/cola/permits/1135892278_1613627038.25/0/jmespath-java-1-11-375-sources-1-jar/com/amazonaws/jmespath/JmesPathLiteral.java
* /opt/cola/permits/1135892278_1613627038.25/0/jmespath-java-1-11-375-sources-1-jar/com/amazonaws/jmespath/JmesPathSubExpression.java
* /opt/cola/permits/1135892278_1613627038.25/0/jmespath-java-1-11-375-sources-1-jar/com/amazonaws/jmespath/OpGreaterThan.java
*/
1.502 archiver 2.1.1

1.502.1 Available under license :

Copyright (c) 2012-2014 Chris Talkington, contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.503 commons-beanutils-core 1.8.0

1.503.1 Available under license :

Apache Commons BeanUtils
Copyright 2000-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
1.504 @babel/plugin-syntax-optional-catch-binding 7.8.3

1.504.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.505 hsqldb-jdbc 1.8.0.10

1.505.1 Available under license:

/*
 * For work developed by the HSQL Development Group:
 *
 * Copyright (c) 2001-2010, The HSQL Development Group
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 * *
 * Redistributions of source code must retain the above copyright notice, this
 * list of conditions and the following disclaimer.
 * *
 * Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* Neither the name of the HSQL Development Group nor the names of its
* contributors may be used to endorse or promote products derived from this
* software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQldb.ORG,
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* For work originally developed by the Hypersonic SQL Group:

* Copyright (c) 1995-2000, The Hypersonic SQL Group.
* All rights reserved.

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this
* list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* Neither the name of the Hypersonic SQL Group nor the names of its
* contributors may be used to endorse or promote products derived from this
* software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP,
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*/
/* Copyright (c) 2001-2010, The HSQL Development Group
/* All rights reserved.
/*
/* Redistribution and use in source and binary forms, with or without
/* modification, are permitted provided that the following conditions are met:
/*
/* Redistributions of source code must retain the above copyright notice, this
/* list of conditions and the following disclaimer.
/*
/* Redistributions in binary form must reproduce the above copyright notice,
/* this list of conditions and the following disclaimer in the documentation
/* and/or other materials provided with the distribution.
/*
/* Neither the name of the HSQL Development Group nor the names of its
/* contributors may be used to endorse or promote products derived from this
/* software without specific prior written permission.
/*
/* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
/* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
/* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
/* ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG,
/* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
/* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
/* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
/* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
/* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
/* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
/* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

1.506 globals 11.12.0

1.506.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.507 extend-shallow 3.0.2

1.507.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015, 2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.508 lodash 4.17.19

1.508.1 Available under license:
Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.509 mime-db 1.35.0 1.35.0

1.509.1 Available under license:

The MIT License (MIT)
Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.510 chrome-page-eval 1.3.0

1.510.1 Available under license:

The MIT License (MIT)

Copyright (c) 2018 bjrmatos

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.511 jasper-compiler 5.5.12

1.511.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.512 commons-compress 2.1.1
1.512.1 Available under license :

Copyright (c) 2014 Chris Talkington, contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.513 caniuse-lite 1.0.30001202
1.513.1 Available under license:
Attribution 4.0 International

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. More considerations for licensors: wiki.creativecommons.org/Considerations_for_licensors
Considerations for the public: By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensor's permission is not necessary for any reason—for example, because of any applicable exception or limitation to copyright—then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public:
wiki.creativecommons.org/Considerations_for_licensees

=======================================================================
Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.
c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

i. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.
Section 2 -- Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

a. reproduce and Share the Licensed Material, in whole or in part; and

b. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a) (4) never produces Adapted Material.

5. Downstream recipients.

a. Offer from the Licensor -- Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

b. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.
6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

   a. retain the following if it is supplied by the Licensor with the Licensed Material:

      i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
ii. a copyright notice;

iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share
all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the
2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 -- Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal
processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the Licensor. The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published atcreativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.

1.514 invert-kv 2.0.0

1.514.1 Available under license :

MIT License

Copyright (c) 2020 Anand Chowdhary <mail@anandchowdhary.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.515 graphics2d 0.30

1.515.1 Available under license:

Copyright (c) 2011-12, vernon adams (vern@newtypography.co.uk), with Reserved Font Names 'Antonio'

This Font Software is licensed under the SIL Open Font License, Version 1.1.
This license is copied below, and is also available with a FAQ at: http://scripts.sil.org/OFL

------------------------------
SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007
------------------------------

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS
"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS
Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.
2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor 
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing 
the Work or Derivative Works thereof, You may choose to offer, 
and charge a fee for, acceptance of support, warranty, indemnity, 
or other liability obligations and/or rights consistent with this 
License. However, in accepting such obligations, You may act only 
on Your own behalf and on Your sole responsibility, not on behalf 
of any other Contributor, and only if You agree to indemnify, 
defend, and hold each Contributor harmless for any liability 
incurred by, or claims asserted against, such Contributor by reason 
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following 
boilerplate notice, with the fields enclosed by brackets "[]" 
replaced with your own identifying information. (Don't include 
the brackets!) The text should be enclosed in the appropriate 
comment syntax for the file format. We also recommend that a 
file or class name and description of purpose be included on the 
same "printed page" as the copyright notice for easier 
identification within third-party archives.

Copyright yyyy [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); 
you may not use this file except in compliance with the License. 
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software 
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.

1.516 guava 30.1.1-jre
1.516.1 Available under license :

    Apache License
    Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.517 babel-plugin-syntax-optional-chaining
7.8.3
1.517.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.518 lodash.defaults 4.2.0
1.518.1 Available under license :
Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history
The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.519 apiguardian-api 1.1.0
1.519.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
Copyright 2002-2017 the original author or authors.
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

1.520 jaxb-core 2.3.1
1.520.1 Available under license:
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
Version 1.0

* 1. Definitions.
   o 1.1. ?Contributor? means each individual or entity that creates or contributes to the creation of Modifications.
   o 1.2. ?Contributor Version? means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
   o 1.3. ?Covered Software? means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
   o 1.4. ?Executable? means the Covered Software in any form other than Source Code.
   o 1.5. ?Initial Developer? means the individual or entity that first makes Original Software available under this License.
   o
1.6. "Larger Work? means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License? means this document.

1.8. "Licensable? means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications? means the Source Code and Executable form of any of the following:

   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   
   B. Any new file that contains any part of the Original Software or previous Modification; or
   
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software? means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims? means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code? means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You? (or ?Your?) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, ?You? includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, ?control? means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of
3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the
terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN ?AS IS? BASIS, WITHOUT
WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT
8. U.S. GOVERNMENT END USERS.

The Covered Software is a ?commercial item,? as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of ?commercial computer software? (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and ?commercial computer software documentation? as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction?s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys? fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

1.521 mime 1.6.0

1.521.1 Available under license :

The MIT License (MIT)

Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.522 colors.js 1.0.3
1.522.1 Available under license :
  Original Library
    - Copyright (c) Marak Squires

  Additional Functionality
    - Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.523 domelementtype 1.1.3 1.1.3
1.523.1 Available under license:
Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.524 httpcomponents-core 4.4.10
1.524.1 Available under license:

Apache HttpCore
Copyright 2005-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.525 drill-storage-kafka 1.19.0

1.525.1 Available under license:

Drill: Contrib: Storage: Kafka
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.526 object-keys 1.0.12

1.526.1 Available under license:

The MIT License (MIT)

Copyright (C) 2013 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.527 commons-net 3.6

1.527.1 Available under license :

Apache Commons Net
Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate.
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include}
the brackets! The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.528 co 4.6.0 4.6.0
1.528.1 Available under license :
(The MIT License)

Copyright (c) 2014 TJ Holowaychuk &lt;tj@vision-media.ca&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.529 topo 1.1.0

1.529.1 Available under license :
The MIT License (MIT)

Copyright (c) 2016 Andreas Deuschlinger

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.530 semver 5.7.1 5.7.1

1.530.1 Available under license :
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.531 jets3t 0.6.1

1.531.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * jets3t : Java Extra-Tasty S3 Toolkit (for Amazon S3 online storage service)
 * This is a java.net project, see https://jets3t.dev.java.net/
 *
 * Copyright 2007 James Murty
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/gui/ItemPropertiesDialog.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/gui/skins/html/SkinnedJMenuItem.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/utils/signedurl/GatekeeperClientUtils.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/apps/cockpit/CLObjectTableModel.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/apps/cockpitlite/GlobalFeatureModel.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/apps/cockpitlite/ToggleAclDialog.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/apps/cockpitlite/ProgressPanel.java
No license file was found, but licenses were detected in source scan.

/*
 * jets3t : Java Extra-Tasty S3 Toolkit (for Amazon S3 online storage service)
 * This is a java.net project, see https://jets3t.dev.java.net/
 *
 * Copyright 2006 James Murty
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
*/

jar/src/org/jets3t/gui/skins/html/SkinnedJTextArea.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/multithread/LookupACLEvent.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/multithread/CopyObjectsEvent.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/acl/EmailAddressGrantee.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/multithread/ServiceEvent.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/S3ObjectsChunk.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/impl/rest/httpclient/RestS3Service.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/model/S3Object.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/utils/ByteFormatter.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/gui/skins/html/SkinnedJCheckBox.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/multithread/S3ObjectsChunk.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/Constants.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/utils/RestUtils.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/multithread/DeleteObjectsEvent.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/gui/HyperlinkActivatedListener.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/apps/cockpit/gui/SignedGetUrlDialog.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/apps/cockpit/gui/LoginLocalFolderPanel.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/apps/cockpit/gui/LoginCredentialsPanel.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/servlets/gatekeeper/GatekeeperServlet.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/multithread/S3ServiceMulti.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/utils/XmlResponsesSaxParser.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/gui/skins/html/SkinnedJHtmlLabel.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/apps/synchronize/Synchronize.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/multithread/S3ServiceMulti.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/service/utils/ByteFormatter.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/jets3t/gui/HyperlinkActivatedListener.java
Open Source Used In DNAC Platform Antelope 1845
jar/src/org/jets3t/servlets/gatekeeper/impl/DefaultUrlSigner.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/gui/skins/html/SkinnedJButton.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/utils/signedurl/SignedUrlAndObject.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/apps/cockpit/gui/LoginPassphrasePanel.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/multithread/CreateObjectsEvent.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/tests/HttpClientRestS3ServiceTest.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/multithread/GetObjectHeadsEvent.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/io/UnrecoverableIOException.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/multithread/DownloadObjectsEvent.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/model/S3BucketLoggingStatus.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/acl/GroupGrantee.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/multithread/S3ServiceEventAdaptor.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/gui/skins/html/SkinnedLookAndFeel.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/gui/skins/html/SkinnedJScrollPane.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/acl/GroupGrantee.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/model/S3BucketLoggingStatus.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/apps/cockpit/gui/AWSCredentialsDialog.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/acl/GranteeInterface.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/multithread/GetObjectHeadsEvent.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/gui/skins/html/SkinnedJProgressbar.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/apps/cockpit/gui/AccessControlDialog.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/io/UnrecoverableIOException.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/multithread/DownloadObjectsEvent.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/apps/cockpit/StartupDialog.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/gui/skins/html/SkinnedJScrollPane.java
* /opt/cola/permits/1139709450_1616468284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/jets3t/service/acl/GroupGrantee.java
jar/src/org/jets3t/service/multithread/S3ServiceSimpleMulti.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/multithread/ThreadWatcher.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/appsuploader/Uploader.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/acl/CanonicalGrantee.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/security/EncryptionUtil.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/security/S3ServiceException.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/acl/AccessControlList.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/model/CreateBucketConfiguration.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/gui/UserInputFields.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/io/InterruptableInputStream.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/multithread/CancelEventTrigger.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/samples/CodeSamples.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/model/BaseS3Object.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/gui/skins/html/SkinnedJRadioButton.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/utils/FileComparer.java
No license file was found, but licenses were detected in source scan.

/*
* $HeadURL: https://svn.apache.org/repos/asf/jakarta/commons/proper/httpclient/branches/HTTPCLIENT_3_0_BRANCH/src/contrib/org/apache/commons/httpclient/contrib/proxy/PluginProxyUtil.java $ *
* $Revision: 1.5 $ *
* $Date: 2007/01/18 11:11:27 $ *
*
* ====================================================================
* Copyright 1999-2004 The Apache Software Foundation
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
*
* /

Found in path(s):
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-jar/src/org/apache/commons/httpclient/contrib/proxy/PluginProxyUtil.java
No license file was found, but licenses were detected in source scan.

/*
* jets3t : Java Extra-Tasty S3 Toolkit (for Amazon S3 online storage service)
* This is a java.net project, see https://jets3t.dev.java.net/
* Copyright 2008 James Murty
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/servlets/gatekeeper/TransactionIdProvider.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/servlets/gatekeeper/impl/DefaultTransactionIdProvider.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/gui/ObjectsAttributesDialog.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/gui/CopyObjectsDialog.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/samples/UrlSigningExample.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t/apps/cockpit/gui/ObjectTableModel.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/io/RepeatableFileInputStream.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/io/RepeatableInputStream.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/apps/cockpit/Cockpit.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/apps/cockpit/gui/BucketTableModel.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/samples/StorePassphraseBasedLogin.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/samples/SamplesUtils.java
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2.jar/src/org/jets3t/service/multithread/ListObjectsEvent.java

No license file was found, but licenses were detected in source scan.

/**
 * $HeadURL:
 * https://svn.apache.org/repos/asf/jakarta/commons/proper/httpclient/branches/HTTPCLIENT_3_0_BRANCH/src/contrib/org/apache/commons/httpclient/contrib/proxy/ProxyDetectionException.java $
 * $Revision: 1.1 $
 * $Date: 2006/11/10 10:36:45 $
 */
Copyright 1999-2004 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This software consists of voluntary contributions made by many
individuals on behalf of the Apache Software Foundation. For more
information on the Apache Software Foundation, please see
<http://www.apache.org>/.

Found in path(s):
/opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-
jar/src/org/apache/commons/httpclient/contrib/proxy/ProxyDetectionException.java
No license file was found, but licenses were detected in source scan.

$HeadURL:
https://svn.apache.org/repos/asf/jakarta/commons/proper/httpclient/branches/HTTPCLIENT_3_0_BRANCH/src/con
trib/org/apache/commons/httpclient/contrib/proxy/PluginProxyTestApplet.java$
$Revision: 1.1 $
$Date: 2006/11/10 10:36:45 $

Copyright 1999-2004 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
* ====================================================================
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* *
*/

Found in path(s):
* /opt/cola/permits/1139709450_1614684284.5/0/jets3t-0-6-1-sources-2-
  jar/src/org/apache/commons/httpclient/contrib/proxy/PluginProxyTestApplet.java

1.532 jacoco-core 0.7.9
1.532.1 Available under license:

Found license 'Eclipse Public License 1.0' in 'All rights reserved. This program and the accompanying materials are
made available under the terms of the Eclipse Public License v1.0'
Found license 'Eclipse Public License 1.0' in 'under the terms and conditions of the Eclipse Public License Version
1.0'

1.533 mime 1.6.0 1.6.0
1.533.1 Available under license:

The MIT License (MIT)

Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software,

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY Claim, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
1.534 animal-sniffer-annotation 1.14

1.535 protobuf-java 3.14.0
1.535.1 Available under license:

No license file was found, but licenses were detected in source scan.

// Copyright 2008 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/LazyStringList.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/RpcCallback.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/google/protobuf/wrappers.proto
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/RepeatedFieldBuilderV3.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/Message.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/RawMessageInfo.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/ProtocolStringList.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/Utf8.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/type.proto
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/CodedOutputStream.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/BufferAllocator.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/ServiceException.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-jar/com/google/protobuf/MessageInfoFactory.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/UnknownFieldSetSchema.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/WireFormat.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/BlockingRpcChannel.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/ExtensionRegistry.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/Service.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/MapFieldSchemaLite.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/CodedInputStreamReader.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/IterableByteBufferInputStream.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/duration.proto
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/GeneratedMessageLite.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/MessageLiteToString.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/Writer.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/MapField.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/LongArrayList.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/ProtocolMessageEnum.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/TextFormatParseLocation.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/BooleanArrayList.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/BinaryWriter.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/SmallSortedMap.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/AbstractMessage.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/MessageOrBuilder.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/InvalidProtocolBufferException.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/google/protobuf/api.proto
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/BinaryReader.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-
jar/com/google/protobuf/UnsafeByteOperations.java
* /opt/cola/permits/1127748132_1611862246.14/0/protobuf-java-3-14-0-sources-

1.536 jctools-core 2.1.1
1.536.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.537 body-parser 1.18.3 1.18.3
1.537.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.538 libxml 2.9.3+dfsg1 1ubuntu0.7
1.538.1 Available under license:

This package was debianized by Vincent Renardias <vincent@waw.com> on Sat, 26 Sep 1998 16:50:54 +0200

It was downloaded from ftp://xmlsoft.org/libxml2/

==============================================

Except where otherwise noted in the source code (trio files, hash.c and list.c) covered by a similar licence but with different Copyright notices:

Copyright (C) 1998-2002 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ""AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

==============================================

trio.c, trio.h:
Copyright (C) 1998 Bjorn Reese and Daniel Stenberg.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED ""AS IS"" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

==============================================

triop.h:
Copyright (C) 2000 Bjorn Reese and Daniel Stenberg.
Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED `AS IS' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

hash.c:
Copyright (C) 2000 Bjorn Reese and Daniel Veillard.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED `AS IS' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

list.c:
Copyright (C) 2000 Gary Pennington and Daniel Veillard.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED `AS IS' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

triodef.h, trionan.c, trionan.h:
Copyright (C) 2001 Bjorn Reese <breese@users.sourceforge.net>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED `AS IS' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

triostr.c, triostr.h:
Copyright (C) 2001 Bjorn Reese and Daniel Stenberg.
Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

1.539 word-wrap 1.2.3

1.539.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.540 jersey-media-jaxb 2.25.1

1.540.1 Available under license:
CDDL+GPL License
http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.541 szmarczak-http-timer 1.1.2
1.541.1 Available under license:
MIT License

Copyright (c) 2018 Szymon Marczak

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.542 websocket-api 9.4.17.v20190418
1.542.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute
the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL
If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

==============================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html
* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----
Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----
OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----
Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.543 slf4j 1.7.26

1.543.1 Available under license :

<!DOCTYPE html PUBLIC "-/W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>SLF4J License</title>
<link rel="stylesheet" type="text/css" media="screen" href="css/site.css" />
</head>
<body>
<script type="text/javascript">prefix='';</script>
Licensing terms for SLF4J

SLF4J source code and binaries are distributed under the MIT license.

Copyright (c) 2004-2017 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

These terms are identical to those of the MIT License, also called the X License or the X11 License, which is a simple, permissive non-copyleft free software license. It is deemed compatible with virtually all types of licenses, commercial or otherwise. In particular, the Free Software Foundation has declared it compatible with other licenses.
GNU GPL. It is also known to be approved by the Apache Software Foundation as compatible with Apache Software License.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "["]
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
1.544.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.545 freemarker 2.3.28

1.545.1 Available under license:

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=========================================================================  
The source code contains the following binaries, which were created at the Apache FreeMarker project, and hence are covered by the same license as the other source files of it:

src/main/misc/overloadedNumberRules/prices.ods
src/manual/en_US/docgen-originals/figures/overview.odg
src/manual/en_US/docgen-originals/figures/model2sketch_with_alpha.png
src/manual/en_US/docgen-originals/figures/tree_with_alpha.png
src/manual/en_US/favicon.png
src/manual/en_US/figures/model2sketch.png
src/manual/en_US/figures/overview.png
src/manual/en_US/figures/model2sketch.png
src/manual/en_US/logo.png
src/manual/zh_CN/favicon.png
src/manual/zh_CN/figures/model2sketch.png
src/manual/zh_CN/figures/overview.png
src/manual/zh_CN/figures/model2sketch.png
src/manual/zh_CN/figures/model2sketch.png
src/manual/zh_CN/logo.png

Apache FreeMarker
Copyright 2015-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
 incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

========================================================================= The Apache FreeMarker documentation contains the following binaries,
which were created at the Apache FreeMarker project, and hence are covered by the same license as the other source files of it:

documentation/_html/docgen-resources/img/patterned-bg.png
documentation/_html/docgen-resources/img/xxe.png
documentation/_html/favicon.png
documentation/_html/figures/model2sketch.png
documentation/_html/figures/overview.png
documentation/_html/figures/tree.png
documentation/_html/logo.png

======================================================================

The documentation includes a selection of icons from various icon sets (fonts), stored together inside these files, which were generated with https://icomoon.io/app/:

documentation/_html/docgen-resources/fonts/icomoon.eot
documentation/_html/docgen-resources/fonts/icomoon.svg
documentation/_html/docgen-resources/fonts/icomoon.ttf
documentation/_html/docgen-resources/fonts/icomoon.woff

The name, license, and attribution of each icon sets (fonts) used follows:

- The documentation includes icons from Entypo pictograms, version 2.0, by Daniel Bruce (http://www.entypo.com/, http://www.danielbruce.se/), licensed under Creative Commons Attribution-ShareAlike 3.0 (CC BY-SA 3.0) (http://creativecommons.org/licenses/by-sa/3.0/legalcode) and under SIL Open Font License 1.1 (http://scripts.sil.org/OFL).

- The documentation includes icons from Font Awesome by Dave Gandy (http://fontawesome.io), licensed under SIL Open Font License 1.1 (http://scripts.sil.org/OFL).

- The documentation includes icons from Material Design icons by Google (http://google.github.io/material-design-icons/), licensed under Creative Common Attribution 4.0 International License (CC-BY 4.0) (https://creativecommons.org/licenses/by/4.0/).

======================================================================

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright holder as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.546 engine.io-client 1.8.5

1.546.1 Available under license :
(The MIT License)

Copyright (c) 2014-2015 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.547 cli-cursor 1.0.2

1.547.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.548 ms 0.7.1 0.7.1
1.548.1 Available under license :
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <rauchg@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.549 nanoid 3.1.12
1.549.1 Available under license :
The MIT License (MIT)

Copyright 2017 Andrey Sitnik <andrey@sitnik.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.550 httpcomponents-client 4.5.13

1.550.1 Available under license:
Apache HttpComponents Client
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work.
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

=========================================================================

This project includes Public Suffix List copied from
<https://publicsuffix.org/list/effective_tld_names.dat>
licensed under the terms of the Mozilla Public License, v. 2.0

Full license text: <http://mozilla.org/MPL/2.0/>

Mozilla Public License Version 2.0
=========================================================================

1. Definitions
--------------

1.1. "Contributor"
means each individual or legal entity that creates, contributes to
the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached
the notice in Exhibit A, the Executable Form of such Source Code
Form, and Modifications of such Source Code Form, in each case
including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described
in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of
version 1.1 or earlier of the License, but not also under the
terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.
1.7. "Larger Work"
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.
2. License Grants and Conditions

--------------------------------

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).
2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this
License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute
or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is free of defects, merchantable, fit for a         *
*  particular purpose or non-infringing. The entire risk as to the      *
*  quality and performance of the Covered Software is with You.         *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,    *
*  repair, or correction. This disclaimer of warranty constitutes an    *
*  essential part of this License. No use of any Covered Software is    *
*  authorized under this License except under this disclaimer.         *
7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License
10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as
defined by the Mozilla Public License, v. 2.0.

1.551 has-ansi 2.0.0 2.0.0

1.551.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.552 range-parser 1.0.3 1.0.3

1.552.1 Available under license:

(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND.
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.553 punycode 2.1.1 2.1.1

1.553.1 Available under license:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.554 got 9.6.0

1.554.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.555 prefixmap 2.0

1.555.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Yet Another UserAgent Analyzer
 * Copyright (C) 2013-2020 Niels Basjes
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/prefixmap/src/test/java/nl/basjes/collections/prefixmap/serialization/AbstractSerializeWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/AbstractSerializationTest.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeStringPrefixMapWithJava.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/AbstractSerializationWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeASCIIPrefixMapWithKryoRegistered.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeStringPrefixMapWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeASCIIPrefixMapWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/AbstractSerializationWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/AbstractSerializationTest.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeStringPrefixMapWithJava.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/AbstractSerializationWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeASCIIPrefixMapWithKryoRegistered.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeStringPrefixMapWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/AbstractSerializationWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeASCIIPrefixMapWithKryoRegistered.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeStringPrefixMapWithKryo.java
prefixmap-e34e4cb/prefixmap/src/test/java/nl/basjes/collections/prefixmap/serialization/AbstractSerializationTest.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeStringPrefixMapKryoRegistered.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeASCIIPrefixMapWithJava.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/SerializeASCIIPrefixMapWithKryo.java
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/java/nl/example/serialization/AbstractSerializeWithJava.java

No license file was found, but licenses were detected in source scan.

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# distributed under the License is distributed on an AS IS BASIS,

Found in path(s):
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/docker/centos7/env.sh
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/docker/centos7/prompt.sh
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/.travis.yml

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018-2020 Niels Basjes
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/tests/java-default/src/test/java/nl/example/TestStringPrefixMapKryoRegistered.java
No license file was found, but licenses were detected in source scan.

This is intended as an overview of the major changes

v2.0
===
- The PrefixMap now implements the Map interface which increases the number of use cases.
- Support for the default Java Serialization and Kryo 4 and Kryo 5

v1.2
===
- Converted this into a Java 9+ module that can still be used as a Java 8 jar dependency.

v1.1
===
- Support for serialization using Kryo

v1.0
---
- Initial release

License

Copyright (C) 2018-2020 Niels Basjes

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-prefixmap-e34e4cb/CHANGELOG.md
No license file was found, but licenses were detected in source scan.

#
# Copyright (C) 2013-2018 Niels Basjes
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an AS IS BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
#
root = true

[*] charset = utf-8
eol = lf
insert_final_newline = true
trim_trailing_whitespace = true
indent_style = space
[*].java
indent_size = 4

[*].yaml
indent_size = 2

[*].xml
indent_size = 2

# Tab indentation (no size specified)
[Makefile]
indent_style = tab

Found in path(s):
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500eba-tar-gz/nielsbasjes-prefixmap-e34e4cb/.editorconfig

No license file was found, but licenses were detected in source scan.

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500eba-tar-gz/nielsbasjes-prefixmap-e34e4cb/serialization/test-base/src/test/resources/log4j.properties
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500eba-tar-gz/nielsbasjes-prefixmap-e34e4cb/prefixmap/src/test/resources/log4j.properties
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500eba-tar-gz/nielsbasjes-prefixmap-e34e4cb/docker/centos7/build_env_checks.sh
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500eba-tar-gz/nielsbasjes-prefixmap-e34e4cb/start-docker.sh
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500eba-tar-gz/nielsbasjes-prefixmap-e34e4cb/docker/centos7/Dockerfile
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500eba-tar-gz/nielsbasjes-prefixmap-e34e4cb/docker/centos7/configure-for-user.sh

No license file was found, but licenses were detected in source scan.

PrefixMap: A Map type datastructure that uses prefixes instead of keys

==============================================================================

[![Travis Build status](https://api.travis-ci.org/nielsbasjes/prefixmap.png?branch=master)](https://travis-ci.org/nielsbasjes/prefixmap)


[![License](https://img.shields.io/license-apache-blue.svg)](https://www.apache.org/licenses/LICENSE-2.0.html)

[![Maven Central](https://img.shields.io/maven-
This is a java library that implements a datastructure I consider to be missing in the java landscape.

The datastructure and its usages
================================

In several situations I have a set of values that are each associated with a different prefix of a String.

The example where I found the biggest need for this was when analyzing the device tags of mobile devices.

- A "GT-I8190N" is a "Samsung" device because it starts with "GT-".
- A "RM-1092" is a "Nokia" device because it starts with "RM-".

So I needed a lookup structure that maps these prefixes to the desired brands.

"GT-" : "Samsung"
"RM-" : "Nokia"

With this code you can now do this type of lookup in a very fast way:

```java
// Parameter caseSensitive=false --> so lookups are caseInsensitive
PrefixMap<String> brandLookup = new StringPrefixMap<>(false);

brandLookup.put("GT-", "Samsung");
brandLookup.put("RM-", "Nokia");

String brandGT = brandLookup.getLongestMatch("GT-I8190N"); // --> "Samsung"
String brandRM = brandLookup.getLongestMatch("RM-1092"); // --> "Nokia"
```

All you need is to add this dependency to your project

```xml
<dependency>
  <groupId>nl.basjes.collections</groupId>
  <artifactId>prefixmap</artifactId>
  <version>2.0</version>
</dependency>
```

Version 2.0 If you are using Kryo 5.x you can use this to register all classes:

```java
StringPrefixMap.configureKryo(kryo);
```

Blog post
==========

A bit more background about this datastructure and how it works can be found in this blog which I wrote about it:
[https://techlab.bol.com/finding-the-longest-matching-string-prefix-fast/]
longest-matching-string-prefix-fast/ ([Local copy](Article.md))

Donations

====
If this project has business value for you then don't hesitate to support me with a small donation.


License

========

Copyright (C) 2018-2020 Niels Basjes

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-
  prefixmap-e34e4cb/README.md
No license file was found, but licenses were detected in source scan.

2020 Niels Basjes
~
~ Licensed under the Apache License, Version 2.0 (the "License");
~ you may not use this file except in compliance with the License.
~ You may obtain a copy of the License at
~
~ https://www.apache.org/licenses/LICENSE-2.0

~
~ Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1174497368_1623906381.73/0/nielsbasjes-prefixmap-v2-0-0-g500ebea-tar-gz/nielsbasjes-
1.556 esprima 2.7.3

1.556.1 Available under license:

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
1.557 commons-net 1.4.1

1.557.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2001-2005 The Apache Software Foundation
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/smtp/SMTPCommand.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ftp/parser/VMSFTPEntryParser.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ftp/FTPFileEntryParser.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/MalformedServerReplyException.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/telnet/TelnetOption.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/DefaultDatagramSocketFactory.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ftp/parser/UnixFTPEntryParser.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/tftp/TFTP.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ProtocolCommandEvent.java
*/
jar/org/apache/commons/net/io/FromNetASCIIInputStream.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/io/ToNetASCIIOutputStream.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/nntp/NewsgroupInfo.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/SocketFactory.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/ftp/parser/NTFTPEntryParser.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/ftp/FTPConnectionClosedException.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/ftp/FTPReply.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/io/ToNetASCIIInputStream.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/smtp/RelayPath.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/ftp/parser/EnterpriseUnixFTPEntryParser.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/util/ListenerList.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/nntp/NNTPCommand.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/ntp/TimeInfo.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/UDPClient.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
jar/org/apache/commons/net/nntp/ArticlePointer.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2003-2004 The Apache Software Foundation
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *  http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
Open Source Used In DNAC Platform Antelope 1929

Found in path(s):
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/SimpleOptionHandler.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/TelnetNotificationHandler.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/SuppressGAOptionHandler.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/InvalidTelnetOptionException.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/TelnetOptionHandler.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/TelnetInputStream.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/TerminalTypeOptionHandler.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/EchoOptionHandler.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/TelnetClient.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-jar/org/apache/commons/net/telnet/Telnet.java

No license file was found, but licenses were detected in source scan.

/* ====================================================================
 * The Apache Software License, Version 1.1
 * Copyright (c) 2001-2005 The Apache Software Foundation.  All rights
 * reserved.
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 * 3. The end-user documentation included with the redistribution,
 * if any, must include the following acknowledgment:
 * "This product includes software developed by the
 * Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgment may appear in the software itself,
 * if and wherever such third-party acknowledgments normally appear. */
* 4. The names "Apache" and "Apache Software Foundation" and
   * "Apache Commons" must not be used to endorse or promote products
   * derived from this software without prior written permission. For
   * written permission, please contact apache@apache.org.
   *
   * 5. Products derived from this software may not be called "Apache",
   * nor may "Apache" appear in their name, without
   * prior written permission of the Apache Software Foundation.
   *
   * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
   * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
   * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
   * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
   * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
   * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
   * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
   * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
   * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
   * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
   * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
   * SUCH DAMAGE.
   *
   * ==========================================================================

   * This software consists of voluntary contributions made by many
   * individuals on behalf of the Apache Software Foundation. For more
   * information on the Apache Software Foundation, please see
   */

Found in path(s):
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/nntp/Article.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2005 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1.jar/org/apache/commons/net/ftp/parser/FTPTimestampParserImpl.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1.jar/org/apache/commons/net/ftp/parser/MVSFTPEntryParser.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1.jar/org/apache/commons/net/ftp/FTPClientConfig.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1.jar/org/apache/commons/net/ftp/Configurable.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1.jar/org/apache/commons/net/ftp/parser/ConfigurableFTPFileEntryParserImpl.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1.jar/org/apache/commons/net/ftp/parser/FTPTimestampParser.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2001-2005 The Apache Software Foundation
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 *     http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/**
 * List all new newsgroups added to the NNTP server since a particular
 * date subject to the conditions of the specified query. If no new
 * newsgroups were added, a zero length array will be returned. If the
 * command fails, null will be returned.
 * <p>
 * @param query  The query restricting how to search for new newsgroups.
 * @return An array of NewsgroupInfo instances containing the information
 * for each new newsgroup added to the NNTP server. If no newsgroups
 * were added, a zero length array will be returned. If the command
 * fails, null will be returned.
 * @exception NNTPConnectionClosedException
 *   If the NNTP server prematurely closes the connection as a result
 *   of the client being idle or some other reason causing the server
 *   to send NNTP reply code 400. This exception may be caught either
 */
* as an IOException or independently as itself.
* @exception IOException If an I/O error occurs while either sending a
*    command to the server or receiving a reply from the server.
* ***/

Found in path(s):
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/nntp/NNTPClient.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2004 The Apache Software Foundation
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ftp/FTPLListParseEngine.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ftp/parser/ParserInitializationException.java
* /opt/cola/permits/1136474281_1613772384.01/0/commons-net-1-4-1-sources-1-
  jar/org/apache/commons/net/ftp/parser/CompositeFileEntryParser.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2004-2005 The Apache Software Foundation
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and

1.558 readable-stream 1.1.14

1.558.1 Available under license:

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.559 statuses 1.5.0 1.5.0

1.559.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.560 jersey-container-servlet 2.25.1
1.560.1 Available under license :
CDDL+GPL License
http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.561 spawn-wrap 2.0.0
1.561.1 Available under license :
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.562 send 0.16.2 0.16.2
1.562.1 Available under license:
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.563 sqlline 1.9.0
1.563.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
  // Licensed to Julian Hyde under one or more contributor license
  // agreements. See the NOTICE file distributed with this work for
  // additional information regarding copyright ownership.
  //
  // Julian Hyde licenses this file to you under the Modified BSD License
  // (the "License"). You may not use this file except in compliance with
  // the License. You may obtain a copy of the License at:
  //
  // http://opensource.org/licenses/BSD-3-Clause
  */

Found in path(s):
* /opt/cola/permits/1179408911_1626107141.97/0/sqlline-1-9-0-sources-jar/sqlline/Application.java
* /opt/cola/permits/1179408911_1626107141.97/0/sqlline-1-9-0-sources-jar/sqlline/ReflectiveCommandHandler.java
* /opt/cola/permits/1179408911_1626107141.97/0/sqlline-1-9-0-sources-jar/sqlline/Dialect.java
No license file was found, but licenses were detected in source scan.
SQLLine Manual
Introduction
License and Terms of Use
Obtaining SQLLine
Installation
Required software
Installing SQLLine
Using SQLLine
Running SQLLine
Connecting to a database
Issuing SQL commands
Command Completion
Command Reference

all
all — Execute SQL against all active connections
appconfig
appconfig - Sets application configuration class name
autocommit
autocommit — Enable or disable automatic transaction commit mode
batch
batch — Start or execute a batch of SQL statements
brief
brief — Enable terse output mode
close
close — Close the active connection
columns
columns — Display columns of a table
commandhandler
commandhandler — Adds command handlers with commands
commit
commit — Commit the current transaction
connect
connect — Connect to a database
dbinfo
dbinfo — Provide information about the current database
describe
describe — Describe a table
dropall
dropall — Drop all tables in the database or in the schema
exportedkeys
exportedkeys — List exported foreign keys for a database
go
go — Change to a different active connection
help
help — Display help information

history
history — Display the command history
-d Print timestamps for each event (default for !history)
-f Print full time date stamps in the US format
-E Print full time date stamps in the European format
-i Print full time date stamps in ISO8601 format
-n Suppresses command numbers
-r Reverses the order of the commands

importedkeys
importedkeys — List imported foreign keys for a database

indexes
indexes — Display indexes for a table

isolation
isolation — Set the transaction isolation mode for the active connection

list
list — Display all active connections

metadata
metadata — Invoke arbitrary metadata commands

nickname
nickname - Create a friendly name for the connection (updates command prompt)

outputformat
outputformat — Change the method for displaying SQL results

primarykeys
primarykeys — Display the primary key columns for a table

procedures
procedures — List stored procedures

prompthandler
prompthandler - Set custom prompt handler class name

properties
properties — Connect to the database defined in the specified properties file.

quit
quit — Exit SQLLine

readonly
readonly - Enable or disable readonly connection

reconnect
reconnect — Reconnect to the current database

record
record — Begin recording all output from SQL commands

rehash
rehash — Obtain a list of all tables and columns from the database

rollback
rollback — Roll back the current transaction

run
run — Execute a command script

save
save — Save the current preferences
scan
scan — Scan class path for JDBC drivers
schemas
schemas — List all the schemas in the database
script
script — Save executed commands to a file
set
set — Set a preference
sql
sql — Execute a SQL against a database
tables
tables — List all the tables in the database
verbose
verbose — Enable verbose output
Parameter Reference
autocommit
autosave
color
colorScheme
confirm
confirmPattern
csvDelimiter
csvQuoteCharacter
dateformat
escapeOutput
fastconnect
force
headerinterval
historyfile
incremental
isolation
maxcolumnwidth
maxHistoryFileRows
maxHistoryRows
maxwidth
mode
nullValue
numberformat
outputformat
prompt
rightprompt
rowlimit
showheader
shownestederrs
showtime
showwarnings
strictJdbc
silent
### Introduction

- License and Terms of Use
- Obtaining SQLLine
- Installation
- Required software
- Installing SQLLine
- Using SQLLine
- Running SQLLine
- Connecting to a database
- Issuing SQL commands
- Command Completion
- Command Reference

### all

- all — Execute SQL against all active connections

### appconfig

- appconfig - Sets application configuration class name

### autocommit

- autocommit — Enable or disable automatic transaction commit mode

### batch

- batch — Start or execute a batch of SQL statements

### brief

- brief — Enable terse output mode

### close

- close — Close the active connection

### columns

- columns — Display columns of a table

### commandhandler

- commandhandler — Adds command handlers with commands

### commit

- commit — Commit the current transaction

### connect

- connect — Connect to a database

### dbinfo

- dbinfo — Provide information about the current database

### describe

- describe — Describe a table

### dropall
dropall — Drop all tables in the database or in the schema
exportedkeys
exportedkeys — List exported foreign keys for a database
go
go — Change to a different active connection
help
help — Display help information
history
history — Display the command history
  -d Print timestamps for each event (default for !history)
  -f Print full time date stamps in the US format
  -E Print full time date stamps in the European format
  -i Print full time date stamps in ISO8601 format
  -n Suppresses command numbers
  -r Reverses the order of the commands
importedkeys
importedkeys — List imported foreign keys for a database
indexes
indexes — Display indexes for a table
isolation
isolation — Set the transaction isolation mode for the active connection
list
list — Display all active connections
metadata
metadata — Invoke arbitrary metadata commands
outputformat
outputformat — Change the method for displaying SQL results
primarykeys
primarykeys — Display the primary key columns for a table
procedures
procedures — List stored procedures
prompthandler
prompthandler - Set custom prompt handler class name
properties
properties — Connect to the database defined in the specified properties file.
quit
quit — Exit SQLLine
reconnect
reconnect — Reconnect to the current database
record
record — Begin recording all output from SQL commands
rehash
rehash — Obtain a list of all tables and columns from the database
rollback
rollback — Roll back the current transaction
run
run — Execute a command script
save
save — Save the current preferences
schemas
schemas — List all the schemas in the database
scan
scan — Scan class path for JDBC drivers
script
script — Save executed commands to a file
set
set — Set a preference
sql
sql — Execute a SQL against a database
tables
tables — List all the tables in the database
verbose
verbose — Enable verbose output
Parameter Reference
autocommit
autosave
color
colorScheme
confirm
confirmPattern
csvDelimiter
csvQuoteCharacter
dateformat
escapeOutput
fastconnect
force
headerinterval
historyfile
incremental
isolation
maxcolumnwidth
maxHistoryFileRows
maxHistoryRows
maxwidth
mode
nullValue
numberformat
outputformat
prompt
rightprompt
rowlimit
showheader
shownestederrs
showtime
showwarnings
silent
strictJdbc
timeformat
timestampformat
trimscripts
uselinecontinuation
verbose

JDBC Driver Support

Introduction

SQLLine is a pure-Java console based utility for connecting to relational databases and executing SQL commands. It is similar to other command-line database access utilities like sqlplus for Oracle, mysql for MySQL, and isql for Sybase/SQL Server. Since it is pure-Java, it is platform independent, and will run on any platform that can run Java 1.3 or higher.

License and Terms of Use

SQLLine is distributed under the BSD License, meaning that you are free to redistribute, modify, or sell the software with almost no restrictions. For the text of the license, see the license text.

For information on obtaining the software under another license, contact the copyright holder: mwp1@cornell.edu.

Obtaining SQLLine

SQLLine is hosted on SourceForge, and is located at https://sqlline.sf.net. The latest release can be downloaded from https://sqlline.sf.net/download.html.

Installation

Required software
Installing SQLLine
Required software

SQLLine depends on the following software:

Java™ Virtual Machine - SQLLine is a pure java program, and requires Java version 1.3 or higher in order to run. The latest JVM can be downloaded from https://java.sun.com.

JLine - SQLLine uses the JLine console reader for command line editing, tab-completion, and command history. It can be downloaded from https://jline.sf.net. Version 0.8.1 or higher is required.

JDBC™ Driver(s) - Since SQLLine uses the Java Database Connectivity package to connect to your database, you need to obtain the correct JDBC driver libraries for your database. Any JDBC compliant driver can be used for this purpose. Drivers can be located by referring to the section on known JDBC drivers, or else by referring to your database vendor's web site.

Installing SQLLine

There are currently no set standards for installing Java software on different platforms. The simplest method for
installing SQLLine in a platform independant way is as follows:

Create a new directory/folder where you like. This will be referred to as sqllinedir.

Download sqlline.jar into sqllinedir.

Download the latest jline.jar from https://jline.sf.net into sqllinedir.

Download your database's JDBC driver files into sqllinedir. Note that some JDBC drivers require some installation, such as uncompressing or unzipping.

Using SQLLine

Running SQLLine

Connecting to a database

Issuing SQL commands

Command Completion

Running SQLLine

SQLLine can be run with the following command:

```
java {-Djava.ext.dirs=sqllinedir} {sqllinedir/sqlline.jar} [options...] [properties files...]
```

For example, if your sqllinedir is /usr/share/java/ (this is typical for Debian Linux), then you might run:

```
java -Djava.ext.dirs=/usr/share/java/ /usr/share/java/sqlline.jar
```

Once you have started SQLLine, you will a prompt that reads "sqlline> ". From here, you can enter either SQLLine commands (beginning with a "!" character), or a SQL statement (if you are currently connected to a database). For example, entering "!help" will display a list of all the available commands.

Connecting to a database

SQLLine is only useful once you are connected to a database. In order to do so, you need to use the "!connect" command, with the JDBC URL of the database you are going to connect to, followed by the username and password (followed optionally by the driver class name). The JDBC URL is specific to the JDBC driver that you are using for your database, but will always start with "jdbc:”; and usually has the machine name of the database and the name of the database itself as part of the string. Quotes (both single and double) could be used in case of url, username, password contain spaces. Consult your driver's documentation for details on the URL to use.

For example, to connect to an Oracle database, you might enter:

```
sqlline> !connect jdbc:oracle:thin:@localhost:1521:mydb scott tiger oracle.jdbc.driver.OracleDriver
```

Connected to: Oracle (version Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production)
Driver: Oracle JDBC driver (version 9.0.1.0.0)
Autocommit status: true

0: jdbc:oracle:thin:@localhost:1521:mydb>

Once you are connected, the prompt will change the reflect URL of the current connection.

SQLLine can also accept command line options to connect to a database. If a file is specified on the command line, it will be treated as a properties file, and try to connect to the database specified therein. For details, see the properties command documentation.

SQLLine allows you to have multiple connections to databases (either the same database, or a different database) open at the same time. Issuing the connect command multiple times will add the new connection to the list of open connections. The “current” connection (against which all SQL statements will be executed) can be switched with the go command. The currently open connections can be listed with the list command.

Issuing SQL commands

Once you are connected to a database, you can begin SQL statements against the current connection. For example:

Example 1. Issuing SQL commands

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM MYTABLE;

+-----+
| ID  |
+-----+
| 1   |
| 2   |
| 3   |
| 4   |
+-----+
4 rows selected (0.012 seconds)
0: jdbc:oracle:thin:@localhost:1521:mydb>

Command Completion

On UNIX systems, SQLLine allows command-line completion using the tab key. This will be familiar for users of such popular shells as bash and tcsh. In general, hitting the TAB key when part of the way through typing a command will fill in the rest of the command, or else will display a selection of appropriate possibilities.

Command Reference

all
all — Execute SQL against all active connections
appconfig
appconfig - Sets application configuration class name
autocommit
autocommit — Enable or disable automatic transaction commit mode
batch
batch — Start or execute a batch of SQL statements
brief
brief — Enable terse output mode
close
close — Close the active connection
columns
columns — Display columns of a table
commandhandler
commandhandler — Adds command handlers with commands
commit
commit — Commit the current transaction
connect
connect — Connect to a database
dbinfo
dbinfo — Provide information about the current database
describe
describe — Describe a table
dropall
dropall — Drop all tables in the database or in the schema
exportedkeys
exportedkeys — List exported foreign keys for a database
go
go — Change to a different active connection
help
help — Display help information
history
history — Display the command history
  -d Print timestamps for each event (default for !history)
  -f Print full time date stamps in the US format
  -E Print full time date stamps in the European format
  -i Print full time date stamps in ISO8601 format
  -n Suppresses command numbers
  -r Reverses the order of the commands
importedkeys
importedkeys — List imported foreign keys for a database
indexes
indexes — Display indexes for a table
isolation
isolation — Set the transaction isolation mode for the active connection
list
list — Display all active connections
metadata
metadata — Invoke arbitrary metadata commands
outputformat
outputformat — Change the method for displaying SQL results
primarykeys
primarykeys — Display the primary key columns for a table
procedures
procedures — List stored procedures
prompthandler
prompthandler - Set custom prompt handler class name
properties
properties — Connect to the database defined in the specified properties file.
quit
quit — Exit SQLLine
reconnect
reconnect — Reconnect to the current database
record
record — Begin recording all output from SQL commands
rehash
rehash — Obtain a list of all tables and columns from the database
rollback
rollback — Roll back the current transaction
run
run — Execute a command script
save
save — Save the current preferences
schemas
schemas — List all the schemas in the database
scan
scan — Scan class path for JDBC drivers
script
script — Save executed commands to a file
set
set — Set a preference
sql
sql — Execute a SQL against a database
tables
tables — List all the tables in the database
verbose
verbose — Enable verbose output
all

Name

all — Execute SQL against all active connections

Synopsis

!all {SQL statement}
Execute the specified SQL against all the current connections.

Example of "all" command

1: jdbc:mysql://localhost/mydb> !list

2 active connections:
#0 open jdbc:oracle:thin:@localhost:1521:mydb
#1 open jdbc:mysql://localhost/mydb

1: jdbc:mysql://localhost/mydb> !all DELETE FROM COMPANY;

Executing SQL against: jdbc:oracle:thin:@localhost:1521:mydb
4 rows affected (0.004 seconds)

Executing SQL against: jdbc:mysql://localhost/mydb
1 row affected (3.187 seconds)

1: jdbc:mysql://localhost/mydb> 

autocommit

Name

autocommit — Enable or disable automatic transaction commit mode

Synopsis

!autocommit [on/off]

Description

Set autocommit mode on or off. When autocommit is on, every individual SQL statement will be committed after it is issued. Otherwise, the commit command will need to be issued in order to commit any changes to the database.

Example of "autocommit" command

1: jdbc:mysql://localhost/mydb> !autocommit off

Autocommit status: false

1: jdbc:mysql://localhost/mydb>

batch

Name

batch — Start or execute a batch of SQL statements
Synopsis

!batch

Description

Start or execute a batch of SQL statements. The first time the statement is issued, subsequent SQL statements will be deferred until the batch command is issued again. Databases can frequently optimize multiple batched statements (e.g., for bulk data loading) in order to speed up execution.

Example of "batch" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !batch

Batching SQL statements. Run "batch" again to execute the batch.

0: jdbc:oracle:thin:@localhost:1521:mydb> INSERT INTO MYTABLE (ID) VALUES (1);
0: jdbc:oracle:thin:@localhost:1521:mydb> INSERT INTO MYTABLE (ID) VALUES (2);
0: jdbc:oracle:thin:@localhost:1521:mydb> INSERT INTO MYTABLE (ID) VALUES (3);
0: jdbc:oracle:thin:@localhost:1521:mydb> INSERT INTO MYTABLE (ID) VALUES (4);
0: jdbc:oracle:thin:@localhost:1521:mydb> !batch

Running batched SQL statements...
COUNT STATEMENT
1 INSERT INTO MYTABLE (ID) VALUES (1)
1 INSERT INTO MYTABLE (ID) VALUES (2)
1 INSERT INTO MYTABLE (ID) VALUES (3)
1 INSERT INTO MYTABLE (ID) VALUES (4)

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM MYTABLE;

+-----+
| ID  |
+-----+
| 1   |
| 2   |
| 3   |
| 4   |
+-----+
4 rows selected (0.012 seconds)

0: jdbc:oracle:thin:@localhost:1521:mydb>

brief

Name

brief — Enable terse output mode
Synopsis

!brief

Description

Set verbose mode off. This will prevent printing of stack traces when errors occur.

Example of "brief" command

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM BOGUS_TABLE;

Error: ORA-00942: table or view does not exist (state=42000, code=942)
java.sql.SQLException: ORA-00942: table or view does not exist

   at oracle.jdbc.dbaccess.DBErrort.addHandler(DBErrort.java:180)
   at oracle.jdbc.ttc7.TTLlerr.handleError(TTLlerr.java:208)
   at oracle.jdbc.ttc7.Oall7.receive(Oall7.java:543)
   at oracle.jdbc.ttc7.TTC7Protocol.doOall7(TTC7Protocol.java:1451)
   at oracle.jdbc.ttc7.TTC7Protocol.parseExecuteDescribe(TTC7Protocol.java:651)
   at oracle.jdbc.driver.OracleStatement.doExecuteQuery(OracleStatement.java:2110)
   at oracle.jdbc.driver.OracleStatement.doExecuteWithTimeout(OracleStatement.java:2324)
   at oracle.jdbc.driver.OracleStatement.execute(OracleStatement.java:909)
   at sqlline.SqlLine$Commands.sql(SqlLine.java:3183)
   at sqlline.SqlLine.dispatch(SqlLine.java:771)
   at sqlline.SqlLine.begin(SqlLine.java:634)
   at sqlline.SqlLine.main(SqlLine.java:332)

0: jdbc:oracle:thin:@localhost:1521:mydb> !brief

verbose: off

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM BOGUS_TABLE;

Error: ORA-00942: table or view does not exist (state=42000, code=942)

0: jdbc:oracle:thin:@localhost:1521:mydb>

close

Name

close — Close the active connection

Synopsis

!close
Description

Close the current connection and remove it from the connection list.

Example of "close" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !list

1 active connection:
#0 open jdbc:oracle:thin:@localhost:1521:mydb

0: jdbc:oracle:thin:@localhost:1521:mydb> !close

Closing: oracle.jdbc.driver.OracleConnection

sqlline> !list

No active connections

sqlline>

columns

Name

columns — Display columns of a table

Synopsis

!columns {table name}

Description

List the columns in the specified table.

Example of "columns" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !columns EMPLOYEE

+---------------------------+-------------------------+-----------------+------------------+
<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>COLUMN_NAME</th>
<th>DATA_TYPE</th>
<th>TYPE_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>ID</td>
<td>3</td>
<td>NUMBER</td>
</tr>
<tr>
<td>EMPLOYEE</td>
<td>FIRST_NAME</td>
<td>12</td>
<td>VARCHAR2</td>
</tr>
<tr>
<td>EMPLOYEE</td>
<td>LAST_NAME</td>
<td>12</td>
<td>VARCHAR2</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
</tbody>
</table>
commandhandler

Name

commandhandler — Adds command handlers with commands

Synopsis

!'commandhandler {command handler} {command handler}*

Description

Adds command handlers with commands.
Command handler class must have a constructor accepting Sqlline and implement sqlline.CommandHandler

Example of "commandhandler" command

sqlline> 'hello
Unknown command: hello
sqlline> '!commandhandler sqlline.extensions.HelloWorldCommandHandler
sqlline> 'hello
HELLO WORLD

commit

Name

commit — Commit the current transaction

Synopsis

!'commit

Description

Commit the current transaction, if autocommit is off.

Example of "commit" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !autocommit off

Autocommit status: false

0: jdbc:oracle:thin:@localhost:1521:mydb> INSERT INTO EMPLOYEE (ID, FIRST_NAME, LAST_NAME)
VALUES (1, 'Bob', 'Smith');
1 row affected (1.254 seconds)

0: jdbc:oracle:thin:@localhost:1521:mydb> !commit

Commit complete (0.661 seconds)

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM EMPLOYEE;

+--------------+-------------+------------+
| ID           | FIRST_NAME  | LAST_NAME  |
+--------------+-------------+------------+
| 1            | Bob         | Smith      |
+--------------+-------------+------------+
1 row selected (0.309 seconds)

0: jdbc:oracle:thin:@localhost:1521:mydb> connect

Name

connect — Connect to a database

Synopsis

!connect [-p property value] [url] [username password] [driver class]

Description

Connect to a database. Quotes could be used in case of having spaces

Example of "connect" command

sqline> !connect jdbc:oracle:thin:@localhost:1521:mydb scott tiger oracle.jdbc.driver.OracleDriver

Connected to: Oracle (version Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production)
Driver: Oracle JDBC driver (version 9.0.1.0.0)
Autocommit status: true

0: jdbc:oracle:thin:@localhost:1521:mydb>

dbinfo

Name
dbinfo — Provide information about the current database

Synopsis

!dbinfo

Description

List metadata information about the current connection.

Example of "dbinfo" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !dbinfo

allProceduresAreCallable    false
allTablesAreSelectable      false
dataDefinitionCausesTransactionCommit    true
dataDefinitionIgnoredInTransactions false
doesMaxRowSizeIncludeBlobs   true
catalogSeparator
catalogTerm
databaseProductName        Oracle
databaseProductVersion       Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production
defaultTransactionIsolation 2
driverMajorVersion          9
driverMinorVersion          0
driverName                  Oracle JDBC driver
driverVersion               9.0.1.0.0
definitionCharacters       $#
definitionQuoteString      "
definitionBinaryLiteralLength 1000
definitionCatalogNameLength 0
definitionCharLiteralLength 2000
definitionColumnNameLength 30
definitionColumnsInGroupBy 0
definitionColumnsInIndex 32
definitionColumnsInOrderBy 0
definitionColumnsInSelect 0
definitionColumnsInTable 1000
definitionConnections 0
definitionCursorNameLength 0
definitionIndexLength 0
definitionProcedureNameLength 30
definitionRowSize 2000
definitionSchemaNameLength 30
definitionStatementLength 65535
getMaxStatements: 0
getMaxTableNameLength: 30
getMaxTablesInSelect: 0
getMaxUserNameLength: 30
getNumericFunctions: ABS, CEIL, COS, COSH, EXP, FLOOR, LN, LOG, MOD, POWER, ROUND, SIGN, SIN, SINH, SQRT, TAN, TANH, TRUNC, AVG, COUNT, GLB, LUB, MAX, MIN, STDDEV, SUM, VARIANCE
getProcedureTerm: procedure
getSchemaTerm: schema
getSearchStringEscape: //
getSQLKeywords: ACCESS, ADD, ALTER, AUDIT, CLUSTER, COLUMN, COMMENT, COMPRESS, CONNECT, DATE, DROP, EXCLUSIVE, FILE, IDENTIFIED, IMMEDIATE, INCREMENT, INDEX, INITIAL, INTERSECT, LEVEL, LOCK, LONG, MAXEXTENTS, MINUS, MODE, NOAUDIT, NOCOMPRESS, NOWAIT, NUMBER, OFFLINE, ONLINE, PCTFREE, PRIOR, all_PL_SQL_reserved_words
getStringFunctions: CHR, INITCAP, LOWER, LPAD, LTRIM, NLS_INITCAP, NLS_LOWER, NLS_UPPER, REPLACE, RPAD, RTRIM, SOUNDEX, SUBSTR, SUBSTRB, TRANSLATE, UPPER, ASCII, INSTR, INSTRB, LENGTH, LENGTHB, NLSSORT, CHARTOROWID, CONVERT, HEXTORAW, RAWTOHEX, ROWIDTOCHAR, TO_CHAR, TO_DATE, TO_LABEL, TO_MULTI_BYTE, TO_NUMBER, TO_SINGLE_BYTE
getSystemFunctions: DUMP, GREATEST, GREATEST_LB, LEAST, LEAST_UB, NVL, UID, USER, USERENV, VSIZE
getTimeDateFunctions: ADD_MONTHS, LAST_DAY, MONTHS_BETWEEN, NEW_TIME, NEXT_DAY, ROUND, SYSDATE, TRUNC
getUrl: jdbc:oracle:thin:@localhost:1521:mydb
getUserName: MYDB
isCatalogAtStart: false
isReadOnly: false
<table>
<thead>
<tr>
<th>TABLE_SCHEM</th>
<th>TABLE_NAME</th>
<th>COLUMN_NAME</th>
<th>DATA_TYPE</th>
<th>TYPE_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYDB</td>
<td>EMPLOYEE</td>
<td>ID</td>
<td>3</td>
<td>NUMBER</td>
</tr>
<tr>
<td>MYDB</td>
<td>EMPLOYEE</td>
<td>FIRST_NAME</td>
<td>12</td>
<td>VARCHAR2</td>
</tr>
<tr>
<td>MYDB</td>
<td>EMPLOYEE</td>
<td>LAST_NAME</td>
<td>12</td>
<td>VARCHAR2</td>
</tr>
</tbody>
</table>
dropall

Name

dropall — Drop all tables in the database or in the schema

Synopsis

!dropall [schema_pattern]

Description

Drop every table in the database.

Example of "dropall" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !dropall

Really drop every table in the database? (y/n) y

1/5 DROP TABLE COMPANY;
2/5 DROP TABLE EMPLOYEE;
3/5 DROP TABLE PROJECTS;
4/5 DROP TABLE EMPLOYEE_PROJECTS;
5/5 DROP TABLE COMPANYEMPLOYEES;

0: jdbc:oracle:thin:@localhost:1521:mydb>

Warning on using "dropall"

Warning

This command will destroy all data in the database (subject to access restrictions). Use with extreme caution.

exportedkeys

Name

exportedkeys — List exported foreign keys for a database

Synopsis

!exportedkeys [table name]
List all the foreign keys that are exported by the specified table.

Example of "exportedkeys" command

```
0: jdbc:hsqldb:db-hypersonic> !exportedkeys COMPANY

+----------------+---------------+----------------+-------------+
| PKTABLE_SCHEM  | PKTABLE_NAME  | PKCOLUMN_NAME  | FKTABLE_CAT |
+----------------+---------------+----------------+-------------+
|                | COMPANY       | COMPANY_ID     |             |
+----------------+---------------+----------------+-------------+
0: jdbc:hsqldb:db-hypersonic>
```

go

Name

go — Change to a different active connection

Synopsis

```
!go {connection number}
```

Description

Switch to the specified connection.

Example of "go" command

```
0: jdbc:oracle:thin:@localhost:1521:mydb> !list

2 active connections:
#0 open jdbc:oracle:thin:@localhost:1521:mydb
#1 open jdbc:mysql://localhost/mydb

0: jdbc:oracle:thin:@localhost:1521:mydb> !go 1

1: jdbc:mysql://localhost/mydb> !go 0

0: jdbc:oracle:thin:@localhost:1521:mydb>
```

help

Name
help — Display help information

Synopsis

!help

Description

Display information about allowed commands.

Example of "help" command

sqline> !help
!all Execute the specified SQL against all the current connections
!appconfig Set custom application configuration class name
!autocommit Set autocommit mode on or off
!batch Start or execute a batch of statements
!brief Set verbose mode off
!call Execute a callable statement
!close Close the current connection to the database
!closeall Close all current open connections
!columns List all the columns for the specified table
!commandhandler Add a command handler
!commit Commit the current transaction (if autocommit is off)
!connect Open a new connection to the database
!dbinfo Give metadata information about the database
!describe Describe a table
!dropall Drop all tables in the current database
!exportedkeys List all the exported keys for the specified table
!go Select the current connection
!help Print a summary of command usage
!history Display the command history
-d Print timestamps for each event (default for !history)
-f Print full time date stamps in the US format
-E Print full time date stamps in the European format
-i Print full time date stamps in ISO8601 format
-n Suppresses command numbers
-r Reverses the order of the commands
!importedkeys List all the imported keys for the specified table
!indexes List all the indexes for the specified table
!isolation Set the transaction isolation for this connection
!list List the current connections
!manual Display the SQLLine manual
!metadata Obtain metadata information
!nativesql Show the native SQL for the specified statement
!nickname Create a friendly name for the connection (updates command
prompt)

!outputformat Set the output format for displaying results (table, vertical, csv, tsv, xmlattrs, xmlelements, json, ansiconsole)

!primarykeys List all the primary keys for the specified table

!procedures List all the procedures

!prompthandler Set custom prompt handler class name

!properties Connect to the database specified in the properties file(s)

!quit Exits the program

!reconnect Reconnect to the database

!record Record all output to the specified file

!rehash Fetch table and column names for command completion

!rollback Roll back the current transaction (if autocommit is off)

!run Run a script from the specified file

!save Save the current variables and aliases

!schemas List all the schemas in the database

!scan Scan for installed JDBC drivers

!script Start saving a script to a file

!set Set a sqlline variable

!sql Execute a SQL command

!tables List all the tables in the database

!typeinfo Display the type map for the current connection

!verbose Set verbose mode on

Variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoCommit</td>
<td>true/false</td>
<td>Enable/disable automatic transaction commit</td>
</tr>
<tr>
<td>autoSave</td>
<td>true/false</td>
<td>Automatically save preferences</td>
</tr>
<tr>
<td>color</td>
<td>true/false</td>
<td>Control whether color is used for display</td>
</tr>
<tr>
<td>colorScheme</td>
<td>chester/dark/dracula/light/obsidian/solarized/vs2010</td>
<td>Syntax highlight schema</td>
</tr>
<tr>
<td>confirm</td>
<td>true/false</td>
<td>Prompts for confirmation before running commands specified in confirmPattern</td>
</tr>
<tr>
<td>confirmPattern</td>
<td>pattern</td>
<td>Pattern commands to issue prompt before execution</td>
</tr>
<tr>
<td>csvDelimiter</td>
<td>String</td>
<td>Delimiter in csv outputFormat</td>
</tr>
<tr>
<td>csvQuoteCharacter</td>
<td>char</td>
<td>Quote character in csv outputFormat</td>
</tr>
<tr>
<td>dateFormat</td>
<td>pattern</td>
<td>Format dates using SimpleDateFormat pattern</td>
</tr>
<tr>
<td>escapeOutput</td>
<td>true/false</td>
<td>Escape control symbols in output</td>
</tr>
<tr>
<td>fastConnect</td>
<td>true/false</td>
<td>Skip building table/column list for tab-completion</td>
</tr>
<tr>
<td>force</td>
<td>true/false</td>
<td>Continue running script even after errors</td>
</tr>
<tr>
<td>headerInterval</td>
<td>integer</td>
<td>The interval between which headers are displayed</td>
</tr>
<tr>
<td>historyFile</td>
<td>path</td>
<td>File in which to save command history. Default is</td>
</tr>
<tr>
<td>incremental</td>
<td>true/false</td>
<td>Do not receive all rows from server before printing the first row; uses fewer resources,</td>
</tr>
</tbody>
</table>


Open Source Used In DNAC Platform Antelope 1961
especially for long-running queries, but column
widths may be incorrect
incrementalBufferRows integer Enter incremental mode if a query returns more
than "incrementalBufferRows".

isolation LEVEL Set transaction isolation level

maxColumnWidth integer The maximum width to use when displaying columns

maxHeight integer The maximum height of the terminal

maxWidth integer The maximum width of the terminal

maxHistoryFileRows integer The maximum number of history rows
to store in history file

maxHistoryRows integer The maximum number of history rows
to store in memory

mode emacs/vi The editing mode

nullValue String Use String in place of NULL values

numberFormat pattern Format numbers using DecimalFormat pattern

outputFormat table/vertical/csv/tsv/xmlattrs/xmlelements/json/ansiconsole
    Format mode for result display

prompt pattern Format prompt

propertiesFile path File from which SQLLine reads properties on
    startup; default is
    $HOME/.sqlline/sqlline.properties (UNIX, Linux,
    macOS), $HOME/sqlline/sqlline.properties (Windows)

rightPrompt pattern Format right prompt

rowLimit integer Maximum number of rows returned from a query; zero
    means no limit

showCompletionDesc true/false Display help for completions

showElapsedTime true/false Display execution time when verbose

showHeader true/false Show column names in query results

showNestedErrs true/false Display nested errors

showWarnings true/false Display connection warnings

silent true/false Be more silent

strictJdbc true/false Use strict JDBC

timeFormat pattern Format times using SimpleDateFormat pattern

timeout integer Query timeout in seconds; less than zero means no

timeout

timestampFormat pattern Format timestamps using SimpleDateFormat pattern

trimScripts true/false Remove trailing spaces from lines read from script
    files

useLineContinuation true/false Use line continuation

verbose true/false Show verbose error messages and debug info

Key-strokes:
alt-b Backward word
alt-c Capitalize word
alt-d Kill word
alt-f Forward word
alt-h Next color scheme
alt-l Lowercase word
alt-n   History search forward
alt-p   History search backward
alt-t   Transpose words
alt-u   Uppercase word

ctrl-a   To the beginning of line
ctrl-b   Backward char
ctrl-d   Delete char
ctrl-e   To the end of line
ctrl-f   Forward char
ctrl-h   Backward delete char
ctrl-i   Complete
ctrl-j   Enter
ctrl-k   Kill the line
ctrl-m   Enter
ctrl-n   Down line or history
ctrl-p   Up line or history
ctrl-r   History incremental search backward
ctrl-s   History incremental search forward
ctrl-t   Transpose chars
ctrl-u   Kill the whole line
ctrl-w   Backward kill the line

Comments, bug reports, and patches go to ???
sqline>

history

Name

history — Display the command history
-d Print timestamps for each event (default for !history)
-f Print full time date stamps in the US format
-E Print full time date stamps in the European format
-i Print full time date stamps in ISO8601 format
-n Suppresses command numbers
-r Reverses the order of the commands

Synopsis

!history

Description

Display the command history.

Example of "history" command
sqlline> !history

1. 'outputformat xmlelements
2. SELECT * FROM PCPOINT;
3. 'outputformat vertical
4. SELECT * FROM PCPOINT;
5. 'tables

sqlline>

importedkeys

Name

importedkeys — List imported foreign keys for a database

Synopsis

'importedkeys {table name}

Description

List all the foreign keys that are imported by the specified table.

Example of "importedkeys" command

0: jdbc:hsqldb:db-hypersonic> 'importedkeys EMPLOYEE

+----------------+---------------+----------------+-------------+
<table>
<thead>
<tr>
<th>PKTABLE_SCHEM</th>
<th>PKTABLE_NAME</th>
<th>PKCOLUMN_NAME</th>
<th>FKTABLE_CAT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COMPANY</td>
<td>COMPANY_ID</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
+----------------+---------------+----------------+-------------+

0: jdbc:hsqldb:db-hypersonic>

indexes

Name

indexes — Display indexes for a table

Synopsis

'indexes {table name}

Description
List all the indexes on the specified table name.

Example of "indexes" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !indexes EMPLOYEE

+-------------+-------------+-----------------+-------+--------------+
| TABLE_NAME  | NON_UNIQUE  |   INDEX_NAME    | TYPE  | COLUMN_NAME  |
+-------------+-------------+-----------------+-------+--------------+
| EMPLOYEE    | 0           |                 | 0     |              |
| EMPLOYEE    | 0           | SYS_C003115849  | 1     | ID           |
| EMPLOYEE    | 1           | FIRST_NAME      | 1     | FIRST_NAME   |
| EMPLOYEE    | 1           | LAST_NAME       | 1     | LAST_NAME    |
+-------------+-------------+-----------------+-------+--------------+

0: jdbc:oracle:thin:@localhost:1521:mydb>

isolation

Name

isolation — Set the transaction isolation mode for the active connection

Synopsis

!isolation {isolation level}

Description

Set the isolation level for the current transaction. For a description of the different isolation levels, see https://java.sun.com/j2se/1.3/docs/api/java/sql/Connection.html#field_detail.

If the specified isolation level is not supported or DEFAULT is specified then there will be used level returned by java.sql.DatabaseMetaData#getDefaultTransactionIsolation

Example of "isolation" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !isolation TRANSACTION_READ_UNCOMMITTED

Error: READ_COMMITTED and SERIALIZABLE are the only valid transaction levels (state=,code=17030)

0: jdbc:oracle:thin:@localhost:1521:mydb> !isolation TRANSACTION_SERIALIZABLE

Transaction isolation: TRANSACTION_SERIALIZABLE

0: jdbc:oracle:thin:@localhost:1521:mydb>

list
Name

list — Display all active connections

Synopsis

!list

Description

List all the current connections.

Example of "list" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !list

2 active connections:
#0 open     jdbc:oracle:thin:@localhost:1521:mydb
#1 open     jdbc:mysql://localhost/mydb

0: jdbc:oracle:thin:@localhost:1521:mydb>

metadata

Name

metadata — Invoke arbitrary metadata commands

Synopsis

!metadata {methodname} {params...}

Description

Execute an arbitrary metadata method against the current connection. refpurpose are separated by spaces. Use "" for a blank String, and null for a null parameter. For information on available metadata methods, see https://java.sun.com/j2se/1.3/docs/api/java/sql/DatabaseMetaData.html.

Example of "metadata" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !metadata supportsSelectForUpdate

ture

0: jdbc:oracle:thin:@localhost:1521:mydb> !metadata getTypeInfo
<table>
<thead>
<tr>
<th>TYPE_NAME</th>
<th>DATA_TYPE</th>
<th>PRECISION</th>
<th>LITERAL_PREFIX</th>
<th>LITERAL_SUFFIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER</td>
<td>-7</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER</td>
<td>-6</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER</td>
<td>-5</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LONG RAW</td>
<td>-4</td>
<td>2147483647</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>RAW</td>
<td>-3</td>
<td>2000</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>LONG</td>
<td>-1</td>
<td>2147483647</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>CHAR</td>
<td>1</td>
<td>2000</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>NUMBER</td>
<td>2</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER</td>
<td>4</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLOAT</td>
<td>6</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REAL</td>
<td>7</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARCHAR2</td>
<td>12</td>
<td>4000</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>DATE</td>
<td>93</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRUCT</td>
<td>2002</td>
<td>0</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>ARRAY</td>
<td>2003</td>
<td>0</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>BLOB</td>
<td>2004</td>
<td>4294967295</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOB</td>
<td>2005</td>
<td>4294967295</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>REF</td>
<td>2006</td>
<td>0</td>
<td>'</td>
<td>'</td>
</tr>
</tbody>
</table>

0: jdbc:oracle:thin:@localhost:1521:mydb> !meta getProcedureColumns null % %JAVA% %

<table>
<thead>
<tr>
<th>PROCEDURE_CAT</th>
<th>PROCEDURE_SCHEM</th>
<th>PROCEDURE_NAME</th>
<th>COLUMN_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBMS_JAVA</td>
<td>SYS</td>
<td>DROPJAVA</td>
<td>OPTIONS</td>
</tr>
<tr>
<td>XMLPARSER</td>
<td>SYS</td>
<td>GETJAVARELEASEVERSION</td>
<td></td>
</tr>
<tr>
<td>DBMS_JAVA</td>
<td>SYS</td>
<td>LOADJAVA</td>
<td>OPTIONS</td>
</tr>
<tr>
<td>DBMS_JAVA</td>
<td>SYS</td>
<td>LOADJAVA</td>
<td>OPTIONS</td>
</tr>
<tr>
<td>DBMS_JAVA</td>
<td>SYS</td>
<td>RESOLVER</td>
<td></td>
</tr>
</tbody>
</table>

0: jdbc:oracle:thin:@localhost:1521:mydb>

outputformat

Name

outputformat — Change the method for displaying SQL results

Synopsis

!outputformat {format name}

Description
Set the mode for displaying results from statements. This is useful for saving output from SQL statements to either a visually pleasing format or an easily parseable format.

In the case of csv or table output format there could be specified additional parameters like csv delimiter, csv quote character, max column width.

Such settings will change corresponding properties' values. If the parameters not specified then properties are left as is.

Example of table output formatting

```
0: jdbc:oracle:thin:@localhost:1521:mydb> !outputformat table
0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;
+-------------+------------+
| COMPANY_ID  |    NAME    |
+-------------+------------+
| 1           | Apple      |
| 2           | Sun        |
| 3           | IBM        |
| 4           | Microsoft  |
+-------------+------------+
4 rows selected (0.012 seconds)
```

Example of vertical output formatting

```
0: jdbc:oracle:thin:@localhost:1521:mydb> !outputformat vertical
0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;
COMPANY_ID  1
NAME        Apple
COMPANY_ID  2
NAME        Sun
COMPANY_ID  3
NAME        IBM
COMPANY_ID  4
NAME        Microsoft
4 rows selected (0.011 seconds)
```

Example of CSV output formatting

```
0: jdbc:calcite:model=target/test-classes/mod> !set outputFormat csv ### @
0: jdbc:calcite:model=target/test-classes/mod> SELECT * FROM COMPANY;
@COMPANY_ID@###@NAME@
@1@###@Apple@
@2@###@Sun@
@3@###@IBM@
```
Example of TSV output formatting

0: jdbc:oracle:thin:@localhost:1521:mydb> !outputformat tsv
0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;
'COMPANY_ID'  'NAME'
'1'  'Apple'
'2'  'Sun'
'3'  'IBM'
'4'  'Microsoft'
4 rows selected (0.013 seconds)

Example of XML attribute output formatting

0: jdbc:oracle:thin:@localhost:1521:mydb> !outputformat xmlattr
0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;
<resultset>
  <result COMPANY_ID="1" NAME="Apple"/>
  <result COMPANY_ID="2" NAME="Sun"/>
  <result COMPANY_ID="3" NAME="IBM"/>
  <result COMPANY_ID="4" NAME="Microsoft"/>
</resultset>
4 rows selected (0.012 seconds)

Example of XML element output formatting

0: jdbc:oracle:thin:@localhost:1521:mydb> !outputformat xmlelements
0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;
<resultset>
  <result>
    <COMPANY_ID>1</COMPANY_ID>
    <NAME>Apple</NAME>
  </result>
  <result>
    <COMPANY_ID>2</COMPANY_ID>
    <NAME>Sun</NAME>
  </result>
  <result>
    <COMPANY_ID>3</COMPANY_ID>
    <NAME>IBM</NAME>
  </result>
  <result>
    <COMPANY_ID>4</COMPANY_ID>
    <NAME>Microsoft</NAME>
  </result>
</resultset>
4 rows selected (0.02 seconds)
0: jdbc:oracle:thin:@localhost:1521:mydb>
Example of JSON output formatting
0: jdbc:calcite:model=target/test-classes/mod> !outputformat json
0: jdbc:calcite:model=target/test-classes/mod> SELECT * FROM COMPANY;
{"resultset":[
{"COMPANY_ID":1,"NAME":"Apple"},
{"COMPANY_ID":2,"NAME":"Sun"},
{"COMPANY_ID":3,"NAME":"IBM"},
{"COMPANY_ID":4,"NAME":"Microsoft"}
]}
4 rows selected (0.03 seconds)
0: jdbc:calcite:model=target/test-classes/mod>
Example of AnsiConsole output formatting
0: jdbc:calcite:model=target/test-classes/mod> !outputformat ansiconsole
0: jdbc:calcite:model=target/test-classes/mod> SELECT * FROM COMPANY;
COMPANY_ID NAME
1
Apple
2
Sun
3
IBM
4
Microsoft
4 rows selected (0.03 seconds)
0: jdbc:calcite:model=target/test-classes/mod>
primarykeys
Name
primarykeys — Display the primary key columns for a table
Synopsis
!primarykeys {table name}
Description
List all the primary keys for the specified table.
Example of "primarykeys" command
0: jdbc:hsqldb:db-hypersonic> !primarykeys COMPANY
+--------------+-------------+--------------+----------+----------+
| TABLE_SCHEM | TABLE_NAME | COLUMN_NAME | KEY_SEQ |

Open Source Used In DNAC Platform Antelope 1970

PK_N |


procedures

Name

procedures — List stored procedures

Synopsis

!procedures [procedure name pattern]

Description

Display a list of all the stored procedures that are exposed by the current database.

Example of "procedures" command

```
0: jdbc:oracle:thin:@localhost:1521:mydb> !procedures %JDBC%
```

```
+----------------+------------------+------------------+
| PROCEDURE_CAT  | PROCEDURE_SCHEM  |  PROCEDURE_NAME  |
|----------------+------------------+------------------+
| WK_ADM         | WKSYS            | GET_JDBC_DRIVER  |
| WK_ADM         | WKSYS            | SET_JDBC_DRIVER  |
+----------------+------------------+------------------+
```

```
0: jdbc:oracle:thin:@localhost:1521:mydb>
```

properties

Name

properties — Connect to the database defined in the specified properties file.

Synopsis

!properties {file}

Description

Connect to the database defined in the specified properties file. The properties file is expected to have the following values:
url: the database URL to which to connect

driver: the driver class that handles the URL. If unspecified, SQLLine with automatically scan the CLASSPATH for an appropriate driver

user: the username to use to connect to the database. If unspecified, SQLLine will prompt for it.

password: the password to use to connect to the database. If unspecified, SQLLine will prompt for it.

The key names in the properties file can also use JDO semantics for the properties. These properties are "javax.jdo.option.ConnectionURL", "javax.jdo.option.ConnectionDriverName", "javax.jdo.option.ConnectionUserName", and "javax.jdo.option.ConnectionPassword". Furthermore, any properties that end with "ConnectionURL", "ConnectionDriverName", "ConnectionUserName", and "ConnectionPassword" will be used for the connection.

Example of "properties" command

sqlline> !properties test.properties

Connecting to jdbc:mysql://localhost/mydb

Enter password for jdbc:mysql://localhost/mydb: *****

Connected to: MySQL (version 3.23.52-log)
Driver: MySQL-AB JDBC Driver (version 3.0.8-stable ( $Date: 2008/12/10 $, $Revision: #3 $ ))
Autocommit status: true
Transaction isolation: TRANSACTION_READ_COMMITTED

0: jdbc:mysql://localhost/mydb>

quit

Name

quit — Exit SQLLine

Synopsis

!quit

Description

Exit SQLLine.

Example of "reconnect" command

0: jdbc:hsqldb:db-hypersonic> !quit
Closing: org.hsqldb.jdbcConnection

reconnect

Name

reconnect — Reconnect to the current database

Synopsis

!reconnect

Description

Reconnect to the current database.

Example of "reconnect" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !reconnect

Reconnecting to "jdbc:oracle:thin:@localhost:1521:mydb"...

Closing: oracle.jdbc.driver.OracleConnection@4428d3

Connected to: Oracle (version Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production)
Driver: Oracle JDBC driver (version 9.0.1.0.0)
Autocommit status: true

0: jdbc:oracle:thin:@localhost:1521:mydb>

record

Name

record — Begin recording all output from SQL commands

Synopsis

!record {file name}

Description

Save all output from the session to the specified file.

Example of "record" command
0: jdbc:oracle:thin:@localhost:1521:mydb> !record /tmp/mysession.out
Saving all output to "/tmp/mysession.out". Enter "record" with no arguments to stop it.

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;
+-------------+------------+
| COMPANY_ID  |    NAME    |
+-------------+------------+
| 3           | IBM        |
| 4           | Microsoft  |
| 1           | Apple      |
| 2           | Sun        |
+-------------+------------+
4 rows selected (0.011 seconds)

0: jdbc:oracle:thin:@localhost:1521:mydb> !record
Recording stopped.

0: jdbc:oracle:thin:@localhost:1521:mydb>

rehash

Name

rehash — Obtain a list of all tables and columns from the database

Synopsis

!rehash

Description

Get a list of all tables and columns from the database in order to include them in the list for tab-completion of SQL statements. This is done automatically on connect when the fastconnect option is enabled.

Example of "rehash" command

0: jdbc:hsqldb:db-hypersonic> !rehash

Building list of tables and columns for tab-completion (set fastconnect to true to skip)...
7/7 (100%) Done
Done

0: jdbc:hsqldb:db-hypersonic>

Warning on using "rehash"
Caution

This operation can be extremely slow for some databases.

rollback

Name

rollback — Roll back the current transaction

Synopsis

!rollback

Description

Rollback the current transaction, if autocommit is off.

Example of "rollback" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !autocommit off

Autocommit status: false

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;

+-------------+------------+
| COMPANY_ID  |    NAME    |
+-------------+------------+
| 3           | IBM        |
| 4           | Microsoft  |
| 1           | Apple      |
| 2           | Sun        |
+-------------+------------+
4 rows selected (0.011 seconds)

0: jdbc:oracle:thin:@localhost:1521:mydb> DELETE FROM COMPANY;

4 rows affected (0.004 seconds)

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;

+-------------+-------+
| COMPANY_ID  | NAME  |
+-------------+-------+
|           |       |
+-------------+-------+

No rows selected (0.01 seconds)
Rollback complete (0.016 seconds)

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM COMPANY;

+-------------+------------+
| COMPANY_ID  |    NAME    |
+-------------+------------+
| 3           | IBM        |
| 4           | Microsoft  |
| 1           | Apple      |
| 2           | Sun        |
+-------------+------------+
4 rows selected (0.01 seconds)

run

Name

run — Execute a command script

Synopsis

!run {file name}

Description

Run the individual commands specified in the file name. The file should consist of individual SQL statements or SQLLine commands. Lines beginning with "#" are interpreted as comments and ignored. If any errors occur while running the script, the script will be aborted, unless the force preference is set to true.

Example of "run" command

0: jdbc:hsqldb:db-hypersonic> !run example.sql
1/11 CREATE TABLE COMPANY (COMPANY_ID INT, NAME VARCHAR(255));
No rows affected (0.001 seconds)
2/11 INSERT INTO COMPANY VALUES (1, 'Apple');
1 row affected (0 seconds)
3/11 INSERT INTO COMPANY VALUES (2, 'Sun');
1 row affected (0 seconds)
4/11 INSERT INTO COMPANY VALUES (3, 'IBM');
1 row affected (0.001 seconds)
5/11 INSERT INTO COMPANY VALUES (4, 'Microsoft');
1 row affected (0.015 seconds)
CREATE TABLE EMPLOYEE (ID INT, FIRST_NAME VARCHAR(255), LAST_NAME VARCHAR(255), COMPANY INT);
No rows affected (0.004 seconds)

CREATE INDEX FIRST_NAME ON EMPLOYEE(FIRST_NAME);
No rows affected (0 seconds)

CREATE INDEX LAST_NAME ON EMPLOYEE(LAST_NAME);
No rows affected (0.001 seconds)

ALTER TABLE EMPLOYEE ADD CONSTRAINT EMP_COMPANY FOREIGN KEY (COMPANY) REFERENCES COMPANY (COMPANY_ID);
No rows affected (0 seconds)

INSERT INTO EMPLOYEE (ID, FIRST_NAME, LAST_NAME, COMPANY) VALUES (234, 'William', 'Gates', 4);
1 row affected (0.001 seconds)

SELECT * FROM COMPANY,EMPLOYEE WHERE EMPLOYEE.COMPANY = COMPANY.COMPANY_ID;

<table>
<thead>
<tr>
<th>COMPANY_ID</th>
<th>NAME</th>
<th>ID</th>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
<th>COMPANY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Microsoft</td>
<td>234</td>
<td>William</td>
<td>Gates</td>
<td>4</td>
</tr>
</tbody>
</table>

1 row selected (0.001 seconds)

save

Name

save — Save the current preferences

Synopsis

!save

Description

Save the current preferences.

Example of "save" command

0: jdbc:oracle:thin:@localhost:1521:mydb> !save

Saving preferences to: /Users/mprudhom/.sqlline/sqlline.properties

0: jdbc:oracle:thin:@localhost:1521:mydb>

scan
**Name**

scan — Scan class path for JDBC drivers

**Synopsis**

!scan

**Description**

Scans all the jar files in the CLASSPATH for any JDBC drivers, and prints the class names of the drivers.

Example of "scan" command

sqline> !scan

35 driver classes found

<table>
<thead>
<tr>
<th>Compliant Version</th>
<th>Driver Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>4.0  COM.cloudscape.core.JDBCDriver</td>
</tr>
<tr>
<td>no</td>
<td>1.0  COM.cloudscape.core.RmiJdbcDriver</td>
</tr>
<tr>
<td>yes</td>
<td>7.1  COM.ibm.db2.jdbc.net.DB2Driver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.ddtek.jdbc.db2.DB2Driver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.ddtek.jdbc.informix.InformixDriver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.ddtek.jdbc.oracle.OracleDriver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.ddtek.jdbc.sqlserver.SQLServerDriver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.ddtek.jdbc.sybase.SqlserverDriver</td>
</tr>
<tr>
<td>yes</td>
<td>2.21  com.lapalma.jdbc.IfxDriver</td>
</tr>
<tr>
<td>no</td>
<td>0.2  com.internetcds.jdbc.tds.Driver</td>
</tr>
<tr>
<td>no</td>
<td>0.2  com.internetcds.jdbc.tds.SybaseDriver</td>
</tr>
<tr>
<td>yes</td>
<td>2.2715 com.jnetdirect.jsql.JSQLDriver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.merant.datadirect.jdbc.db2.DB2Driver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.merant.datadirect.jdbc.informix.InformixDriver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.merant.datadirect.jdbc.oracle.OracleDriver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.merant.datadirect.jdbc.sqlserver.SQLServerDriver</td>
</tr>
<tr>
<td>yes</td>
<td>3.2  com.merant.datadirect.jdbc.sybase.SqlserverDriver</td>
</tr>
<tr>
<td>yes</td>
<td>2.2  com.microsoft.jdbc.sqlserver.SQLServerDriver</td>
</tr>
<tr>
<td>no</td>
<td>3.0  com.mysql.jdbc.Driver</td>
</tr>
<tr>
<td>no</td>
<td>3.0  com.mysql.jdbc.NonRegisteringDriver</td>
</tr>
<tr>
<td>yes</td>
<td>4.4  com.pointbase.jdbc.jdbcDriver</td>
</tr>
<tr>
<td>yes</td>
<td>4.4  com.pointbase.jdbc.jdbcEmbeddedDriver</td>
</tr>
<tr>
<td>yes</td>
<td>4.4  com.pointbase.jdbc.jdbcUniversalDriver</td>
</tr>
<tr>
<td>no</td>
<td>5.5  com.sybase.jdbc2.jdbc.SybDriver</td>
</tr>
<tr>
<td>yes</td>
<td>1.0  in.co.daffodil.db.jdbc.DaffodilDBDriver</td>
</tr>
<tr>
<td>no</td>
<td>2.0  interbase.interclient.Driver</td>
</tr>
<tr>
<td>no</td>
<td>0.5  net.sourceforge.jtds.jdbc.Driver</td>
</tr>
<tr>
<td>yes</td>
<td>1.0  oracle.jdbc.OracleDriver</td>
</tr>
<tr>
<td>yes</td>
<td>1.0  oracle.jdbc.driver.OracleDriver</td>
</tr>
</tbody>
</table>
no 3.26 org.enhydra.instantdb.jdbc.idbDriver
no 3.0 org.gjt.mm.mysql.Driver
no 1.7 org.hsqldb.jdbcDriver
no 7.3 org.postgresql.Driver
yes 0.9 org.sourceforge.jxdbcon.JXDBConDriver

sqlline>
schemas

Name

schemas — List all the schemas in the database

Synopsis

!schemas

Description

List all the schemas in the database of the current connection.

Example of "schemas" command

0: jdbc:postgresql:thin:@localhost:5432:mydb> !schemas

+-------------------+---------------+
|    table_schem     | table_catalog |
+-------------------+---------------+
| information_schema |               |
| pg_catalog         |               |
| public             |               |
+-------------------+---------------+

0: jdbc:postgresql:thin:@localhost:5432:mydb>

script

Name

script — Save executed commands to a file

Synopsis

!script {filename}

Description
Start saving all commands entered to the specified file. Once scripting on, it can be finished by entering the script command again with no arguments. Saved scripts can be re-run using the run command.

Example of "script" command

0: jdbc:hsqldb:db-hypersonic> !script /tmp/mysession.script

Saving command script to "/tmp/mysession.script". Enter "script" with no arguments to stop it.

0: jdbc:hsqldb:db-hypersonic> !autocommit off

Autocommit status: false

0: jdbc:hsqldb:db-hypersonic> SELECT * FROM COMPANY;

+-------------+------------+
| COMPANY_ID  |    NAME    |
+-------------+------------+
| 1           | Apple      |
| 2           | Sun        |
| 3           | IBM        |
| 4           | Microsoft  |
+-------------+------------+
4 rows selected (0.001 seconds)

0: jdbc:hsqldb:db-hypersonic> UPDATE COMPANY SET NAME='Apple, Inc.' WHERE NAME='Apple';

1 row affected (0.005 seconds)

0: jdbc:hsqldb:db-hypersonic> !commit

Commit complete (0 seconds)

0: jdbc:hsqldb:db-hypersonic> !script

Script closed. Enter "run /tmp/mysession.script" to replay it.

0: jdbc:hsqldb:db-hypersonic>

set

Name

set — Set a preference

Synopsis

!set {preference} {value}
Description

Set the specified preference to the specified value. If autosave is on, then this will cause preferences to be immediately saved. Otherwise, changed setting can be saved for future SQLLine session either the save command.

Example of "set" command

0: jdbc:hsqldb:db-hypersonic> !set color on

sql

Name

sql — Execute a SQL against a database

Synopsis

!sql {statement}

Description

Issue the specified SQL statement. This is equivalent to entering the statement by itself from the command line.

Example of "sql" command

0: jdbc:hsqldb:db-hypersonic> !sql DELETE FROM COMPANY;

4 rows affected (0.005 seconds)

0: jdbc:hsqldb:db-hypersonic>

tables

Name

tables — List all the tables in the database

Synopsis

!tables

Description

List all the tables in the database of the current connection.

Example of "tables" command
0: jdbc:oracle:thin:@localhost:1521:mydb> !tables

+-----------------+------------+-----------------------------+-------------+
| TABLE_CAT | TABLE_SCHEM |           TABLE_NAME        | TABLE_TYPE |
+-----------+------------+-----------------------------+-------------+
|          | MYDB       | COMPANY                     | TABLE       |
|          | MYDB       | EMPLOYEE                    | TABLE       |
+-----------+------------+-----------------------------+-------------+

0: jdbc:oracle:thin:@localhost:1521:mydb>

verbose

Name

verbose — Enable verbose output

Synopsis

!verbose

Description

Enable verbose mode, which causes stack traces to be printed when errors occur, and enabled outputting of debug information.

Example of "verbose" command

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM BOGUS_TABLE;

Error: ORA-00942: table or view does not exist (state=42000,code=942)

0: jdbc:oracle:thin:@localhost:1521:mydb> !verbose

verbose: on

0: jdbc:oracle:thin:@localhost:1521:mydb> SELECT * FROM BOGUS_TABLE;

Error: ORA-00942: table or view does not exist (state=42000,code=942)
java.sql.SQLException: ORA-00942: table or view does not exist

at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:180)
at oracle.jdbc.ttc7.TTIoer.processError(TTIoer.java:208)
at oracle.jdbc.ttc7.Oall7.receive(Oall7.java:543)
at oracle.jdbc.ttc7.TTC7Protocol.doOall7(TTC7Protocol.java:1451)
at oracle.jdbc.ttc7.TTC7Protocol.parseExecuteDescribe(TTC7Protocol.java:651)
at oracle.jdbc.driver.OracleStatement.doExecuteQuery(OracleStatement.java:2110)
at oracle.jdbc.driver.OracleStatement.doExecuteWithTimeout(OracleStatement.java:2324)
Parameter Reference

autocommit
autosave
color
colorScheme
confirm
confirmPattern
csvDelimiter
csvQuoteCharacter
dateformat
escapeOutput
fastconnect
force
headerinterval
historyfile
incremental
isolation
maxcolumnwidth
maxHistoryFileRows
maxHistoryRows
maxwidth
mode
nullValue
numberformat
outputformat
prompt
rightprompt
rowlimit
showheader
shownestederrs
showtime
showwarnings
silent
strictJdbc
timeformat
timestampformat
trimscripts
uselinecontinuation
verbose
autocommit

If true, then new connections will have autocommit set, otherwise, transactions will need to be explicitly committed or rolled back. Defaults to true. To change the autocommit status for a connection that is already open and active, use the autocommit command instead.

color

If true, then output to the terminal will use color for a more pleasing visual experience. Requires that the terminal support ANSI control codes (most do). Defaults to false.

colorScheme

If chester/dark/dracula/light/obsidian/solarized/vs2010, then this scheme will be used for command/sql syntax highlighting. If default then there is no syntax highlighting.

confirm

If true, then user will be prompted for confirmation if commands matching with confirmPattern are executed.

confirmPattern

If default with confirm = true, user will be prompted for DROP and DELETE commands, otherwise, for the commands as matched in the pattern.

csvDelimiter

Sets the delimiter in csv outputFormat. Setting to default causes usage of "," as a delimiter.

csvQuoteCharacter

Sets the quote character in csv outputFormat. Setting to default causes usage of "\" as a delimiter.

dateformat

The format for how date values are displayed. Setting to default causes date values to be fetched and rendered via ResultSet.getString. Any other setting results in fetch via ResultSet.getObject and rendering via java.text.SimpleDateFormat. For example, the setting "YYYY_MM_dd" yields values like "1970_01_01".

escapeOutput

When set to true, control symbols in output will be escaped. Otherwise they will be printed as is. Default is false.
fastconnect

When false, any new connection will cause SQLLine to access information about the available tables and columns in order to provide them as candidates for tab-completion. This can be a very slow operation for some databases, do by default it is off. Table and column information can always be explicitly retrieved using the rehash command.

force

When set to false, any failures during the execution of the run or record commands will cause execution to abort, preventing the execution of subsequent commands. When set to true, errors will be displayed but otherwise ignored. Defaults to false.

headerinterval

The interval between which the column headers will be redisplayed when using the "table" outputformat. Defaults to 100.

historyfile

The file to which SQLLine will save a record of all the commands issued. Defaults to HOME/sqlline/history on Windows, and HOME/.sqlline/history on all other platforms.

incremental

When set to false, the entire result set is fetched and buffered before being displayed, yielding optimal display column sizing. When set to true, result rows are displayed immediately as they are fetched, yielding lower latency and memory usage at the price of extra display column padding. Defaults to false.

incrementalBufferRows

When incremental set to false and the entire result set has more rows than value of incrementalBufferRows, sqlline will enter incremental mode. In that case, result rows are displayed by batches of size incrementalBufferRows. If value is negative, sqlline never enters incremental mode. Defaults to 1000.

isolation

The default transaction isolation that will be used for new connections. To change the isolation level of the currently active connection, use the isolation command instead.

maxcolumnwidth

The maximum column width to display for each column before truncating data when using the "table" outputformat. Defaults to 15.

maxHistoryFileRows

The maximum number of history rows to store in history file. Defaults to 10000 (org.jline.reader.impl.history.DefaultHistory.DEFAULT_HISTORY_FILE_SIZE).
maxHistoryRows

The maximum number of history rows to store in memory. Defaults to 500 (org.jline.reader.impl.history.DefaultHistory.DEFAULT_HISTORY_SIZE).

maxwidth

The maximum width to display before truncating data when using the "table" outputformat. Defaults to attempting to query the terminal for the current width, falls back to 80.

mode

The editing mode to use in sqlline. Defaults to "emacs".

nullvalue

The value which will be used instead of null values. Setting to default causes 'null' for numbers/dates/times and empty strings for others. For example, the setting *** yields null values like '***'

numberformat

The format for how numeric values are displayed. Setting to default causes numeric values to be fetched and rendered via ResultSet.getString. Any other setting results in fetch via ResultSet.getObject and rendering via java.text.DecimalFormat. For example, the setting "0.###E0" yields scientific notation with up to three fractional digits, values like "6.022E23".

outputformat

The format for how results are displayed. For details, see the information on the outputformat command.

prompt

The tcsh-like format for how prompt is displayed. For example, the setting ‘%[033[1;33m]sqlline%[033[m]>’ yields yellow sqlline and normal angle bracket. Defaults to 'sqlline>’. If for the specific database connection nickname is set then nickname will be used.

rightprompt

The tcsh-like format for how right prompt is displayed. The same patterns are applied as for prompt but it does not care if nickname is set or not. Defaults to “”.

promptscript

A JavaScript program that is evaluated to generate a prompt. If specified, prompt is ignored.

The following variables are pre-defined for use in the JavaScript: connectionIndex currentSchema, databaseProductName, url, userName. For example:
sqlline> !set promptscript 'databaseProductName.toLowerCase() + ":" + connectionIndex + " > "'
hsql database engine:0>

rowlimit

The maximum number of rows to fetch per query. Defaults to 0, which is interpreted as fetching all rows.

showheader

If true, display the names of the columns when displaying results. Defaults to true.

shownestederrs

If true, display any nested errors that are reported on the connection after issuing any database commands. Defaults to false, meaning only the topmost error is displayed.

showtime

If true, display execution time for database commands when in verbose mode. Defaults to true.

showwarnings

If true, display any warnings that are reported on the connection after issuing any database commands. Defaults to true.

silent

If true, then reduce the amount of informational messages displayed. Useful for redirecting a sqlline command to a file for later parsing. Defaults to false.

strictJdbc

If false, then use default values defined by java.sql.DatabaseMetaData in case an implementation of this interface fails with e.g. "Method not supported", otherwise it also fails. Defaults to false.

timeformat

The format for how time values are displayed. Setting to default causes time values to be fetched and rendered via ResultSet.getString. Any other setting results in fetch via ResultSet.getObject and rendering via java.text.SimpleDateFormat. For example, the setting "HH:mm:ss" yields values like "14:12:11".

timestampformat

The format for how timestamp values are displayed. Setting to default causes timestamp values to be fetched and rendered via ResultSet.getString. Any other setting results in fetch via ResultSet.getObject and rendering via java.text.SimpleDateFormat. For example, the setting "dd/MM/YYYY'T'HH:mm:ss" yields values like "01/01/1970T12:32:12".
trimscripts

If true, then trim leading and trailing whitespace from lines as they are processed in scripts; otherwise, preserve whitespace. Defaults to true.

uselinecontinuation

Use line continuation if true. Defaults to true.

verbose

If true, then print out the entire java stack trace whenever an error occurs, as well as displaying debugging information. Defaults to false.

JDBC Driver Support

Project Information

Found in path(s):
* /opt/cola/permits/1179408911_1626107141.97/0/sqlline-1-9-0-sources-jar/sqlline/manual.txt

1.564 aws-java-sdk-core 1.11.837

1.564.1 Available under license:
Apache License
Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to
4. If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

1.565 xerces-j 2.11.0
1.565.1 Available under license:
No license file was found, but licenses were detected in source scan.

UniqueParticleAttribution = Error: cos-nonambig: ((0)) and ([1]) violate the "Unique Particle Attribution" rule.
cos-nonambig = cos-nonambig: {0} and [1] (or elements from their substitution group) violate "Unique Particle Attribution".

Found in path(s):
* /opt/cola/permits/1124833062_1611183141.82/0/xercesimpl-zip/xercesimpl-jar/org/apache/xerces/impl/msg/XMLSchemaMessages.properties

1.566 opentest4j 1.2.0
1.566.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-2018 the original author or authors.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1145468562_1616183623.24/0/opentest4j-1-2-0-sources-1-jar/org/opentest4j/TestSkippedException.java
* /opt/cola/permits/1145468562_1616183623.24/0/opentest4j-1-2-0-sources-1-jar/org/opentest4j/MultipleFailuresError.java
* /opt/cola/permits/1145468562_1616183623.24/0/opentest4j-1-2-0-sources-1-jar/org/opentest4j/AssertionFailedError.java
1.567 bitsyntax 0.0.4 0.0.4
1.567.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.568 bunyan-format 0.2.1
1.568.1 Available under license:
Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.569 babel-plugin-transform-es2015-duplicate-keys 6.24.1

1.569.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The "Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.570 commons-compress 1.19

1.570.1 Available under license:
Apache Commons Compress
Copyright 2002-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

The files in the package org.apache.commons.compress.archivers.sevenz were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/), which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed...
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.571 hbase-shaded-protobuf 3.4.1
1.571.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---

This project incorporates portions of the 'Protocol Buffers' project available
under a '3-clause BSD' license.

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

Apache HBase Patched and Relocated (Shaded) Protobuf
Copyright 2017-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

---

// Protocol Buffers - Google's data interchange format
// Copyright 2008 Google Inc. All rights reserved.
// https://developers.google.com/protocol-buffers/

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.

// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.572 pify 4.0.1
1.572.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.573 p-locate 3.0.0
1.573.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.574 asm-analysis 7.3.1
1.574.1 Available under license:

No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
//    this software without specific prior written permission.
/**
 * Constructs a new {@link SourceInterpreter} for the latest ASM API version. <i>Subclasses must
 * not use this constructor</i>. Instead, they must use the {@link #SourceInterpreter(int)}
 * version.
 */

Found in path(s):
* /opt/cola/permits/1135907546_1613628471.63/0/asm-analysis-7-3-1-sources-1-
  jar/org/objectweb/asm/tree/analysis/SourceInterpreter.java
No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
//    this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1135907546_1613628471.63/0/asm-analysis-7-3-1-sources-1-
  jar/org/objectweb/asm/tree/analysis/BasicValue.java
* /opt/cola/permits/1135907546_1613628471.63/0/asm-analysis-7-3-1-sources-1-
  jar/org/objectweb/asm/tree/analysis/AnalyzerException.java
* /opt/cola/permits/1135907546_1613628471.63/0/asm-analysis-7-3-1-sources-1-
  jar/org/objectweb/asm/tree/analysis/Value.java
* /opt/cola/permits/1135907546_1613628471.63/0/asm-analysis-7-3-1-sources-1-
  jar/org/objectweb/asm/tree/analysis/Verifier.java
* /opt/cola/permits/1135907546_1613628471.63/0/asm-analysis-7-3-1-sources-1-
  jar/org/objectweb/asm/tree/analysis/Frame.java
No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1135907546_1613628471.63/0/asm-analysis-7-3-1-sources-1-jar/org/objectweb/asm/tree/analysis/package.html
No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
// this software without specific prior written permission.
/**
 * Constructs a new {@link BasicInterpreter} for the latest ASM API version. <i>Subclasses must
 * not use this constructor</i>. Instead, they must use the {@link #BasicInterpreter(int)}
 * version.
 */

Found in path(s):
* /opt/cola/permits/1135907546_1613628471.63/0/asm-analysis-7-3-1-sources-1-
jar/org/objectweb/asm/tree/analysis/BasicInterpreter.java

1.575 commons-compress 1.20

1.575.1 Available under license:
Apache Commons Compress
Copyright 2002-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

---

The files in the package org.apache.commons.compress.archivers.sevenz
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)

---

The test file lbzip2_32767.bz2 has been copied from libbzip2's source
repository:

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2019 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions

Open Source Used In DNAC Platform Antelope 2010
are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@acm.org

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.576 hsts 1.0.0
1.576.1 Available under license :
   The MIT License (MIT)

Copyright (c) 2014-2015 Evan Hahn, Adam Baldwin
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.577 figures 2.0.0
1.577.1 Available under license:
   The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.578 lodash.once 4.1.1
1.578.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously
marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of
any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.580 @babel/plugin-transform-block-scoped-functions 7.12.13

1.580.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.581 progress 1.1.8

1.581.1 Available under license:
(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.582 methods 1.1.2

1.582.1 Available under license:
(The MIT License)

Copyright (c) 2013-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015-2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.583 jnr-constants 0.9.0

1.584 output-file-sync 1.1.2
1.584.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 - 2016 Shinnosuke Watanabe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.585 @babel/plugin-transform-sticky-regex
7.12.13
1.585.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.586 @babel/plugin-transform-classes 7.13.0
1.586.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
1.587 http-status-codes 1.4.0 1.4.0

1.587.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.588 combined-stream 1.0.6 1.0.6

1.588.1 Available under license:
Copyright (c) 2011 Debuggable Limited <felix@debuggable.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.589 classgraph 4.6.32

1.590 lie 3.3.0

1.590.1 Available under license :

#Copyright (c) 2014-2018 Calvin Metcalf, Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

1.591 x-xss-protection 1.0.0 1.0.0

1.591.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
1.592 commons-lang3 3.8.1

1.592.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the
Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works
as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.593 babel-traverse 6.26.0 6.26.0

1.593.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
defurnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.594 generate-schema 2.6.0 2.6.0
1.594.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Nijiko Yonskai

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.595 groovy-docgenerator 3.0.8
1.595.1 Available under license :
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE
Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.

Found in path(s):
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/template.allclasses-frame.html
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/index.html
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/template.overview-frame.html
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/template.package-summary.html
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/template.index-all.html
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/overview-summary.html
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/template.package-frame.html
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/template.class.html
No license file was found, but licenses were detected in source scan.

/*!
normalize.css v2.1.0 | MIT License | git.io/normalize */

Found in path(s):
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/stylesheet.css
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an

Open Source Used In DNAC Platform Antelope 2032
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*

Found in path(s):
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/DocGenerator.groovy
No license file was found, but licenses were detected in source scan.

# to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# software distributed under the License is distributed on an

Found in path(s):
* /opt/cola/permits/1218307070_1634931267.31/0/groovy-docgenerator-3-0-8-sources-jar/org/apache/groovy/docgenerator/DocGenerator.properties

1.596 httpcore5-h 5.0.2
1.596.1 Available under license:

Apache HttpComponents Core HTTP/2
Copyright 2005-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.597 cglib 2.2.0-b21
1.597.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2004 The Apache Software Foundation
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/CallbackHelper.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/RejectModifierPredicate.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/MixinEverythingEmitter.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/MethodInfoTransformer.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/transform/impl/AddInitTransformer.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/MethodInfo.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/InterfaceMaker.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/CallbackInfo.java

No license file was found, but licenses were detected in source scan.

2012 Oracle and/or its affiliates. All rights reserved.

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can
obtain a copy of the License at
https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
or packager/legal/LICENSE.txt. See the License for the specific
language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each
file and include the License file at packager/legal/LICENSE.txt.

GPL Classpath Exception:
Oracle designates this particular file as subject to the "Classpath"
exception as provided by Oracle in the GPL Version 2 section of the License
file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields
enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or
only the GPL Version 2, indicate your decision by adding "[Contributor]
elects to include this software in this distribution under the [CDDL or GPL
Version 2] license." If you don't indicate a single choice of license, a
recipient has the option to distribute your version of this file under
either the CDDL, the GPL Version 2 or to extend the choice of license to
its licensees as provided above. However, if you add GPL Version 2 code
and therefore, elected the GPL Version 2 license, then the option applies
only if the new code is made subject to such option by the copyright
holder.

Found in path(s):
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/pom.xml
No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2005 INRIA, France Telecom
* All rights reserved.
* *
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/LocalVariablesSorter.java
No license file was found, but licenses were detected in source scan.

*/

* Copyright 2003,2004 The Apache Software Foundation
* Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/DebuggingClassWriter.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/LazyLoaderGenerator.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/CollectionUtils.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/CallbackGenerator.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/transform/ClassFilterTransformer.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/FixedValueGenerator.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/ReflectUtils.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/transform/impl/AddStaticInitTransformer.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/reflect/FastClassEmitter.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/beans/BulkBeanEmitter.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/MethodInterceptorGenerator.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/beans/ImmutableBean.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/beans/BeanCopier.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/DefaultNamingPolicy.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/CodeEmitter.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/MixinEmitter.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/EmitUtils.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/TypeUtils.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/Mixin.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/transform/AbstractClassLoader.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/Proxy.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/reflect/FastMethod.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/NoOpGenerator.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/reflect/FastClass.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/DispatcherGenerator.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/KeyFactory.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/InvocationHandlerGenerator.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/VisibilityPredicate.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/reflect/MethodDelegate.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/CallbackFilter.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/MethodWrapper.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/core/AbstractClassGenerator.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/beans/BeanMap.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/beans/BeanMap.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-
jar/net/sf/cglib/reflect/MulticastDelegate.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-
jar/net/sf/cglib/proxy/MethodProxy.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-
jar/net/sf/cglib/beans/BeanMapEmitter.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-
jar/net/sf/cglib/transform/ClassTransformerChain.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2003,2004 The Apache Software Foundation
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-
jar/net/sf/cglib/transform/AbstractTransformTask.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2003 The Apache Software Foundation
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2002,2003 The Apache Software Foundation
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 *      http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/UndeclaredThrowableException.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/MethodInterceptor.java
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/Factory.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2002,2003,2004 The Apache Software Foundation
 */

Open Source Used In DNAC Platform Antelope
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1136701410_1613864270.85/0/cglib-2-2-0-b21-sources-2-jar/net/sf/cglib/proxy/Enhancer.java

1.598 libpciaccess 0.13.4 1
1.598.1 Available under license:

(C) Copyright IBM Corporation 2006, 2007
(C) Copyright Eric Anholt 2006
(C) Copyright Mark Kettenis 2011
(C) Copyright Robert Millan 2012
Copyright 2009, 2012 Red Hat, Inc.
All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation on the rights to use, copy, modify, merge, publish, distribute, sub license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL IBM AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2008 Juan Romero Pardines
Copyright (c) 2008, 2011 Mark Kettenis
Copyright (c) 2009 Michael Lorenz
Copyright (c) 2009, 2012 Samuel Thibault

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (C) 2000 The XFree86 Project, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
XFree86 Project BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the XFree86 Project shall
not be used in advertising or otherwise to promote the sale, use or other
dealings in this Software without prior written authorization from the
XFree86 Project.

Copyright (c) 2007 Paulo R. Zanoni, Tiago Vignatti
Copyright (c) 2009 Tiago Vignatti

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.599 babel-helper-hoist-variables 6.24.1

1.599.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.
MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The "Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.600 engine.io-client 3.4.4

1.600.1 Available under license:

(The MIT License)

Copyright (c) 2014-2015 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.601 jackson-dataformat-yaml 2.10.0
1.601.1 Available under license:

    # Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses. This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.602 open-csv 5.4
1.602.1 Available under license:

No license file was found, but licenses were detected in source scan.

<url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
Open Source Used In DNAC Platform Antelope 2053

No license file was found, but licenses were detected in source scan.

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,
1.603 object.assign 4.1.1

1.603.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.604 has-values 1.0.0

1.604.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.605 websocket-client 9.3.27.v20190418

1.606 libcroco 0.6.11 1

1.606.1 Available under license :
GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's
free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don’t assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library
General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)
Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception,
the source code distributed need not include anything that is normally
distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.607 domutils 1.7.0 1.7.0
1.607.1 Available under license:
Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.608 on-headers 1.0.1 1.0.1

1.608.1 Available under license:
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.609 batik-svgpp 1.13

1.609.1 Available under license:

Apache Batik
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the
Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for
Standardization for the definition of character entities used in the software's
documentation.

This product includes images from the Tango Desktop Project
(http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.610 cheerio 0.22.0

1.610.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
1.611 destroy 1.0.4

1.611.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.612 commons-math 2.1

1.612.1 Available under license:

Apache Commons Math
Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

===============================================================================

The LinearConstraint, LinearObjectiveFunction, LinearOptimizer, RelationShip, SimplexSolver and SimplexTableau classes in package org.apache.commons.math.optimization.linear include software developed by Benjamin McCann (http://www.benmccann.com) and distributed with the following copyright: Copyright 2009 Google Inc.

===============================================================================

This product includes software developed by the University of Chicago, as Operator of Argonne National Laboratory.

The LevenbergMarquardtOptimizer class in package org.apache.commons.math.optimization.general includes software translated from the lmder, lmpar and qrsolv Fortran routines from the Minpack package.

Minpack Copyright Notice (1999) University of Chicago. All rights reserved

===============================================================================

The GraggBulirschStoerIntegrator class in package org.apache.commons.math.ode.nonstiff includes software translated from the odex Fortran routine developed by E. Hairer and G. Wanner.

Original source copyright:
Copyright (c) 2004, Ernst Hairer

===============================================================================

The EigenDecompositionImpl class in package org.apache.commons.math.linear includes software translated from some LAPACK Fortran routines. Original source copyright:
Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

===============================================================================

The MersenneTwister class in package org.apache.commons.math.random includes software translated from the 2002-01-26 version of the Mersenne-Twister generator written in C by Makoto Matsumoto and Takuji Nishimura. Original source copyright:
Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
All rights reserved

===============================================================================

The complete text of licenses and disclaimers associated with the the original sources enumerated above at the time of code translation are in the LICENSE.txt file.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE COMMONS MATH DERIVATIVE WORKS:

The Apache commons-math library includes a number of subcomponents
whose implementation is derived from original sources written
in C or Fortran. License terms of the original sources
are reproduced below.

===============================================================================
For the lmder, lmpar and qrsolv Fortran routine from minpack and translated in
the LevenbergMarquardtOptimizer class in package
org.apache.commons.math.optimization.general
Original source copyright and license statement:

Minpack Copyright Notice (1999) University of Chicago. All rights reserved

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above
copyright notice, this list of conditions and the following
disclaimer.

2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials
provided with the distribution.

3. The end-user documentation included with the
redistribution, if any, must include the following
acknowledgment:

"This product includes software developed by the
University of Chicago, as Operator of Argonne National
Laboratory.

Alternately, this acknowledgment may appear in the software
itself, if and wherever such third-party acknowledgments
normally appear.
4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.

5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE, EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGES.

===============================================================================

Copyright and license statement for the odex Fortran routine developed by E. Hairer and G. Wanner and translated in GraggBulirschStoerIntegrator class in package org.apache.commons.math.ode.nonstiff:

Copyright (c) 2004, Ernst Hairer

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===============================================================================

Copyright and license statement for the original lapack fortran routines translated in EigenDecompositionImpl class in package org.apache.commons.math.linear:

Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

$COPYRIGHT$

Additional copyrights may follow

$HEADER$

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.

- Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright and license statement for the original Mersenne twister C
routines translated in MersenneTwister class in package
org.apache.commons.math.random:

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. The names of its contributors may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.613 loose-envify 1.4.0
1.613.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Andres Suarez <zertos@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.614 object-inspect 1.10.3 1.10.3
1.614.1 Available under license :
MIT License

Copyright (c) 2013 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.615 is-dotfile 1.0.3
1.615.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.616 ajv-keywords 3.2.0 3.2.0

1.616.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.617 path-to-regexp 2.1.0
1.617.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.618 hive-hbase-handler 3.1.2
1.618.1 Available under license:

Hive HBase Handler
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must...
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.619 setprototypeof 1.1.1

1.619.1 Available under license:
Copyright (c) 2015, Wes Todd

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.620 callsite 1.0.0

1.620.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.621 @babel/plugin-transform-block-scoping

7.12.13

1.621.1 Available under license:
    MIT License

    Copyright (c) 2014-present Sebastian McKenzie and other contributors

    Permission is hereby granted, free of charge, to any person obtaining
    a copy of this software and associated documentation files (the
    "Software"), to deal in the Software without restriction, including
    without limitation the rights to use, copy, modify, merge, publish,
    distribute, sublicense, and/or sell copies of the Software, and to
    permit persons to whom the Software is furnished to do so, subject to
    the following conditions:

    The above copyright notice and this permission notice shall be
    included in all copies or substantial portions of the Software.

    THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
    EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
    MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
    NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
    LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
    OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
    WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.622 icu 65

1.622.1 Available under license:

    #. -*. makefile.-*
    #. Copyright (C) 2016 and later: Unicode, Inc. and others.
    #. License & terms of use: http://www.unicode.org/copyright.html
    #. BSD-specific setup (FreeBSD, OpenBSD, NetBSD, *BSD)
    #. Copyright (c) 1999-2013, International Business Machines Corporation and
      #. others. All Rights Reserved.

    #. Commands to generate dependency files
    GEN_DEPS.c=$(CC) -E -MM $(DEFS) $(CPPFLAGS)
    GEN_DEPS.cc=$(CXX) -E -MM $(DEFS) $(CPPFLAGS)

    #. Flags for position independent code
    SHAREDLIBCFLAGS = -fPIC
    SHAREDLIBCXXFLAGS = -fPIC
    SHAREDLIBCPPFLAGS = -DPIC
## Additional flags when building libraries and with threads

THREADSCPPFLAGS = -D_REENTRANT
LIBCPPFLAGS =

## Compiler switch to embed a runtime search path

LD_RPATH=
LD_RPATH_PRE=-Wl,-rpath,

## Compiler switch to embed a library name

LD_SONAME = -Wl,-soname -Wl,$(notdir $(MIDDLE_SO_TARGET))

## Shared library options

LD_SOOPTIONS=-Wl,-Bsymbolic

## Shared object suffix

SO = so

## Non-shared intermediate object suffix

STATIC_O = ao

## Compilation rules

%.$(STATIC_O): $(srcdir)/%.c
$(COMPILE.c) $(STATICCPPFLAGS) $(STATICCFLAGS) -o $@ $<
%.o: $(srcdir)/%.c
$(COMPILE.c) $(DYNAMICCPPFLAGS) $(DYNAMICCFLAGS) -o $@ $<

%.$(STATIC_O): $(srcdir)/%.cpp
$(COMPILE.cc) $(STATICCPPFLAGS) $(STATICCXXFLAGS) -o $@ $<
%.o: $(srcdir)/%.cpp
$(COMPILE.cc) $(DYNAMICCPPFLAGS) $(DYNAMICCXXFLAGS) -o $@ $<

## Dependency rules

%.d: $(srcdir)/%.c
@echo "generating dependency information for $<"
@(SHELL) -ec '$(GEN_DEPS.c) $< \n | sed \"s/\($*\)$i\o\: ]*\A1.o $@ : /g\" > $@; \n [-s $@ ] \|| rm -f $@'

%.d: $(srcdir)/%.cpp
@echo "generating dependency information for $<"
@(SHELL) -ec '$(GEN_DEPS.cc) $< \n | sed \"s/\($*\)$i\o\: ]*\A1.o $@ : /g\" > $@; \n [-s $@ ] \|| rm -f $@'

## Versioned libraries rules

%.$(SO).$(SO_TARGET_VERSION_MAJOR): %.$(SO).$(SO_TARGET_VERSION)
## Bind internal references

# LDflags that pkgdata will use
BIR_LDFLAGS= -Wl,-Bsymbolic

# Dependencies [i.e. map files] for the final library
BIR_DEPS=

## Remove shared library 's'
STATIC_PREFIX_WHEN_USED =
STATIC_PREFIX =

## End BSD-specific setup

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2020 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in https://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining
a copy of the Unicode data files and any associated documentation
(the "Data Files") or Unicode software and any associated documentation
(the "Software") to deal in the Data Files or Software
without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, and/or sell copies of
the Data Files or Software, and to permit persons to whom the Data Files
or Software are furnished to do so, provided that either
(a) this copyright and permission notice appear with all copies
of the Data Files or Software, or
(b) this copyright and permission notice appear in associated
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale,
use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

---------------------

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

The Google Chrome software developed by Google is licensed under the BSD license. Other software included in this distribution is provided under other licenses, as set forth below.

The BSD License
http://opensource.org/licenses/bsd-license.php
Copyright (C) 2006-2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

 Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The word list in cjdict.txt are generated by combining three word lists listed below with further processing for compound word breaking. The frequency is generated with an iterative training against Google web corpora.

Libtabe (Chinese)
- https://sourceforge.net/project/?group_id=1519
- Its license terms and conditions are shown below.

IPADIC (Japanese)
# - http://chasen.aist-nara.ac.jp/chasen/distribution.html
# - Its license terms and conditions are shown below.
#
# ---------COPYING.libtabe ---- BEGIN-------------------
#
# /*
# * Copyright (c) 1999 TaBE Project.
# * Copyright (c) 1999 Pai-Hsiang Hsiao.
# * All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * . Neither the name of the TaBE Project nor the names of its
# * contributors may be used to endorse or promote products derived
# * from this software without specific prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# /*
# * Copyright (c) 1999 Computer Systems and Communication Lab,
# * Institute of Information Science, Academia
# * Sinica. All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * Neither the name of the Computer Systems and Communication Lab
# * nor the names of its contributors may be used to endorse or
# * promote products derived from this software without specific
# * prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-tsa4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsa4
#
# ---------------COPYING.libtable-----END--------------------------------
#
# ---------------COPYING.ipadic-----BEGIN-------------------------------
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise. The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
# project or thereafter. Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# ---------------------------------------COPYING.ipadic-----END----------------------------------

3. Lao Word Break Dictionary Data (laodict.txt)

# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
# Project: http://code.google.com/p/lao-dictionary/
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
#             (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# ---------------------------------------COPYING.ipadic-----END----------------------------------

# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
#
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
4. Burmese Word Break Dictionary Data (burmesedict.txt)

    # Copyright (c) 2014 International Business Machines Corporation
    # and others. All Rights Reserved.
    #
    # This list is part of a project hosted at:
    # github.com/kanyawtech/myanmar-karen-word-lists
    #
    # **************************************************************************
    # Copyright (c) 2013, LeRoy Benjamin Sharon
    # All rights reserved.
    #
    # Redistribution and use in source and binary forms, with or without
    # modification, are permitted provided that the following conditions
    # are met: Redistributions of source code must retain the above
    # copyright notice, this list of conditions and the following
    # disclaimer. Redistributions in binary form must reproduce the
    # above copyright notice, this list of conditions and the following
    # disclaimer in the documentation and/or other materials provided
    # with the distribution.
    #
    # Neither the name Myanmar Karen Word Lists, nor the names of its
    # contributors may be used to endorse or promote products derived
    # from this software without specific prior written permission.
    #
    # THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
    # CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
    # INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
    # MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
    # DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
    # BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
    # EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
    # TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
    # DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
    # ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
    # TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
    # THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
    # SUCH DAMAGE.
    # **************************************************************************

5. Time Zone Database

    ICU uses the public domain data and code derived from Time Zone
    Database for its time zone support. The ownership of the TZ database
is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

# 7. Database Ownership

# The TZ database itself is not an IETF Contribution or an IETF document. Rather it is a pre-existing and regularly updated work that is in the public domain, and is intended to remain in the public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do not apply to the TZ Database or contributions that individuals make to it. Should any claims be made and substantiated against the TZ Database, the organization that is providing the IANA Considerations defined in this RFC, under the memorandum of understanding with the IETF, currently ICANN, may act in accordance with all competent court orders. No ownership claims will be made by ICANN or the IETF Trust on the database or the code. Any person making a contribution to the database or code waives all rights to future claims in that contribution or in the TZ Database.

6. Google double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.623 parseuri 0.0.5 0.0.5

1.623.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.624 lodash.reduce 4.6.0

1.624.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.625 javaewah 0.3.2

1.625.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2009-2012, Daniel Lemire
 * Licensed under APL 2.0.
 */

Found in path(s):
* /opt/cola/permits/1110827308_1606855614.93/0/javaewah-0-3-2-sources-1-jar/javaewah/IntIterator.java
* /opt/cola/permits/1110827308_1606855614.93/0/javaewah-0-3-2-sources-1-jar/javaewah/BufferedRunningLengthWord.java
* /opt/cola/permits/1110827308_1606855614.93/0/javaewah-0-3-2-sources-1-jar/javaewah/EWAHIterator.java
* /opt/cola/permits/1110827308_1606855614.93/0/javaewah-0-3-2-sources-1-
1.626 jackson-databind 2.4.0

1.626.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

1.627 babel-plugin-check-es2015-constants 6.22.0

1.627.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
   following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
   from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
1.628 jetty-servlets 9.4.41.v20210516

1.628.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient’s responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such
Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.
5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
In order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

===================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
===================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and
* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following
artifacts may be included in the distribution or downloaded when ALPN
module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications
are hosted at github and both modified and original are under GPL v2 with
classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2
The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

### 1.629 groovy-testng 3.0.8

#### 1.629.1 Available under license:

No license file was found, but licenses were detected in source scan.

```java
/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1218318693_1634926484.03/0/groovy-testng-3-0-8-sources-jar/org/apache/groovy/plugin/testng/TestNgRunner.java
* /opt/cola/permits/1218318693_1634926484.03/0/groovy-testng-3-0-8-sources-jar/org/codehaus/groovy/testng/TestNgRunner.java

No license file was found, but licenses were detected in source scan.

#
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
```
# with the License. You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.
#
org.apache.groovy.plugin.testng.TestNgRunner

Found in path(s):
* /opt/cola/permits/1218318693_1634926484.03/0/groovy-testng-3-0-8-sources-jar/META-INF/services/org.apache.groovy.plugin.GroovyRunner

1.630 regenerator-runtime 0.10.5
1.630.1 Available under license :
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2014, Facebook, Inc.
 * All rights reserved.
 *
 * This source code is licensed under the BSD-style license found in the
 * https://raw.github.com/facebook/regenerator/master/LICENSE file. An
 * additional grant of patent rights can be found in the PATENTS file in
 * the same directory.
 */

Found in path(s):
* /opt/cola/permits/1110631590_1606842066.7/0/regenerator-runtime-0-10-5-4-tgz/package/runtime.js

1.631 camelcase 5.3.1
1.631.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.632 escape-html 1.0.3 1.0.3

1.632.1 Available under license :
(The MIT License)

Copyright (c) 2012-2013 TJ Holowaychuk
Copyright (c) 2015 Andreas Lubbe
Copyright (c) 2015 Tiancheng "Timothy" Gu

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.633 parse-filepath 1.0.2

1.633.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.634 should-util 1.0.1 1.0.1

1.634.1 Available under license:

MIT License

Copyright (c) 2016 Should.js assertion library

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.635 mime-lookup 0.0.2 0.0.2
1.635.1 Available under license:

Original work Copyright (c) 2010 Benjamin Thomas, Robert Kieffer
Modified work Copyright 2015 Jayson Harshbarger

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.636 regenerator-runtime 0.11.1 0.11.1
1.636.1 Available under license:

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Apache HttpCore NIO  
Copyright 2005-2018 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (http://www.apache.org/).

Apache License  
Version 2.0, January 2004  
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.638 swagger-jaxrs2 2.0.7
1.638.1 Available under license :
Copyright 2018 SmartBear Software

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at [apache.org/licenses/LICENSE-2.0][http://www.apache.org/licenses/LICENSE-2.0]

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.639 balanced-match 1.0.0 1.0.0
1.639.1 Available under license :
(MIT)

Copyright (c) 2013 Julian Gruber &lt;julian@juliangruber.com&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.640 metadata-extractor 2.13.0
1.640.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2002-2019 Drew Noakes and contributors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.641 rimraf 2.4.5 2.4.5
1.641.1 Available under license :

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.642 shebang-command 1.2.0

1.642.1 Available under license:
The MIT License (MIT)

Copyright (c) Kevin Martensson <kevinmartensson@gmail.com> (github.com/kevva)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.643 jctools-core 2.0.1

1.644 metrics-json 4.0.2

1.645 lz4-java 1.7.1

1.645.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.646 jakarta-ws-rs-api 2.1.6

1.646.1 Available under license:

# Notices for the Jakarta RESTful Web Services Project

This content is produced and maintained by the **Jakarta RESTful Web Services** project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaxrs

## Trademarks

**Jakarta RESTful Web Services** is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaxrs-api

## Third-party Content

This project leverages the following third party content.
## Cryptography

Content may contain encryption software. The country in which you are currently
may have restrictions on the import, possession, and use, and/or re-export to
another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the import,
possession, or use, and re-export of encryption software, to see if this is
permitted.

# Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE
PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION
OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
   Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from
and are Distributed by that particular Contributor. A Contribution
"originates" from a Contributor if it was added to the Program by
such Contributor itself or anyone acting on such Contributor's behalf.
Contributions do not include changes or additions to the Program that
are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which
are necessarily infringed by the use or sale of its Contribution alone
or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this
Agreement.

"Recipient" means anyone who receives the Program under this Agreement
or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other
form, that is based on (or derived from) the Program and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that
results from an addition to, deletion from, or modification of the
contents of the Program, including, for purposes of clarity any new file
in Source Code form that contains any contents of the Program. Modified
Works shall not include works that contain only declarations,
interfaces, types, classes, structures, or files of the Program solely
in each case in order to link to, bind by name, or subclass the Program
or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available
in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making
modifications, including but not limited to software source code,
documentation source, and configuration files.

"Secondary License" means either the GNU General Public License,
Version 2.0, or any later versions of that license, including any
exceptions or additional permissions as identified by the initial
Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby
grants Recipient a non-exclusive, worldwide, royalty-free copyright
license to reproduce, prepare Derivative Works of, publicly display,
publicly perform, Distribute and sublicense the Contribution of such
Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby
grants Recipient a non-exclusive, worldwide, royalty-free patent
license under Licensed Patents to make, use, sell, offer to sell,
import and otherwise transfer the Contribution of such Contributor,
if any, in Source Code or other form. This patent license shall
apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness
ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged
intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of
the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s),}
version(s), and exceptions or additional permissions here)."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis
or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other
recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.
3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does. Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program ‘Gnomovision’ (which makes passes at compilers) written by
James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

---

## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is
making a combined work based on this library. Thus, the terms and
conditions of the GNU General Public License version 2 cover the whole
combination.

As a special exception, the copyright holders of this library give you
permission to link this library with independent modules to produce an
executable, regardless of the license terms of these independent
modules, and to copy and distribute the resulting executable under
terms of your choice, provided that you also meet, for each linked
independent module, the terms and conditions of the license of that
module. An independent module is a module which is not derived from or
based on this library. If you modify this library, you may extend this
exception to your version of the library, but you are not obligated to
do so. If you do not wish to do so, delete this exception statement
from your version.

1.647 lombok 1.18.20

1.647.1 Available under license :

Copyright (C) 2009-2021 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

==============================================================================

Licenses for included components:

org.ow2.asm:asm
org.ow2.asm:asm-analysis
org.ow2.asm:asm-commons
org.ow2.asm:asm-tree
org.ow2.asm:asm-util

ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
rzwitserloot/com.zwitserloot.cmdreader

Copyright 2010 Reinier Zwitserloot.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

rzwitserloot/lombok.patcher

Copyright (C) 2009-2021 The Project Lombok Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.648 dbf-reader 0.3.0
1.648.1 Available under license:

MIT License

Copyright (c) 2016 Yellow Feather Ltd

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.649 reselect 3.0.1 3.0.1
1.649.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2016 Reselect Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.650.1 Available under license :
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.651 jackson-datatype-joda 2.4.2
1.651.1 Available under license :
This copy of Jackson JSON processor streaming parser/generator is licensed under the Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.652 process-nextick-args 1.0.7 1.0.7
1.652.1 Available under license:

# Copyright (c) 2015 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

1.653 jsreport-xlsx 2.5.0

1.653.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.654 busboy 0.2.14

1.654.1 Available under license:

Copyright Brian White. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.655 bytes 2.3.0

1.655.1 Available under license:

(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Jed Watson <jed.watson@me.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.656 call-me-maybe 1.0.1 1.0.1

1.656.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Eric McCarthy

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.657 google-http-client 1.20.0

1.658 mime-types 2.1.19 2.1.19

1.658.1 Available under license:
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ‘AS IS’, WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.659 d-conf 0.24.0-2
1.659.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations
below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no
charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a
table of data to be supplied by an application program that uses
the facility, other than as an argument passed when the facility
is invoked, then you must make a good faith effort to ensure that,
in the event an application does not supply such function or
table, the facility still operates, and performs whatever part of
its purpose remains meaningful.

(For example, a function in a library to compute square roots has
a purpose that is entirely well-defined independent of the
application. Therefore, Subsection 2d requires that any
application-supplied function or table used by this function must
be optional: if the application does not supply it, the square
root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.
For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.
10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
1.660 semver 5.6.0

1.660.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.661 ultron 1.0.2 1.0.2

1.661.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Hive Metastore
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.663 get-intrinsic 1.1.1

1.663.1 Available under license :

MIT License

Copyright (c) 2020 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
**1.664 winser-with-api 1.0.1**

**1.664.1 Available under license:**
No license file was found, but licenses were detected in source scan.

# WinSer

Run node.js applications as windows services using [nssm](http://nssm.cc).

## Installation

$ npm install winser

## Command line arguments

- `-v`, `--version` show winser version and exit
- `-a`, `--autostart` start the application immediately after installation
- `-i`, `--install` install the node application as a windows service
- `-r`, `--remove` remove the windows service for the node application
- `-x`, `--stop` stop the service before uninstalling
- `-s`, `--silent` suppress any information in the console
- `-c`, `--confirmation` ask for confirmation before installing/uninstalling
- `-p`, `--path <ARG1>` path to the node application you want to install as a service (current path by default)
- `-n`, `--name <ARG1>` name for service (default: name from package.json)
- `-d`, `--description <ARG1>` service description (default: description from package.json)
- `--displayname <ARG1>` display name for service
- `--startcmd <ARG1>` use this command for start service (default: scripts.start from package.json)
- `--startuptype <ARG1>` set service startup type on boot (auto|manual|delayed|disabled) ("auto" by default)
- `--env <ARG1>` propogate environment variable (multiple)
- `--startwithnpm` use "npm start" as a startcmd
- `--set <ARG1>` call nssm "set" command with arguments (multiple)

## Method 1

I really like this method, in the package.json:

```js
```scripts``": {
    "postinstall": "winser -i -s -c",
    "preuninstall": "winser -r -x -s",
}
````

Then, in order to install a node application in lets say a server I will do this:

```bash
npm install git://github.com/myprivate/repository/url.git
```
The arguments in the **postinstall** script means:

- i install
- s silent, don't display any information
- c ask for confirmation. This is very helpful because during development you don't want to install/uninstall the package as a windows service but you will often run "npm install" in the folder, then you can cancel with an 'n'.

The arguments in the **preuninstall** script means:

- x stop the service before uninstalling
- r remove the service
- s silent, don't display any information

## Method 2

Add these two scripts to your package.json:

```js
"scripts": {
  "install-windows-service": "winser -i",
  "uninstall-windows-service": "winser -r"
}
```

Then you can install your service as:

```bash
npm run-script install-windows-service
```

## How it works

When you install your node.js program as a windows service, your program is registered using nssm.exe (which is inside the module folder). Once you start the service nssm.exe is run and nssm.exe will execute start command.

Start command may be:
- npm "start" action from package.json.
- "node <main>“, where <main> is a main section in package.json.
- command provided by "--startcmd <command>" option.

Remember that the default npm action for "start" is "node server.js".

The name of the service will be the same from your package.json "name" setting.

## Credits
This project is heavily inspired in

- [Node.js on windows by Tatham Oddie](http://blog.tatham.oddie.com.au/2011/03/16/node-js-on-windows/)

And it uses:

- [nssm](http://nssm.cc)

## License

(The MIT License)

Copyright (c) 2011 Jose Romaniello &lt;jfromaniello@gmail.com&gt;

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):

* /opt/cola/permits/1226013673_1636487527.37/0/winser-with-api-1-0-1-tgz/package/readme.md

1.665 mime-types 2.1.30 2.1.30

1.665.1 Available under license :

(The MIT License)

Copyright (c) 2014 Jonathan Ong &lt;me@jongleberry.com&gt;
Copyright (c) 2015 Douglas Christopher Wilson &lt;doug@somethingdoug.com&gt;

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.666 mime-pull 1.9.3

1.667 objenesis 2.1
1.667.1 Available under license :

// ------------------------------------------------------------------
// NOTICE file corresponding to the section 4d of The Apache License,
// Version 2.0, in this case for Objenesis
// ------------------------------------------------------------------

Objenesis
Copyright 2006-2013 Joe Walnes, Henri Tremblay, Leonardo Mesquita

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, 
work stoppage, computer failure or malfunction, or any and all 
other commercial damages or losses), even if such Contributor 
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing 
the Work or Derivative Works thereof, You may choose to offer, 
and charge a fee for, acceptance of support, warranty, indemnity, 
or other liability obligations and/or rights consistent with this 
License. However, in accepting such obligations, You may act only 
on Your own behalf and on Your sole responsibility, not on behalf 
of any other Contributor, and only if You agree to indemnify, 
defend, and hold each Contributor harmless for any liability 
incurred by, or claims asserted against, such Contributor by reason 
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following 
boilerplate notice, with the fields enclosed by brackets "[]" 
replaced with your own identifying information. (Don't include 
the brackets!) The text should be enclosed in the appropriate 
comment syntax for the file format. We also recommend that a 
file or class name and description of purpose be included on the 
same "printed page" as the copyright notice for easier 
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. 
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software 
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.
1.669

de.huxhorn.lilith.logback.appender.multiplex-classic 0.9.44

1.669.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:
a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

  0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

  1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.
4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the
Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

==================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
==================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message
The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

*javax.servlet:javax.servlet-api
*javax.annotation:javax.annotation-api
*javax.transaction:javax.transaction-api
*javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

MortBay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.671 aws-java-sdk-core 1.11.375
1.671.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * <p/>
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with
 * the License. A copy of the License is located at
 * <p/>
 * http://aws.amazon.com/apache2.0
 * <p/>
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions
 * and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/partitions/model/Partitions.java
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * <p/>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p/>
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/log/InternalLogApi.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/EnvironmentVariableCredentialsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/http/IdleConnectionReaper.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/DefaultAWSCredentialsProviderChain.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/InstanceProfileCredentialsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/AWSCredentialsProviderChain.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/CRC32MismatchException.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/CustomBackoffStrategy.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/DynamoDBBackoffStrategy.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/EC2MetadataClient.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/ClasspathPropertiesFileCredentialsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/HandlerContextAware.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/AwsHostNameUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/AwsHostNameUtils.java

No license file was found, but licenses were detected in source scan.

*/

* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements.  See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership.  The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
*
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/BasicNameValuePair.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/URLEncodedUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/NameValuePair.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/transform/LegacyErrorUnmarshaller.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/transform/Unmarshaller.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/SigningAlgorithm.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/AmazonClientException.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/transform/SimpleTypeUnmarshallers.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/XmlUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/transform/VoidUnmarshaller.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/SignatureVersion.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/http/HttpMethodName.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/AWSCredentials.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not
 * use this file except in compliance with the License. A copy of the License is
 * located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
You may not use this file except in compliance with the License.
A copy of the License is located at

http://aws.amazon.com/apache2.0

or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/http/RepeatableInputStreamRequestEntity.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/config/InternalConfig.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/event/ProgressEventType.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/http/conn/SdkPlainSocketFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/policy/conditions/ConditionFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/AWSServiceMetrics.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/ApacheHttpClientConfig.java
Open Source Used In DNAC Platform Antelope 2205

* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/metrics/ServiceMetricCollector.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/metrics/SimpleServiceMetricType.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/http/impl/client/SdkHttpRequestRetryHandler.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/util/RuntimeHttpUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/retry/RetryUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/waiters/AcceptorPathMatcher.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/policy/conditions/ArnCondition.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/policy/package-info.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/BasicAWSCredentials.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/NoOpSigner.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/event/ProgressListenerChain.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/transform/JsonUnmarshallerContextImpl.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/handlers/RequestHandler2.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/metrics/SimpleThroughputMetricType.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/event/ProgressEvent.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/internal/IdentityEndpointBuilder.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/waiters/WaiterState.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/AbstractAWSSigner.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/handlers/HandlerChainFactory.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/QueryStringSigner.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/Request.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/waiters/WaiterAcceptor.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/waiters/WaiterExecution.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/ProxyAuthenticationMethod.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/metrics/ByteThroughputHelper.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/AWS4UnsignedPayloadSigner.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/ServiceNameFactory.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/internal/SdkPredicate.java
* /opt/cola/permits/1135892258_1613626888.74/jar/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/waiters/HttpFailureStatusAcceptor.java
No license file was found, but licenses were detected in source scan.

/* Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.

* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/
Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/profile/path/AwsDirectoryBasePathProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/profile/path/AwsProfileFileLocationProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/internal/NullAsEmptyBodyProtocolRequestMarshaller.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/AwsProfileFileLocationProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/AwsProfileFileLocationProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/AwsProfileFileLocationProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/AwsProfileFileLocationProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/AwsProfileFileLocationProvider.java
Open Source Used In DNAC Platform Antelope 2212

* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/protocol/json/internal/SimpleTypePathMarshallers.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/transform/MapUnmarshaller.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/protocol/json/internal/ValueToStringConverters.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/client/builder/AwsSyncClientBuilder.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/retry/v2/MaxNumberOfRetriesCondition.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/RequestConfig.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/protocol/json/internal/HeaderMarshallers.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/protocol/Protocol.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/transform/ListUnmarshaller.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/regions/AwsRegionProviderChain.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/retry/v2/AndRetryCondition.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/profile/path/AwsProfileFileLocationProviderChain.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/AWSStaticCredentialsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/client/ClientExecutionParams.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/protocol/json/JsonErrorShapeMetadata.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/http/SdkHttpMetadata.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/auth/profile/internal/ProfileStaticCredentialsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/retry/V2CompatibleBackoffStrategyAdapter.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1.jar/com/amazonaws/retry/V2CompatibleBackoffStrategy.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/MarshallLocation.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/transform/PathMarshallers.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/retry/v2/OrRetryCondition.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/AwsProfileRegionProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/client/builder/AwsClientBuilder.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/presign/PresignerParams.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/JodaTime.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/StringMapBuilder.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/JodaTime.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/StringMapBuilder.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-2018 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
* http://aws.amazon.com/apache2.0
*
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/SdkSSLContext.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/annotation/package-info.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/util/UnreliableFilterInputStream.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/event/RequestProgressInputStream.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/profile/package-info.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/profile/progress/ProgressEventFilter.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/profile/progress/ProgressEventFilter.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/profile/internal/securitytoken/ProfileCredentialsServiceLoader.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/profile/internal/securitytoken/ProfileCredentialsServiceLoader.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/profile/ProfileCredentialsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/profile/ProfileCredentialsProvider.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights
 * Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/event/ProgressInputStream.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/internal/AWS4SignerRequestParams.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/event/request/ProgressSupport.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014-2018 Amazon Technologies, Inc.
 */
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* 
* http://aws.amazon.com/apache2.0
* 
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/http/conn/ssl/TLSProtocol.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/http/conn/ssl/SdkTLSSocketFactory.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* 
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
* for applicable license terms and NOTICE.txt for applicable notices.
* 
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* 
* http://aws.amazon.com/apache2.0
* 
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/Classes.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/DateUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/BinaryUtils.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/PolicyUtils.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2012-2018 Amazon Technologies, Inc.
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
* for applicable license terms and NOTICE.txt for applicable notices.
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* http://aws.amazon.com/apache2.0
*
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/Md5Utils.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2016-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/InMemoryRegionsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/partitions/PartitionMetadataProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/LegacyRegionXmlLoadUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/partitions/model/CredentialScope.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/partitions/model/Region.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/models/RegionMetadataFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredIonFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/http/apache/SdkProxyRoutePlanner.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/LegacyRegionXmlMetadataBuilder.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/partitions/model/Endpoint.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkIonGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/transform/SimpleTypeCborUnmarshallers.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/models/RegionMetadataImpl.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/models/AbstractRegionMetadataProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/partitions/model/Service.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/partitions/model/PartitionRegionImpl.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/models/RegionImpl.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/regions/RegionMetadataProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

/**
 * Interface for providing AWS region information. Implementations are free to use any strategy for
 * providing region information.
 */

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/regions/AwsRegionProvider.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2018 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/jmx/spi/JmxInfoProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
*/
*
* Copyright (c) 2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*
*/

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkCborGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/protocol/json/SdkStructuredCborFactory.java
jar/com/amazonaws/protocol/json/SdkStructuredPlainJsonFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/protocol/json/StructuredJsonGenerator.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/protocol/json/SdkJsonProtocolFactory.java
No license file was found, but licenses were detected in source scan.

/
* Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* *
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/internal/config/SignerConfigJsonHelper.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/internal/config/SignerConfig.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/internal/config/JsonIndex.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/internal/config/Builder.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/util/ImmutableMapParameter.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/Protocol.java
No license file was found, but licenses were detected in source scan.

/
* Copyright 2012-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
* with
* the License. A copy of the License is located at
* *
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR
* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions
* and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/annotation/GuardedBy.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/annotation/Immutable.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/annotation/NotThreadSafe.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/annotation/SdkProtectedApi.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/annotation/SdkInternalApi.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/annotation/ThreadSafe.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/annotation/SdkTestInternalApi.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/annotation/Beta.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
jar/com/amazonaws/handlers/AsyncHandler.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
  *
  * http://aws.amazon.com/apache2.0
  *
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/auth/RequestSigner.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/ImmutableRequest.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/auth/SignerAsRequestSigner.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License"). You may not
* use this file except in compliance with the License. A copy of the License is
* located at
  *
  * http://aws.amazon.com/apache2.0
  *
* or in the "license" file accompanying this file. This file is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/transform/SimpleTypeIonUnmarshallers.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/protocol/json/IonFactory.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/protocol/json/IonParser.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2016. Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
Licensed under the Apache License, Version 2.0 (the "License").
You may not use this file except in compliance with the License.
A copy of the License is located at

http://aws.amazon.com/apache2.0

or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
permissions and limitations under the License.

/*

Found in path(s):
*/

*/

No license file was found, but licenses were detected in source scan.

/*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is
* distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either
* express or implied. See the License for the specific language
* governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/CredentialUtils.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
* for applicable license terms and NOTICE.txt for applicable notices.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/internal/ReleasableInputStream.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/internal/ResettableInputStream.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at

---

Open Source Used In DNAC Platform Antelope 2229
No license file was found, but licenses were detected in source scan.
/*
 * Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/EC2CredentialsFetcher.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/EC2ContainerCredentialsProviderWrapper.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/retry/internal/CredentialsEndpointRetryParameters.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/ContainerCredentialsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/retry/internal/CredentialsEndpointRetryPolicy.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/ContainerCredentialsRetryPolicy.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/ConnectionUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/CredentialsEndpointProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/ContainerCredentialsRetryPolicy.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/internal/EC2CredentialsUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/auth/ContainerCredentialsProvider.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-jar/com/amazonaws/retry/internal/CredentialsEndpointRetryPolicy.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 * or in the "license" file accompanying this file. This file is distributed
 */
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/internal/MetricAware.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/jmx/SdkMBeanRegistrySupport.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/jmx/MBeans.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/Base32.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/internal/MetricsInputStream.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/auth/RegionAwareSigner.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/auth/ServiceAwareSigner.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/EC2MetadataUtils.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/HttpClientWrappingInputStream.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/AbstractBase32Codec.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/Codec.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/Base64.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/Base64Codec.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/SdkRuntime.java
* /opt/cola/permits/1135892258_1613626888.74/0/aws-java-sdk-core-1-11-375-sources-1-
  jar/com/amazonaws/util/IOUtils.java
1.672 registry-url 5.1.0

1.672.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.673 ajv-keywords 1.5.1
1.673.1 Available under license :
   The MIT License (MIT)

Copyright (c) 2016 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.674 fast-deep-equal 3.1.3
1.674.1 Available under license :
   MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
1.675 babel-helper-builder-binary-assignment-operator-visitor 6.24.1

1.675.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
   following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
   from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTicular PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.676 glob 6.0.4 6.0.4
1.676.1 Available under license :
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.677 hadoop-distcp 3.2.2
1.677.1 Available under license :

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of
org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of
AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java),
which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of
Gson 2.2.4,
which has the following notices:

    The Netty Project
    =================

Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

-------------------------------------------------------------------------------

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression
and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java
This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google’s data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava
This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.alto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack
This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of
Commons Codec 1.4,
which has the following notices:
* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab.Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)
  ================================================================
The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of
Commons Lang 2.6,
which has the following notices:
* This product includes software from the Spring Framework, under the Apache License 2.0 (see:
  StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of
Apache Log4j 1.2.17,
which has the following notices:
* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of
"Java Concurrency in Practice" book annotations 1.0,
which has the following notices:
* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative
  Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,
which has the following notices:

* ==============================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
* ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit.org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbitjavax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbitjavax.mail.glassfish
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception. https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions. http://openjdk.java.net/legal/gplv2+ce.html

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay
The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common http://hadoop.apache.org/ (Apache 2.0 license)
This library containd statically linked libstdc++. This inclusion is allowed by "GCC RUNtime Library Exception" http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html

== Contributors ==
* Tatu Saloranta
* Providing benchmark suite
* Alec Wysoker
* Performance and memory usage improvement

The binary distribution of this product bundles binaries of Xerces2 Java Parser 2.9.1, which has the following notices:

* =========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xerces Java distribution. ==
=========================================================================

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

The binary distribution of this product bundles binaries of Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

  This product includes software developed at
  The Apache Software Foundation (http://www.apache.org/).

  --
  This product incorporates portions of the 'Hadoop' project

  Copyright 2007-2009 The Apache Software Foundation

  Licensed under the Apache License v2.0
  --
  Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
  It is licensed Creative Commons Attribution 3.0.
  See https://creativecommons.org/licenses/by/3.0/us/
  We changed the logo by stripping the colored background, inverting
  it and then rotating it some.

  Later we found that vectorfree.com image is not properly licensed.
  The original is owned by vectorportal.com. The original was
  relicensed so we could use it as Creative Commons Attribution 3.0.
  The license is bundled with the download available here:
  --
  This product includes portions of the Bootstrap project v3.0.0

  Copyright 2013 Twitter, Inc.

  Licensed under the Apache License v2.0

  This product uses the Glyphicons Halflings icon set.

  http://glyphicons.com/

  Copyright Jan Kovak

  Licensed under the Apache License v2.0 as a part of the Bootstrap project.

  --
  This product includes portions of the Guava project v14, specifically
  'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

  Copyright (C) 2007 The Guava Authors

  Licensed under the Apache License, Version 2.0
The binary distribution of this product bundles binaries of Phoenix Core 4.7.0, which has the following notices:

Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5 (https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 2008-2015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 2007-2008 Robert Niederreiter, Jens Klein
# Copyright 2004-2005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of Plexus Cipher: encryption/decryption Component 1.4, which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of software.amazon.ion:ion-java 1.0.1, which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of joda-time:joda-time:2.9.9 which has the following notices:
* = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

This product includes software developed by
Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of Ehcache 3.3.1, which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of snakeyaml (https://bitbucket.org/asomov/snakeyaml), which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of swagger-annotations (https://github.com/swagger-api/swagger-core), which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of metrics-core 3.2.4 which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.
For the org.apache.hadoop.util.bloom.* classes:

/**
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL
 * nor the names of its contributors may be used to endorse or
 * promote products derived from this software without specific prior
 * written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
 * ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor
the names of its contributors may be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/
/**
 *
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 *
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 *
 * Abstract: The main routine
 *
 */

For src/main/native/src/org/apache/hadoop/io/compress/lz4/\{lz4.h,lz4.c,lz4hc.h,lz4hc.c\},

/*
LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at:
- LZ4 source repository: http://code.google.com/p/lz4/
- LZ4 public forum: https://groups.google.com/forum/#!forum/lz4c

/*

For hadoop-common-project/hadoop-common/src/main/native/gtest

---------------------------------------------------------------------

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1

---------------------------------------------------------------------
(GO license)
This is a work derived from Russ Cox’s RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni (https://github.com/fusesource/leveldbjni), which is available under the following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of FuseSource Corp. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from
the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

============

The binary distribution of this product bundles binaries of leveldb
(http://code.google.com/p/leveldb/), which is available under the following
license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
THE binary distribution of this product bundles binaries of snappy 
(http://code.google.com/p/snappy/), which is available under the following 
license:

Copyright 2011, Google Inc. 
All rights reserved. 

Redistribution and use in source and binary forms, with or without 
modification, are permitted provided that the following conditions are 
met:

* Redistributions of source code must retain the above copyright 
notice, this list of conditions and the following disclaimer. 
* Redistributions in binary form must reproduce the above 
copyright notice, this list of conditions and the following disclaimer 
in the documentation and/or other materials provided with the 
distribution. 
* Neither the name of Google Inc. nor the names of its 
contributors may be used to endorse or promote products derived from 
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT 
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR 
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT 
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, 
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT 
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, 
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY 
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT 
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE 
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

--------------------------------------------------------------------------------

Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

--------------------------------------------------------------------------------

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

---------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
em-helpers v0.5.13
em-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

--------------------------------------------------------------------------------
Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this
file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the
License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
express or implied. See the License for the specific language governing permissions and
limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js

--------------------------------------------------------------------------------
The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software
and associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js

The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery

Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

-------------------------------------------------------------------------------------------------
MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

For:
oj! Algorithms - version 43.0

================================================================================================================================


This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as
documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js
--------------------------------------------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js
--------------------------------------------------------------------------------

D3 is available under a 3-clause BSD license. For details, see:
The binary distribution of this product bundles these dependencies under the following license:
HSQLDB Database 2.3.4

(HSQL License)
"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ""AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.”

The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

---------------------------------------------------------------
(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered
Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct
or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made,
and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients.
of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.
4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.
7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER
NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE,
SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF
COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY
PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY
CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF
GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER
COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE
POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO
LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO
THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT
ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO
THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in
48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as
that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer
software documentation as such terms are used in 48C.F.R.12.212 (Sept.
227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software
with only those rights set forth herein. This U.S. Government Rights clause is
in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision
that addresses Government rights in computer software under this
License.

9. MISCELLANEOUS.
This License represents the complete agreement
concerning subject matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent necessary to
make it enforceable. This License shall be governed by the law of the
jurisdiction specified in a notice contained within the Original Software
(except to the extent applicable law, if any, provides otherwise), excluding
such jurisdictions conflict-of-law provisions. Any litigation relating to this
License shall be subject to the jurisdiction of the courts located in the
jurisdiction and venue specified in a notice contained within the Original
Software, with the losing party responsible for costs, including, without
limitation, court costs and reasonable attorneys fees and expenses. The
application of the United Nations Convention on Contracts for the International
Sale of Goods is expressly excluded. Any law or regulation which provides that
the language of a contract shall be construed against the drafter shall not
apply to this License. You agree that You alone are responsible for compliance
with the United States export administration regulations (and the export control
laws and regulation of any other countries) when You use, distribute or
otherwise make available any Covered Software.
10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
grizzly-http 2.2.21
grizzly-http-server 2.2.21
grizzly-http-servlet 2.2.21
grizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is
grounded: (1) for code that You delete from the Original Software, or (2) for
infringements caused by: (i) the modification of the Original Software, or (ii)
the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and
subject to third party intellectual property claims, each Contributor hereby
grants You a world-wide, royalty-free, non-exclusive license:
(a) under
intellectual property rights (other than patent or trademark) Licensable by
Contributor to use, reproduce, modify, display, perform, sublicense and
distribute the Modifications created by such Contributor (or portions thereof),
either on an unmodified basis, with other Modifications, as Covered Software
and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the
making, using, or selling of Modifications made by that Contributor either alone
and/or in combination with its Contributor Version (or portions of such
combination), to make, use, sell, offer for sale, have made, and/or otherwise
dispose of: (1) Modifications made by that Contributor (or portions thereof);
and (2) the combination of Modifications made by that Contributor with its
Contributor Version (or portions of such combination).
(c) The licenses granted
in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications available to a third
party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2)
for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with
other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or
otherwise make available in Executable form must also be made available in
Source Code form and that Source Code form must be distributed only under the
terms of this License. You must include a copy of this License with every copy
of the Source Code form of the Covered Software You distribute or otherwise make
available. You must inform recipients of any such Covered Software in Executable
form as to how they can obtain such Covered Software in Source Code form in a
reasonable manner on or through a medium customarily used for software
exchange.
3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other
than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the
Covered Software available under the terms of the version of the License under
which You originally received the Covered Software. If the Initial Developer
includes a notice in the Original Software prohibiting it from being distributed
or otherwise made available under any subsequent version of the License, You
must distribute and make the Covered Software available under the terms of the
version of the License under which You originally received the Covered Software.
Otherwise, You may also choose to use, distribute or otherwise make the Covered
Software available under the terms of any subsequent version of the License
published by the license steward.

4.3. Modified Versions.
When You are an
Initial Developer and You want to create a new license for Your Original
Software, You may create and use a modified version of this License if You: (a)
rename the license and remove any references to the name of the license steward
(except to note that the license differs from this License); and (b) otherwise
make it clear that the license contains terms which differ from this
License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS
LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE
IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR
NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED
SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY
RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE
COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF
WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED
SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will
terminate automatically if You fail to comply with terms herein and fail to cure
such breach within 30 days of becoming aware of the breach. Provisions which, by
their nature, must remain in effect beyond the termination of this License shall
survive.

6.2. If You assert a patent infringement claim (excluding declaratory
judgment actions) against Initial Developer or a Contributor (the Initial
Developer or Contributor against whom You assert such claim is referred to as
Participant) alleging that the Participant Software (meaning the
Contributor Version where the Participant is a Contributor or the Original
Software where the Participant is the Initial Developer) directly or indirectly
infringes any patent, then any and all rights granted directly or indirectly to
You by such Participant, the Initial Developer (if the Initial Developer is not
the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this
License shall, upon 60 days notice from Participant terminate prospectively and
automatically at the expiration of such 60 day notice period, unless if within
such 60 day period You withdraw Your claim with respect to the Participant
Software against such Participant either unilaterally or pursuant to a written
agreement with Participant.

6.3. If You assert a patent infringement claim
against Participant alleging that the Participant Software directly or
indirectly infringes any patent where such claim is resolved (such as by license
or settlement) prior to the initiation of patent infringement litigation, then
the reasonable value of the licenses granted by such Participant under Sections
2.1 or 2.2 shall be taken into account in determining the amount or value of any
payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2
above, all end user licenses that have been validly granted by You or any
distributor hereunder prior to termination (excluding licenses granted to You by
any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT
(INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL
DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR
ANY
SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT,
SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING,
WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER
FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN
IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS
LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL
INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW
PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR
LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND
LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered
Software is a commercial item, as that term is defined in 48 C.F.R. 2.101
(Oct. 1995), consisting of commercial computer software (as that term is
defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software
documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995).
Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4
(June 1995), all U.S. Government End Users acquire Covered Software with only
those rights set forth herein. This U.S. Government Rights clause is in lieu of,
and supersedes, any other FAR, DFAR, or other clause or provision that addresses
Government rights in computer software under this License.
9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04
--------------------------------------------------------------------------------
By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is
The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
   where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient’s responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program. Contributors may not remove or alter any copyright notices contained within the Program.
Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.
6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of
any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.
(2-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
to create and reproduce Derivative Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other
jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats.

All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.
The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author").
Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the
The binary distribution of this product bundles these dependencies under the following license:

jamon-runtime 2.4.1

(MPL 2.0)

Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.
1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions
2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).
2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not
Incompatible With Secondary Licenses, this License permits You to additionally
distribute such Covered Software under the terms of such Secondary License(s),
so that the recipient of the Larger Work may, at their option, further
distribute the Covered Software under the terms of either this License or such
Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including
copyright notices, patent notices, disclaimers of warranty, or limitations of
liability) contained within the Source Code Form of the Covered Software, except
that You may alter any license notices to the extent required to remedy known
factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity
or liability obligations to one or more recipients of Covered Software. However,
You may do so only on Your own behalf, and not on behalf of any Contributor. You
must make it absolutely clear that any such warranty, support, indemnity, or
liability obligation is offered by You alone, and You hereby agree to indemnify
every Contributor for any liability incurred by such Contributor as a result of
warranty, support, indemnity or liability terms You offer. You may include
additional disclaimers of warranty and limitations of liability specific to any
jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with
respect to some or all of the Covered Software due to statute, judicial order,
or regulation then You must: (a) comply with the terms of this License to the
maximum extent possible; and (b) describe the limitations and the code they
affect. Such description must be placed in a text file included with all
distributions of the Covered Software under this License. Except to the extent
prohibited by statute or regulation, such description must be sufficiently
detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You
fail to comply with any of its terms. However, if You become compliant, then the
rights granted under this License from a particular Contributor are reinstated
(a) provisionally, unless and until such Contributor explicitly and finally
terminates Your grants, and (b) on an ongoing basis, if such Contributor fails
to notify You of the non-compliance by some reasonable means prior to 60 days
after You have come back into compliance. Moreover, Your grants from a
particular Contributor are reinstated on an ongoing basis if such Contributor
notifies You of the non-compliance by some reasonable means, this is the first
time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without
reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If you choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then
You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

/*--
   Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
   All rights reserved.

   Redistribution and use in source and binary forms, with or without
   modification, are permitted provided that the following conditions
   are met:

   1. Redistributions of source code must retain the above copyright
      notice, this list of conditions, and the following disclaimer.

   2. Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions, and the disclaimer that follows
      these conditions in the documentation and/or other materials
      provided with the distribution.

   3. The name "JDOM" must not be used to endorse or promote products
      derived from this software without prior written permission.  For
      written permission, please contact <request_AT_jdom_DOT_org>.

   4. Products derived from this software may not be called "JDOM", nor
      may "JDOM" appear in their name, without prior written permission
      from the JDOM Project Management <request_AT_jdom_DOT_org>.

   In addition, we request (but do not require) that you include in the
   end-user documentation provided with the redistribution and/or in the
   software itself an acknowledgement equivalent to the following:
   "This product includes software developed by the
   JDOM Project (http://www.jdom.org/)."

   Alternatively, the acknowledgment may be graphical using the logos

   THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS
1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work
of the same nature; a dramatic or dramatico-musical work; a
choreographic work or entertainment in dumb show; a musical
composition with or without words; a cinematographic work to which are
assimilated works expressed by a process analogous to cinematography;
a work of drawing, painting, architecture, sculpture, engraving or
lithography; a photographic work to which are assimilated works
expressed by a process analogous to photography; a work of applied
art; an illustration, map, plan, sketch or three-dimensional work
relative to geography, topography, architecture or science; a
performance; a broadcast; a phonogram; a compilation of data to the
extent it is protected as a copyrightable work; or a work performed by
a variety or circus performer to the extent it is not otherwise
considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this
License who has not previously violated the terms of this License with
respect to the Work, or who has received express permission from the
Licensor to exercise rights under this License despite a previous
violation.

h. "Publicly Perform" means to perform public recitations of the Work and
to communicate to the public those public recitations, by any means or
process, including by wire or wireless means or public digital
performances; to make available to the public Works in such a way that
members of the public may access these Works from a place and at a
place individually chosen by them; to perform the Work to the public
by any means or process and the communication to the public of the
performances of the Work, including by public digital performance; to
broadcast and rebroadcast the Work by any means including signs,
sounds or images.

i. "Reproduce" means to make copies of the Work by any means including
without limitation by sound or visual recordings and the right of
fixation and reproducing fixations of the Work, including storage of a
protected performance or phonogram in digital form or other electronic
medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce,
limit, or restrict any uses free from copyright or rights arising from
limitations or exceptions that are provided for in connection with the
copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License,
Licensor hereby grants You a worldwide, royalty-free, non-exclusive,
perpetual (for the duration of the applicable copyright) license to
exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more
   Collections, and to Reproduce the Work as incorporated in the
   Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation,
including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under
the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to
make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author’s honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the
Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.


Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.678 netty-handler 4.1.42.Final

1.678.1 Available under license:

No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/META-INF/native-image/io.netty/handler/native-image.properties

No license file was found, but licenses were detected in source scan.
Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
jar/io/netty/handler/ssl/JdkDefaultApplicationProtocolNegotiator.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslEngine.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslSessionContext.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/CipherSuiteFilter.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslX509Certificate.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/JdkSslContext.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSsl.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/SslContext.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/ApplicationProtocolConfig.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/SniHandler.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2017 The Netty Project
* *
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/Conscrypt.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/ConscryptAlpnSslEngine.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/SslCompletionEvent.java
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/SslHandshakeCompletionEvent.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/SslHandshakeCompletionEvent.java
No license file was found, but licenses were detected in source scan.
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/ApplicationProtocolNegotiationHandler.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/ClientAuth.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/SslContextBuilder.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/ApplicationProtocolAccessor.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/ApplicationProtocolNames.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/OpenSslSessionTicketKey.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2018 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/OpenSslTLsv13X509ExtendedTrustManager.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/SignatureAlgorithmConverter.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/OpenSslX509KeyManagerFactory.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/SignatureAlgorithmConverter.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1.jar/io/netty/handler/ssl/OpenSslX509KeyManagerFactory.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
jar/io/netty/handler/ssl/OpenSslCachingX509KeyManagerFactory.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslX509TrustManagerWrapper.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/DefaultOpenSslKeyMaterial.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslHandlerSession.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslSession.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslKeyMaterialProvider.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/ExtendedOpenSslSession.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslPrivateKey.java

No license file was found, but licenses were detected in source scan.

/
* Copyright 2016 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License, version
* 2.0 (the "License"); you may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/flow/FlowControlHandler.java

No license file was found, but licenses were detected in source scan.

/
* Copyright 2012 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/logging/LogLevel.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/logging/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/stream/ChunkedWriteHandler.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/timeout/WriteTimeoutException.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/timeout/ReadTimeoutException.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/timeout/IdleState.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/ssl/util/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/stream/ChunkedStream.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/timeout/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/ssl/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/ssl/SSlHandler.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/timeout/WriteTimeoutHandler.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/stream/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/timeout/ReadTimeoutHandler.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/timeout/IdleStateEvent.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/timeout/TimeoutException.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/stream/ChunkedNioFile.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/logging/LoggingHandler.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/logging/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
  jar/io/netty/handler/ssl/NotSslRecordException.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
jar/io.netty.handler/stream/ChunkedFile.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io.netty.handler/stream/ChunkedInput.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io.netty.handler/stream/ChunkedNioStream.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io.netty.handler/timeout/IdleStateHandler.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/META-INF/maven/io.netty/netty-handler/pom.xml

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io.netty.handler/traffic/AbstractTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 * /

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/PemEncoded.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslCertificateException.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/Java8SslUtils.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/DelegatingSslContext.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/PemX509Certificate.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/PemPrivateKey.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslKeyMaterialManager.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/flow/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/package-info.java
*/
jar/io/netty/handler/flush/FlushConsolidationHandler.java
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
jar/io/netty/handler/ssl/PemValue.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

// Try the OpenJDK's proprietary implementation.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
jar/io/netty/handler/ssl/util/SelfSignedCertificate.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-
jar/io/netty/handler/traffic/GlobalChannelTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.
/*
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
  * /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/OpenSslPrivateKeyMethod.java
  * /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/PseudoRandomFunction.java
  * /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/address/package-info.java
  * /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/ssl/SslMasterKeyHandler.java
  * /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/address/DynamicAddressConnectHandler.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
  * /opt/ws_local/PERMITS_SQL/1072871132_1595214674.28/0/netty-handler-4-1-42-final-sources-1-jar/io/netty/handler/traffic/GlobalTrafficShapingHandler.java
1.679 gir1.2-gdkpixbuf-2.0 2.32.2-1ubuntu1.6

1.679.1 Available under license:

GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
   59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor’s protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don’t assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to
permit developers of non-free programs to use free libraries, while
preserving your freedom as a user of such programs to change the free
libraries that are incorporated in them. (We have not seen how to achieve
this as regards changes in header files, but we have achieved it as regards
changes in the actual functions of the Library.) The hope is that this
will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, while the latter only
works together with the library.

Note that it is possible for a library to be covered by the ordinary
General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which
contains a notice placed by the copyright holder or other authorized
party saying it may be distributed under the terms of this Library
General Public License (also called "this License"). Each licensee is
addressed as "you".

A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is
included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

d) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the source code distributed need not include anything that is normally
distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Library General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Library General Public License for more details.

You should have received a copy of the GNU Library General Public
License along with this library; if not, write to the
Free Software Foundation, Inc., 59 Temple Place - Suite 330,
Boston, MA 02111-1307 USA.
Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.680 sqlite 3.30.1

1.681 babel-plugin-transform-es2015-spread 6.22.0 6.22.0

1.681.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.682 deep-equal 0.2.2
1.682.1 Available under license:

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.683 bson 1.0.9
1.683.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.684 map-cache 0.2.2

1.684.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.685 httpsnippet 1.16.5 1.16.5
1.685.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Mashape (https://www.mashape.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.686 javax-el-api 2.2.4
1.686.1 Available under license :
Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 1997-2011 Oracle
and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General
Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL")
Found license 'General Public License 2.0' in /* Copyright (c) 1997-2011 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License * Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You may obtain a copy of the License at * http://www.apache.org/licenses/LICENSE-2.0 * distributed under the License is distributed on an "AS IS" BASIS,'

1.687 json-path 2.4.0

1.687.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011 the original author or authors.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/InvalidJsonException.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/InvalidCriteriaException.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/JsonPathException.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/ParseContext.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/Configuration.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/internal/PathLateBindingValue.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/internal/Utils.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/internal/function/latebinding/PathLateBindingValue.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/ParseContext.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-jar/com/jayway/jsonpath/Configuration.java
* /opt/cola/permits/1199743787_1630577685.84/0/json-path-2-4-0-sources-7-
1.688 regjsgen 0.5.2

1.688.1 Available under license:

The MIT License (MIT)

Copyright 2014-2020 Benjamin Tan <https://ofcr.se/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.689 commons-pool 2.6.0

1.689.1 Available under license:

Apache Commons Pool
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE COMMONS POOL DERIVATIVE WORKS:

The LinkedBlockingDeque implementation is based on an implementation written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at

http://creativecommons.org/licenses/publicdomain

1.690 escape 3.6.0

1.690.1 Available under license:
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.691 native-promise-only 0.8.1 0.8.1
1.691.1 Available under license :

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.692 @babel/compat-data 7.13.12
1.692.1 Available under license :

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.693 drill-storage-elasticsearch 1.19.0

1.693.1 Available under license:

Drill : Contrib : Storage : ElasticSearch
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensors" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.694 ip-address 5.3.1
1.694.1 Available under license :

No license file was found, but licenses were detected in source scan.

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#     http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/ws.local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/IPAddressResources.properties
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2020 Sean C Foley
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * or at
 * https://github.com/seancfoley/IPAddress/blob/master/LICENSE
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
*/

Found in path(s):
* /opt/ws.local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/IPAddressSection.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2018 Sean C Foley
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * or at
 * https://github.com/seancfoley/IPAddress/blob/master/LICENSE
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/AddressComparator.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/large/IPAddressLargeDivisionGrouping.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/AddressDivisionGroupingBase.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/IPAddressSegment.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/AddressStringParameters.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/IPAddressSegmentSeries.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/validate/Validator.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/validate/IPAddressProvider.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/mac/MACAddress.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/standard/IPAddressDivisionGrouping.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/HostNameException.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/validate/HostIdentifierStringValidator.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/SizeMismatchException.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/standard/IPAddressDivisionGrouping.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/IPv4AddressSegment.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/validate/HostIdentifierStringValidator.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/IPv6Address.java

Open Source Used In DNAC Platform Antelope 2359
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2020 Sean C Foley
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* or at
* https://github.com/seancfoley/IPAddress/blob/master/LICENSE
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 Sean C Foley
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
* or at
*     https://github.com/seancfoley/IPAddress/blob/master/LICENSE
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/ipv6/IPv6AddressSeqRange.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/AddressGenericDivision.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/validate/package-info.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/validate/ParsedAddressGrouping.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/util/package-info.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/package-info.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/util/sql/package-info.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/package-info.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/util/package-info.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/package-info.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/validate/ParsedMACAddress.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/ipv4/IPv4AddressSeqRange.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/mac/package-info.java
* /opt/ws_local/PERMITS_SQL/1077201543_1596778639.63/0/ipaddress-5-3-1-sources-jar/inet/ipaddr/format/package-info.java
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright 2015 MongoDB Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.696 sparsebitset 1.2

1.696.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text
file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and
description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.697 async-replace 1.0.1
1.697.1 Available under license :
MIT

1.698 javassist 3.22.0-GA
1.698.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
 * Javassist, a Java-bytecode translator toolkit.
 * Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.
 * *
 * The contents of this file are subject to the Mozilla Public License Version
 * 1.1 (the "License"); you may not use this file except in compliance with
 * the License. Alternatively, the contents of this file may be used under
 * the terms of the GNU Lesser General Public License Version 2.1 or later,
 * or the Apache License Version 2.0.
 * *
 * Software distributed under the License is distributed on an "AS IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
 * for the specific language governing rights and limitations under the
 * License.
 * */

Found in path(s):
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
jar/javassist/bytecode/LocalVariableAttribute.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/TokenId.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/Loader.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/DeprecatedAttribute.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/stackmap/TypeTag.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/ast/BinExpr.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/rmi/Proxy.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/runtime/DotClass.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/.CtNewWrappedConstructor.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/SyntaxError.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/LocalVariableTypeAttribute.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/expr/Handler.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/convert/TransformNewClass.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/ast/StringL.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/analysis/SubroutineScanner.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/ast/StringL.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/StackMap.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/runtime/Desc.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/expr/ExprEditor.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/ClassClassPath.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/expr/Expr.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/rmi/StubGenerator.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/ByteStream.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/rmi/ObjectImporter.java
Open Source Used In DNAC Platform Antelope 2375
Open Source Used In DNAC Platform Antelope 2376

*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/util/proxy/SecurityActions.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/convert/TransformFieldAccess.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/AccessFlag.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/InnerClassesAttribute.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/runtime/Inner.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/analysis/Type.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/ast/Pair.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/SerialVersionUID.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/AnnotationsAttribute.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/CodeIterator.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/util/proxy/DefineClassHelper.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/reflect/Sample.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/Modifier.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/KeywordTable.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/scopedpool/ScopedClassPool.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/reflect/Reflection.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/reflect/CannotReflectException.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/ConstantAttribute.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/ast/IntConst.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/ast/Declarator.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/compiler/NoFieldException.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/convert/TransformAccessArrayField.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/analysis/IntQueue.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/CtClass.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/reflect/CannotInvokeException.java
*/opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/util/proxy/SerializedProxy.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/analysis/Util.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/MethodExitException.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/AttributeInfo.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/analysis/Subroutine.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/reflect/Metalevel.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/tools/reflect/ClassMetaobject.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/expr/FieldAccess.java
* /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-jar/javassist/bytecode/ClassFilePrinter.java
No license file was found, but licenses were detected in source scan.

/*
 * Javassist, a Java-bytecode translator toolkit.
 * Copyright (C) 2004 Bill Burke. All Rights Reserved.
 *
 * The contents of this file are subject to the Mozilla Public License Version
 * 1.1 (the "License"); you may not use this file except in compliance with
 * the License. Alternatively, the contents of this file may be used under
 * the terms of the GNU Lesser General Public License Version 2.1 or later,
 * or the Apache License Version 2.0.
 *
 * Software distributed under the License is distributed on an "AS IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
 * for the specific language governing rights and limitations under the
 * License.
 */

Found in path(s):
/*
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/AnnotationMemberValue.java
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/FloatMemberValue.java
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/ClassMemberValue.java
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/CharMemberValue.java
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/ShortMemberValue.java
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/EnumMemberValue.java
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/IntegerMemberValue.java
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/ArrayMemberValue.java
 * /opt/cola/permits/1136405427_1613762483.76/0/javassist-3-22-0-ga-sources-4-
 * jar/javassist/bytecode/annotation/ByteMemberValue.java
 */
1.699 script-manager 0.10.2

1.699.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jan Blaha

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.700 remove-trailing-separator 1.1.0

1.700.1 Available under license:

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
 REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

1.701 unicode-match-property-ecmascript
1.0.4
1.701.1 Available under license :
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
“Software”), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.702 jsonify 0.0.0
1.702.1 Available under license :
No license file was found, but licenses were detected in source scan.

# jsonify

This module provides Douglas Crockford's JSON implementation without modifying
any globals.

`stringify` and `parse` are merely exported without respect to whether or not a
global `JSON` object exists.
# methods

``` js
var json = require('jsonify');
```

## json.parse(source, reviver)

Return a new javascript object from a parse of the `source` string.

If a `reviver` function is specified, walk the structure passing each name/value pair to `reviver.call(parent, key, value)` to transform the `value` before parsing it.

## json.stringify(value, replacer, space)

Return a string representation for `value`.

If `replacer` is specified, walk the structure passing each name/value pair to `replacer.call(parent, key, value)` to transform the `value` before stringifying it.

If `space` is a number, indent the result by that many spaces.
If `space` is a string, use `space` as the indentation.

# install

With [npm](http://npmjs.org) do:

```
npm install jsonify
```

To use this module in the browser, check out [browserify](https://github.com/substack/node-browserify).

# license

public domain

Found in path(s):

* /opt/cola/permits/1108587829_1609900447.26/0/jsonify-master-zip/jsonify-master/readme.markdown
1.703 querystringify 2.2.0

1.703.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.704 es-abstract 1.18.0

1.704.1 Available under license:
The MIT License (MIT)

Copyright (C) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.705 eslint-import-resolver-node 0.3.4

1.705.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Ben Mosher

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.706 object-assign 4.1.0 4.1.0

1.706.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.707 junit-platform-engine 1.7.0

1.707.1 Available under license:

Eclipse Public License - v 2.0

=================================

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

### 1. Definitions

Contribution means:
* **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
* **b)** in the case of each subsequent Contributor:
  * **i)** changes to the Program, and
  * **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations,
interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)** distributing or **b)** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

### 2. Grant of Rights

**a)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

**b)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

**c)** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

**d)** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

**e)** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

### 3. Requirements

**3.1** If a Contributor Distributes the Program in any form, then:

* **a)** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and
the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
* **(i)** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
* **(ii)** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
* **(iii)** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
* **(iv)** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

**3.2** When the Program is Distributed as Source Code:

* **a)** it must be made available under this Agreement, or if the Program **(i)** is combined with other material in a separate file or files made available under a Secondary License, and **(ii)** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and
* **b)** a copy of this Agreement must be included with each copy of the Program.

**3.3** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

### 4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **a)** promptly notify the Commercial Contributor in writing of such claim, and **b)** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

### 5. No Warranty
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY
APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION,
ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the
appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights
under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable
laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

### 6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY
APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

### 7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or
enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such
provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit)
alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes
such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such
litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or
conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and
distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this
Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the
Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the
right to publish new versions (including revisions) of this Agreement from time to time. No one other than the
Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement
Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable
separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program
(including Contributions) may always be Distributed subject to the version of the Agreement under which it was
received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the
Program (including its Contributions) under the new version.
Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

#### Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-$<subcomponent>$.md files.

-----

1.708 json-stable-stringify 1.0.1

1.708.1 Available under license:

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
1.709 ultron 1.0.2

1.709.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.710 objenesis 2.5.1

1.710.1 Available under license:

// NOTICE file corresponding to the section 4d of The Apache License,
// Version 2.0, in this case for Objenesis
// -------------------------------

Objenesis
Copyright 2006-2017 Joe Walnes, Henri Tremblay, Leonardo Mesquita

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.711 @babel/highlight 7.10.4

1.711.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.712 json-schema-faker 0.5.0-rcv.34 0.5.0-rcv.34

1.712.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2018 Alvaro Cabrera

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.713 is-callable 1.2.2

1.713.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.714 fbjs 0.8.17 0.8.17

1.714.1 Available under license:
MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.715 number-is-nan 1.0.1

1.715.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.716 regenerator-transform 0.14.5

1.716.1 Available under license:
MIT License

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.717 commander 1.2.0
1.717.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Laracasts <jeffrey@laracasts.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.718 indexof 0.0.1 0.0.1
1.719 end-of-stream 1.4.4 1.4.4
1.719.1 Available under license :  
The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.720 p-locate 2.0.0
1.720.1 Available under license :  
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
1.721 accessors-smart 1.2

1.721.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011 JSON-SMART authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1177937262_1625568439.63/0/accessors-smart-1-2-sources-5-jar/net/minidev/asm/BeansAccess.java
* /opt/cola/permits/1177937262_1625568439.63/0/accessors-smart-1-2-sources-5-jar/net/minidev/asm/DynamicClassLoader.java
* /opt/cola/permits/1177937262_1625568439.63/0/accessors-smart-1-2-sources-5-jar/net/minidev/asm/Accessor.java
* /opt/cola/permits/1177937262_1625568439.63/0/accessors-smart-1-2-sources-5-jar/net/minidev/asm/ASMUtil.java

1.722 duplexify 3.7.1 3.7.1

1.722.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.723 @babel/template 7.10.4

1.723.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.724 make-dir 3.1.0
1.724.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.725 ansi-styles 2.2.1 2.2.1

1.725.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.726 mkdirp 0.5.1 0.5.1

1.726.1 Available under license:
Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.727 kerb-server 1.0.0

1.727.1 Available under license:

Kerby-kerb Server
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly

Open Source Used In DNAC Platform Antelope 2405
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.728 jsreport-fs-store 2.9.1
1.728.1 Available under license:
MIT

1.729 string_decoder 0.10.31 0.10.31
1.729.1 Available under license:
Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE SOFTWARE.

1.730 lodash.isstring 4.0.1
1.730.1 Available under license:
Copyright 2012-2016 The Dojo Foundation <http://dojofoundation.org/>
Based on Underscore.js, copyright 2009-2016 Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.731 calcite-core 1.26.0

1.731.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Apache Calcite  
Copyright 2012-2020 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (http://www.apache.org/).

This product is based on source code originally developed  
by DynamoBI Corporation, LucidEra Inc., SQLstream Inc. and others  
under the auspices of the Eigenbase Foundation  
and released as the LucidDB project.

The web site includes files generated by Jekyll.

1.732 escape-goat 2.1.1

1.732.1 Available under license:  
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the  
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit  
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the  
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,  
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A  
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR  
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN  
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.733 camelize 1.0.0 1.0.0

1.733.1 Available under license:  
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated  
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the  
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit  
persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.734 phantom-page-eval 1.2.0
1.734.1 Available under license:
   The MIT License (MIT)

Copyright (c) 2018 bjrmatos

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.735 duplexer3 0.1.4
1.735.1 Available under license:
   Copyright (c) 2013, Deoxxa Development

======================================
All rights reserved.
-------------------

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of Deoxxa Development nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY DEOXXA DEVELOPMENT "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL DEOXXA DEVELOPMENT BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.736 raw-body 2.3.3 2.3.3
1.736.1 Available under license :
The MIT License (MIT)

Copyright (c) 2013-2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.737 hive-service-rpc 3.1.2

1.737.1 Available under license:

Hive Service RPC
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.738 capnproto 0.5.3-2ubuntu1.1

1.738.1 Available under license:

Copyright (c) 2013-2014 Sandstorm Development Group, Inc. and contributors
Licensed under the MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright 2008, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# This file contains a list of people who've made non-trivial contribution to the Google C++ Testing Framework project. People who commit code to the project are encouraged to add their names here. Please keep the list sorted by first names.

Ajay Joshi <jaj@google.com>
Balzs Dn <balazs.dan@gmail.com>
Bharat Mediratta <bharat@menalto.com>
Chandler Carruth <chandlerc@google.com>
Chris Prince <cprince@gmail.com>
Chris Taylor <taylorc@gmail.com>
Dan Egnor <egnor@gmail.com>
Eric Roman <eroman@chromium.org>
Hady Zalek <hady.zalek@gmail.com>
Jeffrey Yasskin <jyasskin@gmail.com>
Ji Sigursson <joi@gmail.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kenton Varda <kenton@gmail.com>
Manuel Klimek <klimek@gmail.com>
Markus Heule <markus.heule@gmail.com>
Mika Raento <mikie@iki.fi>
Mikls Fazekas <mfazekas@szemafor.com>
1.739 ms 2.0.0 2.0.0

1.739.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Zeit, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.740 netty-transport 4.1.59.Final

1.740.1 Available under license:
No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
* /opt/cola/permits/1145449543_1616156406.95/0/netty-transport-4-1-59-final-1-jar/META-INF/native-image/io.netty/transport/native-image.properties

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Implementation-Title: Netty/Transport
Bundle-Description: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients.
Automatic-Module-Name: io.netty.transport
Bundle-License: https://www.apache.org/licenses/LICENSE-2.0
Bundle-SymbolicName: io.netty.transport
Implementation-Version: 4.1.59.Final
Built-By: root
Bnd-LastModified: 1612782105444
Bundle-ManifestVersion: 2
Implementation-Vendor-Id: io.netty
Bundle-DocURL: https://netty.io/
Bundle-Vendor: The Netty Project
Import-Package: io.netty.buffer;version="[4.1,5)";io.netty.resolver;version="[4.1,5)";io.netty.util;version="[4.1,5)";io.netty.util.concurrent;version="[4.1,5)";io.netty.util.internal;version="[4.1,5)";io.netty.util.internal.logging;version="[4.1,5)";sun.nio.ch;resolution:=optional;org.eclipse.jetty.npn;version="[1,2)";resolution:=optional;org.eclipse.jetty.alpn;version="[1,2)";resolution:=optional
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.6))"
Tool: Bnd-2.4.1.201501161923
Implementation-Vendor: The Netty Project

---

Open Source Used In DNAC Platform Antelope 2422
Bundle-Name: Netty/Transport
Bundle-Version: 4.1.59.Final
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_252
Implementation-URL: https://netty.io/netty-transport/

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~   https://www.apache.org/licenses/LICENSE
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

1.741 cookie 0.1.5
1.741.1 Available under license :
   (The MIT License)

Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.742 jackson-jaxrs-base 2.8.4

1.742.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.743 cacheable-request 6.1.0

1.743.1 Available under license:

MIT License

Copyright (c) 2017 Luke Childs

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.744 slash 1.0.0 1.0.0

1.744.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.745 eslint 3.19.0

1.745.1 Available under license:

ESLint
Copyright JS Foundation and other contributors, https://js.foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.746 color-name 1.1.4

1.746.1 Available under license :

The MIT License (MIT)
Copyright (c) 2015 Dmitry Ivanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.747 accepts 1.2.13 1.2.13

1.747.1 Available under license :

(The MIT License)
Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.748 netty-handler-proxy 4.1.42.Final

1.748.1 Available under license:

No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871204_1595214741.64/0/netty-handler-proxy-4.1.42-final-sources-1-
jar/META-INF/maven/io.netty/netty-handler-proxy/pom.xml

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871204_1595214741.64/0/netty-handler-proxy-4-1-42-final-sources-1.jar/io/netty/handler/proxy/ProxyConnectionEvent.java
* /opt/ws_local/PERMITS_SQL/1072871204_1595214741.64/0/netty-handler-proxy-4-1-42-final-sources-1.jar/io/netty/handler/proxy/HttpProxyHandler.java
* /opt/ws_local/PERMITS_SQL/1072871204_1595214741.64/0/netty-handler-proxy-4-1-42-final-sources-1.jar/io/netty/handler/proxy/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871204_1595214741.64/0/netty-handler-proxy-4-1-42-final-sources-1.jar/io/netty/handler/proxy/Socks4ProxyHandler.java
* /opt/ws_local/PERMITS_SQL/1072871204_1595214741.64/0/netty-handler-proxy-4-1-42-final-sources-1.jar/io/netty/handler/proxy/ProxyConnectException.java
* /opt/ws_local/PERMITS_SQL/1072871204_1595214741.64/0/netty-handler-proxy-4-1-42-final-sources-1.jar/io/netty/handler/proxy/ProxyHandler.java
* /opt/ws_local/PERMITS_SQL/1072871204_1595214741.64/0/netty-handler-proxy-4-1-42-final-sources-1.jar/io/netty/handler/proxy/Socks5ProxyHandler.java

1.749 is-typedarray 1.0.0

1.749.1 Available under license:
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.750 commons-pool 2.6.2
1.751 google-discovery-to-swagger 2.1.0 2.1.0
1.751.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 APIs-guru

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.752 amcharts3-responsive 1.0.5 1.0.5
1.752.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.753 safe-buffer 5.1.2
1.753.1 Available under license:
The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.754 blob 0.0.4 0.0.4

1.754.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.755 es6-promise 3.3.1 3.3.1

1.755.1 Available under license:

Copyright (c) 2014 Yehuda Katz, Tom Dale, Stefan Penner and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.756 babel-preset-env 1.7.0 1.7.0

1.756.1 Available under license:

MIT License

Copyright (c) 2016-2017 Babel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.757 to-fast-properties 2.0.0

1.757.1 Available under license:

MIT License

Copyright (c) 2014 Petka Antonov
2015 Sindre Sorhus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.758 react-textarea-autosize 6.1.0

1.758.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013 Andrey Popp

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
1.759 commons-logging 1.1.1
1.759.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

// ------------------------------------------------------------------
// NOTICE file corresponding to the section 4d of The Apache License,
// Version 2.0, in this case for Commons Logging
// ------------------------------------------------------------------

Commons Logging
Copyright 2001-2007 The Apache Software Foundation

This product includes/uses software(s) developed by 'an unknown organization'
- Unnamed - avalon-framework:avalon-framework:jar:4.1.3
- Unnamed - log4j:log4j:jar:1.2.12
- Unnamed - logkit:logkit:jar:1.0.1

1.760 jnr-ffi 2.1.7
1.760.1 Available under license:

/*
 * Copyright (C) 2010 Wayne Meissner
 *
 * This file is part of the JNR project.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product
names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: HOW TO APPLY THE APACHE LICENSE TO YOUR WORK

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
1.761 loader-utils 1.4.0

1.761.1 Available under license:
Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.762 pinkie 1.0.0 1.0.0

1.762.1 Available under license:
The MIT License (MIT)

Copyright (c) Vsevolod Strukchinsky <floatdrop@gmail.com> (github.com/floatdrop)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
This Confluent Community License Agreement Version 1.0 (the Agreement) sets forth the terms on which Confluent, Inc. (Confluent) makes available certain software made available by Confluent under this Agreement (the Software). BY INSTALLING, DOWNLOADING, ACCESSING, USING OR DISTRIBUTING ANY OF THE SOFTWARE, YOU AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO SUCH TERMS AND CONDITIONS, YOU MUST NOT USE THE SOFTWARE. IF YOU ARE RECEIVING THE SOFTWARE ON BEHALF OF A LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE THE ACTUAL AUTHORITY TO AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT ON BEHALF OF SUCH ENTITY. Licensee means you, an individual, or the entity on whose behalf you are receiving the Software.

1. LICENSE GRANT AND CONDITIONS.

1.1 License. Subject to the terms and conditions of this Agreement, Confluent hereby grants to Licensee a non-exclusive, royalty-free, worldwide, non-transferable, non-sublicenseable license during the term of this Agreement to: (a) use the Software; (b) prepare modifications and derivative works of the Software; (c) distribute the Software (including without limitation in source code or object code form); and (d) reproduce copies of the Software (the License). Licensee is not granted the right to, and Licensee shall not, exercise the License for an Excluded Purpose. For purposes of this Agreement, Excluded Purpose means making available any software-as-a-service, platform-as-a-service, infrastructure-as-a-service or other similar online service that competes with Confluent products or services that provide the Software.

1.2 Conditions. In consideration of the License, Licensees distribution of the Software is subject to the following conditions:

(a) Licensee must cause any Software modified by Licensee to carry prominent notices stating that Licensee modified the Software.

(b) On each Software copy, Licensee shall reproduce and not remove or
alter all Confluent or third party copyright or other proprietary notices contained in the Software, and Licensee must provide the notice below with each copy.

This software is made available by Confluent, Inc., under the terms of the Confluent Community License Agreement, Version 1.0 located at http://www.confluent.io/confluent-community-license. BY INSTALLING, DOWNLOADING, ACCESSING, USING OR DISTRIBUTING ANY OF THE SOFTWARE, YOU AGREE TO THE TERMS OF SUCH LICENSE AGREEMENT.

1.3 Licensee Modifications. Licensee may add its own copyright notices to modifications made by Licensee and may provide additional or different license terms and conditions for use, reproduction, or distribution of Licensees modifications. While redistributing the Software or modifications thereof, Licensee may choose to offer, for a fee or free of charge, support, warranty, indemnity, or other obligations. Licensee, and not Confluent, will be responsible for any such obligations.

1.4 No Sublicensing. The License does not include the right to sublicense the Software, however, each recipient to which Licensee provides the Software may exercise the Licenses so long as such recipient agrees to the terms and conditions of this Agreement.

2. TERM AND TERMINATION. This Agreement will continue unless and until earlier terminated as set forth herein. If Licensee breaches any of its conditions or obligations under this Agreement, this Agreement will terminate automatically and the License will terminate automatically and permanently.

3. INTELLECTUAL PROPERTY. As between the parties, Confluent will retain all right, title, and interest in the Software, and all intellectual property rights therein. Confluent hereby reserves all rights not expressly granted to Licensee in this Agreement. Confluent hereby reserves all rights in its trademarks and service marks, and no licenses therein are granted in this Agreement.

4. DISCLAIMER. CONFLUENT HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE.

5. LIMITATION OF LIABILITY. CONFLUENT WILL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, LOST PROFITS OR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, INDIRECT, OR DIRECT DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING OUT OF THIS AGREEMENT. THE FOREGOING SHALL APPLY TO THE EXTENT PERMITTED BY APPLICABLE LAW.

6. GENERAL.
6.1 Governing Law. This Agreement will be governed by and interpreted in accordance with the laws of the state of California, without reference to its conflict of laws principles. If Licensee is located within the United States, all disputes arising out of this Agreement are subject to the exclusive jurisdiction of courts located in Santa Clara County, California. USA. If Licensee is located outside of the United States, any dispute, controversy or claim arising out of or relating to this Agreement will be referred to and finally determined by arbitration in accordance with the JAMS International Arbitration Rules. The tribunal will consist of one arbitrator. The place of arbitration will be Palo Alto, California. The language to be used in the arbitral proceedings will be English. Judgment upon the award rendered by the arbitrator may be entered in any court having jurisdiction thereof.

6.2 Assignment. Licensee is not authorized to assign its rights under this Agreement to any third party. Confluent may freely assign its rights under this Agreement to any third party.

6.3 Other. This Agreement is the entire agreement between the parties regarding the subject matter hereof. No amendment or modification of this Agreement will be valid or binding upon the parties unless made in writing and signed by the duly authorized representatives of both parties. In the event that any provision, including without limitation any condition, of this Agreement is held to be unenforceable, this Agreement and all licenses and rights granted hereunder will immediately terminate. Waiver by Confluent of a breach of any provision of this Agreement or the failure by Confluent to exercise any right hereunder will not be construed as a waiver of any subsequent breach of that right or as a waiver of any other right.

Apache Kafka
Copyright 2016 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This distribution has a binary dependency on jersey, which is available under the CDDL License. The source code of jersey can be found at https://github.com/jersey/jersey/.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing
Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Java ClassMate library was originally written by Tatu Saloranta (tatu.saloranta@iki.fi)

Other developers who have contributed code are:

* Brian Langel

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="Home page of The Apache Software Foundation">
<link rel="apple-touch-icon" sizes="57x57" href="/favicons/apple-touch-icon-57x57.png">
<link rel="apple-touch-icon" sizes="60x60" href="/favicons/apple-touch-icon-60x60.png">
<link rel="apple-touch-icon" sizes="72x72" href="/favicons/apple-touch-icon-72x72.png">
<link rel="apple-touch-icon" sizes="76x76" href="/favicons/apple-touch-icon-76x76.png">
<link rel="apple-touch-icon" sizes="114x114" href="/favicons/apple-touch-icon-114x114.png">
<link rel="apple-touch-icon" sizes="120x120" href="/favicons/apple-touch-icon-120x120.png">
<link rel="apple-touch-icon" sizes="144x144" href="/favicons/apple-touch-icon-144x144.png">
<link rel="apple-touch-icon" sizes="152x152" href="/favicons/apple-touch-icon-152x152.png">
<link rel="apple-touch-icon" sizes="180x180" href="/favicons/apple-touch-icon-180x180.png">
<link rel="icon" type="image/png" href="/favicons/favicon-32x32.png" sizes="32x32">
<link rel="icon" type="image/png" href="/favicons/favicon-194x194.png" sizes="194x194">
<link rel="icon" type="image/png" href="/favicons/favicon-96x96.png" sizes="96x96">
<link rel="manifest" href="/favicons/manifest.json">
<link rel="shortcut icon" href="/favicons/favicon.ico">
<meta name="msapplication-TileColor" content="#603cba">
Apache License, Version 2.0

<!-- Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 . Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. -->

</head>

<body>

<!-- Navigation -->
<header>
<nav class="navbar navbar-default navbar-fixed-top">
<div class="container">
<div class="navbar-header">
<button class="navbar-toggle" type="button" data-toggle="collapse" data-target="#mainnav-collapse">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a href="#" class="navbar-brand"><span class="glyphicon glyphicon-home"></span></a>
</div>
<div class="collapse navbar-collapse" id="mainnav-collapse">
<div style="line-height:20px; padding-top:5px; float:left"><a href="/">Home</a> &raquo; <a href="/licenses/">Licenses</a></div>
<ul class="nav navbar-nav navbar-right">
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown">About <span class="caret"></span></a>
<ul class="dropdown-menu" role="menu">
<li><a href="/foundation/">Overview</a></li>
<li><a href="/foundation/members.html">Members</a></li>
<li><a href="/foundation/how-it-works.html">Process</a></li>
<li><a href="/foundation/sponsorship.html">Sponsorship</a></li>
<li><a href="/foundation/glossary.html">Glossary</a></li>
</ul>
</li>
</ul>
</div>
</div>
</nav>
</header>

</body>
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial
ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution.”

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor
that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License;
- You must cause any modified files to carry prominent notices stating that You changed the files;
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets """" replaced with your own...
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This copy of Jackson JSON processor annotations is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

   http://www.apache.org/licenses/LICENSE-2.0

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

   http://www.apache.org/licenses/LICENSE-2.0

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
Files: *
Copyright: 2018 Confluent, Inc.
License: Apache-2

Files: core/*
Copyright: 2015 Confluent, Inc.
License: Confluent Community License

License: Confluent Community License
Licensed under the Confluent Community License; you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.confluent.io/confluent-community-license

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License: Apache-2
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

On Debian systems, the Apache 2.0 license can be found in /usr/share/common-licenses/Apache-2.0.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

This distribution has a binary dependency on jersey, which is available under the CDDL License as described below.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL - Version 1.1)

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under
common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause
the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty
percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.
2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims,
the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use,
reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or
without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use,
practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or
otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the
Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the
combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims,
each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce,
modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions
thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger
Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either
alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer
for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof);
and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such
combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or
otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted
from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version,
or (ii) the combination of Modifications made by that Contributor with other software (except as part of the
Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.
3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of
this License are fulfilled for the Covered Software.

4. Versions of the License.
4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice
from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise
make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Apache Avro
Copyright 2009-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
The following commands were used to generate license and notice files. Replace <VERSION> with the Schema Registry version, <SRC_PATH> with the path to the Schema Registry source directory, and <LICENSE_TOOL_PATH> with the path of the license tool.

cd <SRC_PATH>
mvn package -DskipTests
mkdir /tmp/jars
mkdir /tmp/overrides
cd <LICENSE_TOOL_PATH>
./bin/run_license_job.bash -i /tmp/jars -l <SRC_PATH>/licenses -n <SRC_PATH>/notices -h <SRC_PATH>/licenses-and-notices.html -> /tmp/overrides
<!DOCTYPE html PUBLIC "-/W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>

<body>
<!DOCTYPE HTML PUBLIC "-/W3C//DTD HTML 4.0 Transitional//EN">
<html>
<!Page [ margin: 0.79in ] P [ margin-bottom: 0.08in ] H1 [ margin-top: 0in; margin-bottom: 0.17in; widows:
1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual
property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the
Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.</p><p>3.4. Application of Additional Terms.</p><p>You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients’ rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.</p><p>3.5. Distribution of Executable Versions.</p><p>You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that you are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.</p><p>3.6. Larger Works.</p><p>You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.</p><p>4. Versions of the License.</p><p>4.1. New Versions.</p><p>Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no other than the license steward has the right to modify this License.</p><p>4.2. Effect of New Versions.</p><p>You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.</p><p>4.3. Modified Versions.</p><p>You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.</p><p>5. DISCLAIMER OF WARRANTY.</p><p>Covered Software is provided under this License on an “AS IS” basis, without warranty of any kind, either expressed or implied, including, without limitation, warranties that the Covered Software is free of defects.
MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make
sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

**TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The Program, below, refers to any such program or work, and a work based on the Program; means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and
the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full
You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and any later version, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) &lt;year&gt; &lt;name of author&gt;

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a copyright disclaimer; for details type `show d'. This is free software; you are welcome to redistribute it under certain conditions; type `show c' for details.

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code." exception as provided by Oracle in the License file that accompanied this code.

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.
1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with
code not governed by the terms of this License.
</p>
</li>

<li>
<p>
1.7. "License" means this document.
</p>
</li>

<li>
<p>
1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
</p>
</li>

<li>
<p>
1.9. "Modifications" means the Source Code and Executable form of any of the following:
</p>
<ul>
<li>A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
</li>
<li>B. Any new file that contains any part of the Original Software or previous Modification; or
</li>
<li>C. Any new file that is contributed or otherwise made available under the terms of this License.
</li>
</ul>
</li>
1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.
2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License. Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.
2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- under intellectual property rights (other than patent or trademark) licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

- under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any...
code that Contributor has deleted from the Contributor Version; (2) for infringements caused by:
(i) third party modifications of Contributor Version, or (ii) the combination of Modifications
made by that Contributor with other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that
Contributor.
</p>
</li>
</ul>
</li>

</li>
<p><b>3. Distribution Obligations. </b></p>
<ul>
<li>
<p><b>3.1. Availability of Source Code.</b></p>
<p>Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.
</p>
</li>
<li>
<p><b>3.2. Modifications. </b></p>
<p>The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.
</p>
</li>
<li>
<p><b>3.3. Required Notices. </b></p>
<p></p>
</li>
</ul>
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.
4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You:
(a)&nbsp;rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b)&nbsp;otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN &ldquo;AS IS&rdquo; BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.
6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION MAY NOT APPLY TO YOU.
LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

</li>
</dl>

<li>
<p><b>8. U.S. GOVERNMENT END USERS.</b></p>
</li>

The Covered Software is a &ldquo;commercial item,&rdquo; as that term is defined in 48&nbsp;C.F.R.&nbsp;2.101 (Oct. 1995), consisting of &ldquo;commercial computer software&rdquo; (as that term is defined at 48 C.F.R. &sect;&nbsp;252.227-7014(a)(1)) and &ldquo;commercial computer software documentation&rdquo; as such terms are used in 48&nbsp;C.F.R.&nbsp;12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

</li>

<li>
<p><b>9. MISCELLANEOUS.</b></p>
</li>

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction’s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys’ fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export
administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

</p>
</li>

<li>
<p>10. RESPONSIBILITY FOR CLAIMS.</p>
<p>As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.</p>
</li>

</ul>

<h3>The GNU General Public License (GPL) Version 2, June 1991</h3>

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
<p><b>Preamble</b></p>

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.
Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
<b>8.</b> If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

<b>9.</b> The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

<b>10.</b> If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

<b>11.</b> BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>
</p>
</li>
<li>
</p>
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
</p>
</li>
</p>
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
</p>
</li>
</p>
You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
</p>
</li>
</p>
Also add information on how to contact you by electronic and paper mail.
</p>
</p>
If the program is interactive, make it output a short notice like this when it starts in an interactive mode:
</p>
<ul>
<li>
</p>
Gnomovision version 69, Copyright (C) year name of author<br />
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.
</p>
</li>
</li>
</p>
The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words "Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your
choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

</p>
</li>

</ul>

</body>
</html>

This copy of Jackson JSON processor is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/

A copy is also included with both the the downloadable source code package and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/asl"; and in jar "META-INF/"

The project is licensed under the Confluent Community License, except for client libs, which is under the Apache 2.0 license.

See LICENSE file in each subfolder for detailed license agreement.

Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact
## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. ––>

</head>

<body>

<!– Navigation -->

<header>

<nav class="navbar navbar-default navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button class="navbar-toggle" type="button" data-toggle="collapse" data-target="#mainnav-collapse">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a href="#" class="navbar-brand"><span class="glyphicon glyphicon-home"></span></a>

</div>

<div class="collapse navbar-collapse" id="mainnav-collapse">

<div style="line-height:20px; padding-top:5px; float:left"><a href="/">Home</a>&nbsp;&raquo&nbsp;<a href="/licenses/">Licenses</a></div>

<ul class="nav navbar-nav navbar-right">

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">About <span class="caret"></span></a>

<ul class="dropdown-menu" role="menu">

<li><a href="/foundation/">Overview</a></li>

<li><a href="/foundation/members.html">Members</a></li>

<li><a href="/foundation/how-it-works.html">Process</a></li>

<li><a href="/foundation/sponsorship.html">Sponsorship</a></li>

<li><a href="/foundation/glossary.html">Glossary</a></li>

<li><a href="/foundation/preFAQ.html">FAQ</a></li>

<li><a href="/foundation/contact.html">Contact</a></li>

</ul>

</li>

<li><a href="/index.html#projects-list">Projects</a></li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">People <span class="caret"></span></a>

<ul class="dropdown-menu" role="menu">

<li><a href="http://people.apache.org/">Overview</a></li>


<li><a href="/foundation/how-it-works.html#meritocracy">Meritocracy</a></li>

<li><a href="/foundation/how-it-works.html#roles">Roles</a></li>

<li><a href="http://planet.apache.org/">Planet Apache</a></li>

</ul>

</li>

</ul>

</div>

</div>

</nav>

</header>
The Apache Software Foundation uses various licenses to distribute software and documentation, to accept regular contributions from individuals and corporations, and to accept larger grants of existing software products. These licenses help us achieve our goal of providing reliable and long-lived software products through collaborative open source software development. In all cases, contributors retain full rights to use their original contributions for any other purpose outside of Apache while providing the ASF and its projects the right to distribute and build upon their work within Apache.

The Apache License, Version 2.0 (current) is licensed according to the terms of the documents listed below. The 2.0 version of the Apache License was approved by the ASF in 2004. The goals of this license revision have been to reduce the number of frequently asked questions, to allow the license to be reusable without modification by any project (including non-ASF projects), to allow the license to be included by reference instead of listed in every file, to clarify the license on submission of contributions, to require a patent license on contributions that necessarily infringe the contributor's own patents, and to move comments regarding Apache and other inherited attribution notices to a location outside the license terms (the NOTICE file).

The result is a license that is supposed to be compatible with other open...
source licenses, while remaining true to the original goals of the Apache
Group and supportive of collaborative development across both nonprofit and
commercial organizations. The Apache Software Foundation is still trying to
determine if this version of the Apache License is compatible with the
GPL.</p><p>All packages produced by the ASF are implicitly licensed under the Apache
License, Version 2.0, unless otherwise explicitly stated. More developer
documentation on how to apply the Apache License to your work can be found
in Applying the Apache License, Version 2.0</p><h3 id="1.1">Apache License, Version 1.1 (historic)</h3><p>The 1.1 version of the Apache License was approved by the ASF in 2000. The
primary change from the 1.0 license is in the 'advertising clause' (section
3 of the 1.0 license); derived products are no longer required to include
attribution in their advertising materials, only in their documentation.</p><p>Individual packages licensed under the 1.1 version may have used different
wording due to varying requirements for attribution or mark identification,
but the binding terms were all the same.</p><h3 id="1.0">Apache License, Version 1.0 (historic)</h3><p>This is the original Apache License which applies only to older versions of
Apache packages (such as version 1.2 of the Web server).</p><h1 id="clas">Contributor License Agreements</h1><p>The ASF desires that all contributors of ideas, code, or documentation to
any Apache projects complete, sign, and submit (via fax or
email) an Individual Contributor License Agreement (ICLA). The purpose of this agreement is to clearly define the
terms under which intellectual property has been contributed to the ASF and
thereby allow us to defend the project should there be a legal dispute
regarding the software at some future time. A signed ICLA is required to be
on file before an individual is given commit rights to an ASF project.</p><p>For a corporation that has assigned employees to work on an Apache project,
Corporate CLA (CCLA) is available for contributing intellectual property via the corporation, that may have been assigned as
part of an employment agreement. Note that a Corporate CLA does not remove
the need for every developer to sign their own ICLA as an individual, to
cover any of their contributions which are not owned by the corporation
signing the CCLA.</p><p>The ICLA is not tied to any employer you may have, so it is recommended to
use one's personal email address in the contact details, rather than an @work
address.</p><p>Your Full name will be published unless you provide an alternative Public name.
For example if your full name is Andrew Bernard Charles Dickens, but you wish
to be known as Andrew Dickens, please enter the latter as your Public name.
The email address and other contact details are not published.

When an individual or corporation decides to donate a body of existing software or documentation to one of the Apache projects, they need to execute a formal Software Grant Agreement (SGA) with the ASF. Typically, this is done after negotiating approval with the ASF Incubator or one of the PMCs, since the ASF will not accept software unless there is a viable community available to support a collaborative project.

Documents may be submitted by fax or email.

If submitting by fax, please print, sign, and fax all pages of the document to +1-919-573-9199. Please send documents right-side-up, first page first; and send only one document per fax.

If submitting by email, please fill the form with a pdf viewer, print, sign, scan all pages into a single pdf file, and email the pdf file as an attachment to secretary@apache.org. If possible, send the attachment from the email address in the document. Please send only one document per email.

If you prefer to sign electronically, please fill the form, save it locally (e.g. icla.pdf), and sign the file by preparing a detached PGP signature. For example,

```
gpg --armor --detach-sign icla.pdf
```

The above will create a file icla.pdf.asc. Send both the file and signature as attachments in the same email to secretary@apache.org. Please send only one document (file plus signature) per email. Please do not submit your public key to Apache. Instead, please upload your public key to pgpkeys.mit.edu.

The files typically are named icla.pdf and icla.pdf.asc for individual agreements; ccla.pdf and ccla.pdf.asc for corporate agreements; software-grant.pdf and software-grant.pdf.asc for grants.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.
Community</h5>
<ul class="list-unstyled white" role="menu">
<li><a href="http://community.apache.org/">Overview</a></li>
<li><a href="/foundation/conferences.html">Conferences</a></li>
<li><a href="http://community.apache.org/newcomers">Getting Started</a></li>
<li><a href="/foundation/how-it-works.html">The Apache Way</a></li>
<li><a href="/travel/">Travel Assistance</a></li>
<li><a href="/foundation/getinvolved.html">Get Involved</a></li>
<li><a href="http://community.apache.org/newbiefaq.html">Community FAQ</a></li>
</ul>
<h5 class="white">Innovation</h5>
<ul class="list-unstyled white" role="menu">
<li><a href="http://incubator.apache.org/">Incubator</a></li>
<li><a href="http://labs.apache.org/">Labs</a></li>
<li><a href="/licenses/">Licensing</a></li>
<li><a href="/foundation/license-faq.html">Licensing FAQ</a></li>
<li><a href="/foundation/marks/">Trademark Policy</a></li>
<li><a href="/foundation/contact.html">Contacts</a></li>
</ul>
<h5 class="white">Tech Operations</h5>
<ul class="list-unstyled white" role="menu">
<li><a href="/dev/">Developer Information</a></li>
<li><a href="/dev/infrastructure.html">Infrastructure</a></li>
<li><a href="/security/">Security</a></li>
<li><a href="http://status.apache.org">Status</a></li>
<li><a href="/foundation/contact.html">Contacts</a></li>
</ul>
<h5 class="white">Press</h5>
<ul class="list-unstyled white" role="menu">
<li><a href="/press/">Overview</a></li>
</ul>
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to 
communication on electronic mailing lists, source code control systems, 
and issue tracking systems that are managed by, or on behalf of, the 
Licensor for the purpose of discussing and improving the Work, but 
excluding communication that is conspicuously marked or otherwise 
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity 
on behalf of whom a Contribution has been received by Licensor and 
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of 
this License, each Contributor hereby grants to You a perpetual, 
worldwide, non-exclusive, no-charge, royalty-free, irrevocable 
copyright license to reproduce, prepare Derivative Works of, 
publicly display, publicly perform, sublicense, and distribute the 
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of 
this License, each Contributor hereby grants to You a perpetual, 
worldwide, non-exclusive, no-charge, royalty-free, irrevocable 
(except as stated in this section) patent license to make, have made, 
use, offer to sell, sell, import, and otherwise transfer the Work, 
where such license applies only to those patent claims licensable 
by such Contributor that are necessarily infringed by their 
Contribution(s) alone or by combination of their Contribution(s) 
with the Work to which such Contribution(s) was submitted. If You 
institute patent litigation against any entity (including a 
cross-claim or counterclaim in a lawsuit) alleging that the Work 
or a Contribution incorporated within the Work constitutes direct 
or contributory patent infringement, then any patent licenses 
granted to You under this License for that Work shall terminate 
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the 
Work or Derivative Works thereof in any medium, with or without 
modifications, and in Source or Object form, provided that You 
meet the following conditions:

(a) You must give any other recipients of the Work or 
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices 
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works 
that You distribute, all copyright, patent, trademark, and 
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of 
the Derivative Works; and

d) If the Work includes a "NOTICE" text file as part of its 
distribution, then any Derivative Works that You distribute must 
include a readable copy of the attribution notices contained 
within such NOTICE file, excluding those notices that do not 
pertain to any part of the Derivative Works, in at least one 
of the following places: within a NOTICE text file distributed 
as part of the Derivative Works; within the Source form or 
documentation, if provided along with the Derivative Works; or, 
within a display generated by the Derivative Works, if and 
wherever such third-party notices normally appear. The contents 
of the NOTICE file are for informational purposes only and 
do not modify the License. You may add Your own attribution 
notices within Derivative Works that You distribute, alongside 
or as an addendum to the NOTICE text from the Work, provided 
that such additional attribution notices cannot be construed 
as modifying the License.

You may add Your own copyright statement to Your modifications and 
may provide additional or different license terms and conditions 
for use, reproduction, or distribution of Your modifications, or 
for any such Derivative Works as a whole, provided Your use, 
reproduction, and distribution of the Work otherwise complies with 
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, 
any Contribution intentionally submitted for inclusion in the Work 
by You to the Licensor shall be under the terms and conditions of 
this License, without any additional terms or conditions. 
Notwithstanding the above, nothing herein shall supersede or modify 
the terms of any separate license agreement you may have executed 
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade 
names, trademarks, service marks, or product names of the Licensor, 
except as required for reasonable and customary use in describing the 
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or 
agreed to in writing, Licensor provides the Work (and each 
Contributor provides its Contributions) on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or 
implied, including, without limitation, any warranties or conditions 
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A 
PARTICULAR PURPOSE. You are solely responsible for determining the 
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
This product currently only contains code developed by authors of specific components, as identified by the source code files; if such notes are missing files have been created by Tatu Saloranta.

For additional credits (generally to people who reported problems) see CREDITS file.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.765 path-key 2.0.1

1.765.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.766 convert-source-map 1.5.1 1.5.1

1.766.1 Available under license:

Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

1.767 protobuf-java-util 3.13.0

1.767.1 Available under license:

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Automatic-Module-Name: com.google.protobuf.util
Bnd-LastModified: 1597451582365
Build-Jdk: 11.0.7
Built-By: acozzette
Bundle-Description: Utilities for Protocol Buffers
Bundle-DocURL: https://developers.google.com/protocol-buffers/
Bundle-License: https://opensource.org/licenses/BSD-3-Clause
Bundle-ManifestVersion: 2
Bundle-Name: Protocol Buffers [Util]
Bundle-SymbolicName: com.google.protobuf.util
Bundle-Version: 3.13.0
Created-By: Apache Maven Bundle Plugin
Export-Package: com.google.protobuf.util;version="3.13.0";uses:="com.google.protobuf.util"
1.768 yargs 3.32.0
1.768.1 Available under license:

Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.769 mime-pull 1.9.6
1.769.1 Available under license:

Found license 'General Public License 2.0' in *" Copyright (c) 1997-2014 Oracle and/or its affiliates. All rights
reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2
only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'
Found license 'General Public License 2.0' in "Copyright (c) 1997-2013 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License*

Found license 'General Public License 2.0' in "Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License*

Found license 'General Public License 2.0' in "Copyright (c) 1997-2012 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License*

Found license 'General Public License 2.0' in "Copyright (c) 1997-2015 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License*

---

**1.770 chardet 0.7.0**

**1.770.1 Available under license:**

Copyright (C) 2018 Dmitry Shirokov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

Open Source Used In DNAC Platform Antelope 2526
1.771 arr-diff 4.0.0

1.771.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.772 crypto-js 4.0.0

1.772.1 Available under license :

# License

[The MIT License (MIT)](http://opensource.org/licenses/MIT)

Copyright (c) 2009-2013 Jeff Mott
Copyright (c) 2013-2016 Evan Vosberg

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Text
Copyright 2014-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

1.774 @babel/plugin-syntax-export-namespace-from 7.8.3

1.774.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.775 log4j-over-slf4j 1.7.26

1.776 twill-core 0.6.0-incubating

1.776.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 *  http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137746303_1614121688.86/0/twill-core-0-6-0-incubating-sources-2-jar/org/apache/twill/internal/kafka/client/IntegerEncoder.java
* /opt/cola/permits/1137746303_1614121688.86/0/twill-core-0-6-0-incubating-sources-2-jar/org/apache/twill/internal/ProcessController.java
* /opt/cola/permits/1137746303_1614121688.86/0/twill-core-0-6-0-incubating-sources-2-jar/org/apache/twill/internal/json/LogEntryDecoder.java
* /opt/cola/permits/1137746303_1614121688.86/0/twill-core-0-6-0-incubating-sources-2-jar/org/apache/twill/internal/logging/Loggings.java
* /opt/cola/permits/1137746303_1614121688.86/0/twill-core-0-6-0-incubating-sources-2-jar/org/apache/twill/internal/kafka/client/IntegerPartitioner.java
* /opt/cola/permits/1137746303_1614121688.86/0/twill-core-0-6-0-incubating-sources-2-
jar/org/apache/twill/internal/json/LogThrowableCodec.java

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
/ opt/ cola/ permits/ 1137746303_ 1614121688. 86/ 0/ twill-core-0-6-0-incubating-sources-2-
jar/ org/apache/twill/internal/CompositeService.java

1.777 path-exists 4.0.0
1.777.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
    distribution, then any Derivative Works that You distribute must
    include a readable copy of the attribution notices contained
    within such NOTICE file, excluding those notices that do not
    pertain to any part of the Derivative Works, in at least one
    of the following places: within a NOTICE text file distributed
    as part of the Derivative Works; within the Source form or
    documentation, if provided along with the Derivative Works; or,
    within a display generated by the Derivative Works, if and
    wherever such third-party notices normally appear. The contents
    of the NOTICE file are for informational purposes only and
    do not modify the License. You may add Your own attribution
    notices within Derivative Works that You distribute, alongside
    or as an addendum to the NOTICE text from the Work, provided
    that such additional attribution notices cannot be construed
    as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
you may add Your own permissions as the Contribution, provided
that none of the additional permissions violate the conditions
of this License. You must update the NOTICE text file in the Work
with You contributions, and must include any other necessary notices
of changes in the Source form. The above provides a method of
receiving credit for Your contributions. You may at Your option
choose to state a more restrictive license than the License, but if
You do not make a statement to the contrary, You must at least
include the notice shown in the Section 4 above (and the
Attribution Notices in any notice that You add).

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor.
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.
Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.779 punycode 1.4.1 1.4.1

1.779.1 Available under license :
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.780 ms 2.1.3

1.780.1 Available under license :
The MIT License (MIT)

Copyright (c) 2020 Vercel, Inc.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.781 expand-brackets 2.1.4
1.781.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.782 statuses 1.4.0 1.4.0
1.782.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.783 pcre 8.38-3.1

1.783.1 Available under license:

PCRE LICENCE

Please see the file LICENCE in the PCRE distribution for licensing details.

End
PCRE LICENCE

--------

PCRE is a library of functions to support regular expressions whose syntax
and semantics are as close as possible to those of the Perl 5 language.

Release 8 of PCRE is distributed under the terms of the "BSD" licence, as
specified below. The documentation for PCRE, supplied in the "doc"
directory, is distributed under the same terms as the software itself. The data
in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also
included in the distribution is a set of C++ wrapper functions, and a
just-in-time compiler that can be used to optimize pattern matching. These
are both optional features that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS
-----------------------------

Written by: Philip Hazel
Email local part: ph10
Email domain: cam.ac.uk

University of Cambridge Computing Service,

Copyright (c) 1997-2015 University of Cambridge
All rights reserved.

PCRE JUST-IN-TIME COMPILATION SUPPORT
----------------------------------------

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

Copyright(c) 2010-2015 Zoltan Herczeg
All rights reserved.

STACK-LESS JUST-IN-TIME COMPILER
---------------------------------

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

Copyright(c) 2009-2015 Zoltan Herczeg
All rights reserved.

THE C++ WRAPPER FUNCTIONS
---------------------------

Contributed by: Google Inc.

Copyright (c) 2007-2012, Google Inc.
All rights reserved.
THE "BSD" LICENCE

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

End

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
1.784 debug 3.2.6

1.784.1 Available under license:
(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.785 process-nextick-args 2.0.1

1.785.1 Available under license:

# Copyright (c) 2015 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

1.786 listener-collection 1.2.0

1.786.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Jan Blaha

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

---

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.787 @babel/helper-hoist-variables 7.13.0

1.787.1 Available under license :

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.788 netty-transport-native-epoll 4.1.59.Final

1.788.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 *
*/

Found in path(s):
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollDomainSocketChannelConfig.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/TcpMd5Util.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollEventArray.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollChannelConfig.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/AbstractEpollServerChannel.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-
/*
 * Copyright 2014 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
/**
 * Set the {@code TCP_MD5SIG} option on the socket. See {@code linux/tcp.h} for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
 */
/**
 * Set the {@code TCP_QUICKACK} option on the socket.
 * See <a href="https://linux.die.net//man/7/tcp">TCP_QUICKACK</a> for more details.
 */

Found in path(s):
* /opt/cola/permissions/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollSocketChannelConfig.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Found in path(s):

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/netty_epoll_linuxsocket.h
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/netty_epoll_linuxsocket.c
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/NativeStaticallyReferencedJniMethods.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/LinuxSocket.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 */
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
*   https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
 * Set the {code TCP_MD5SIG} option on the socket. See {code linux/tcp.h} for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
 */

Found in path(s):
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollServerSocketChannelConfig.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 *   https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/Native.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/netty_epoll_native.c
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 */
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

```
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
```

Found in path(s):
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollChannelOption.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollServerSocketChannel.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollDatagramChannel.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollSocketChannel.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollTcpInfo.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/AbstractEpollChannel.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/Epoll.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollEventLoopGroup.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/EpollEventLoop.java
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-jar/io/netty/channel/epoll/package-info.java

No license file was found, but licenses were detected in source scan.
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1136534488_1613780692.52/0/netty-transport-native-epoll-4-1-59-final-sources-
jar/io/netty/channel/epoll/EpollDatagramChannelConfig.java

1.789 librest 0.7.93-1
1.789.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations
below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.
We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.
The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a
table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.
This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms
of the ordinary General Public License).

To apply these terms, attach the following notices to the library.
It is safest to attach them to the start of each source file to most
effectively convey the exclusion of warranty; and each file should
have at least the "copyright" line and a pointer to where the full
notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or
your school, if any, to sign a "copyright disclaimer" for the library,
if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James
Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.790 base64-arraybuffer 0.1.5 0.1.5
1.790.1 Available under license:

Copyright (c) 2012 Niklas von Hertzen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.791 twill-discovery-core 0.6.0-incubating

1.791.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
1.792 gson 2.8.5

1.793 commons-beanutils 1.9.4

1.793.1 Available under license:

Apache Commons BeanUtils
Copyright 2000-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.794 netty 4.1.56.Final

1.794.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HpackDynamicTable.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HpackHuffmanDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HpackHeaderField.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HpackEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HpackUtil.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HpackDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HpackStaticTable.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HpackHuffmanEncoder.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations.

*/

Found in path(s):

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/udt/UdtServerChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/sctp/SctpChannelOption.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/oio/OioByteStreamChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/rxtx/RxtxChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/rxtx/RxtxDeviceAddress.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/rxtx/DefaultRxtxChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/rxtx/RxtxChannelOption.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/rxtx/RxtxChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-channel/rxtx/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/rxtx/RxtxChannel.java

No license file was found, but licenses were detected in source scan.

*/

Copyright 2012 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations.
* under the License.
*/

//The MIT License

//Permission is hereby granted, free of charge, to any person obtaining a copy
//of this software and associated documentation files (the "Software"), to deal
//to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
//furnished to do so, subject to the following conditions:
//The above copyright notice and this permission notice shall be included in
//all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http/websocketx/client/WebSocketClientHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version
 * 2.0 (the "License"); you may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/cors/CorsHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/cors/CorsConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http/cors/OkResponseHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketClientProtocolHandshakeHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketFrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/channel/ChannelId.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/EmptyArrays.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocketClientProtocolHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/memcache/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/concurrent/AbstractEventExecutorGroup.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/xml/XmlFrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/concurrent/ImmediateExecutor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/memcache/binary/DefaultFullBinaryMemcacheResponse.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/concurrent/DefaultPromise.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/channel/udt/UdtChannelOption.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/socks/SocksSubnegotiationVersion.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/memcache/binary/DefaultBinaryMemcacheResponse.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/ConcurrentSet.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/spdy/SpdyFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/NoOpTypeParameterMatcher.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/socksx/v5/Socks5CommandStatus.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/example/http/helloworld/HttpHelloWorldServerInitializer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
Open Source Used In DNAC Platform Antelope 2576

jar/io/netty/handler/codec/spdy/SpdyHeaderBlockJZlibEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http/helloworld/HttpHelloWorldServerHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/AbstractMemcacheObjectAggregator.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/compression/Crc32c.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/FutureListener.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/AbstractReferenceCountedByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/spdy/SpdyCodecUtil.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdySynStreamFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/socksx/SocksVersion.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/LastMemcacheContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheRequestDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/DefaultBinaryMemcacheRequest.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/AbstractBinaryMemcacheMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/LastMemcacheContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheServerCodec.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/socksx/SocksVersion.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/AbstractBinaryMemcacheMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/LastMemcacheContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/DefaultBinaryMemcacheRequest.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheRequestDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/DefaultBinaryMemcacheRequest.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheServerCodec.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/rxtx/RxtxClient.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/rxtx/RxtxClient.java
jar/io/netty/buffer/WrappedByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/spdy/SpdySynStreamFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/DefaultChannelProgressivePromise.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/memcache/binary/BinaryMemcacheRequest.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/memcache/FullMemcacheMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/DefaultMessageSizeEstimator.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/group/ChannelMatchers.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/ChannelOutboundBuffer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/spdy/SpdyHeaderBlockEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/memcache/binary/AbstractBinaryMemcacheDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/memcache/binary/BinaryMemcacheResponseDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/DefaultFutureListeners.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/memcache/MemcacheMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/memcache/DefaultLastMemcacheContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/rxtx/RxtxClientHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codecs/memcache/AbstractMemcacheObjectDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/SimpleLeakAwareByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/ChannelProgressiveFutureListener.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/spdy/DefaultSpdyRstStreamFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/spdy/DefaultSpdyStreamFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/FixedCompositeByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/spdy/SpdyVersion.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/memcache/AbstractMemcacheObjectEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/ReadOnlyByteBufferBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/AddressedEnvelope.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/PendingWrite.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/group/DefaultChannelGroup.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/spdy/DefaultSpdyGoAwayFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/ThreadPerTaskExecutor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/spdy/SpdyStreamStatus.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/DefaultAddressedEnvelope.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/DefaultProgressivePromise.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/socks/SocksRequestType.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/socks5/Socks5CommandType.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/code/spdy/SpdyHeaders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/DefaultThreadFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/AbstractEventExecutor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/group/ChannelGroup.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/socket/oio/DefaultOioSocketChannelConfig.java

Open Source Used In DNAC Platform Antelope 2578
Open Source Used in DNAC Platform Antelope 2579
Open Source Used In DNAC Platform Antelope 2581
/\n* Copyright 2019 The Netty Project
*\n* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*\n* https://www.apache.org/licenses/LICENSE-2.0
*\n* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
// (BSD License: https://www.opensource.org/licenses/bsd-license)
// All rights reserved.
// Redistribution and use in source and binary forms, with or
// * Redistributions of source code must retain the above
// copyright notice, this list of conditions and the
// following disclaimer.
// * Redistributions in binary form must reproduce the above
// following disclaimer in the documentation and/or other
// * Neither the name of the Webbit nor the names of

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocket07FrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocket13FrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http/websocketx/WebSocket08FrameDecoder.java

No license file was found, but licenses were detected in source scan.

/\n* Copyright 2020 The Netty Project
*\n* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*\n* https://www.apache.org/licenses/LICENSE-2.0
*\n* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

Open Source Used In DNAC Platform Antelope 2582
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/KmpSearchProcessorFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/example/http2/helloworld/frame/client/Http2ClientFrameInitializer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/example/http2/helloworld/frame/client/Http2Client.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/AbstractMultiSearchProcessorFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/BitapSearchProcessorFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/AhoCorasicSearchProcessorFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/SearchProcessorFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/example/http2/helloworld/frame/client/Http2ClientStreamFrameResponseHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/AbstractSearchProcessorFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/MultiSearchProcessor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/SearchProcessor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/search/MultiSearchProcessorFactory.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *  http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2020 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
Open Source Used In DNAC Platform Antelope

Antelope 2585

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/haproxy/HAPProxyServer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/channel/StacklessClosedChannelException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/multipart/DeleteFileOnExitHook.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/mqtt/MqttPubReplyMessageVariableHeader.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/stomp/websocket/StompWebSocketChatServerInitializer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshakeException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/SslClosedEngineException.java

- FingerprintTrustManagerFactoryBuilder.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

// https://tools.ietf.org/html/rfc7540#section-8.1.2.2 makes a special exception for TE

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HttpConversionUtil.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 *
Open Source Used In DNAC Platform Antelope 2588

```java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/resolver/dns/PreferredAddressTypeComparator.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/websocketx/WebSocket00FrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/TrustManagerFactoryWrapper.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/websocketx/CloseWebSocketFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/websocketx/CorruptedWebSocketFrameException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/KeyManagerFactoryWrapper.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/mqtt/heartBeat/MqttHeartBeatBroker.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/mqtt/heartBeat/MqttHeartBeatClientHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/internal/svm/CleanerJava6Substitution.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/mqtt/heartBeat/MqttHeartBeatClient.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/dns/DnsQueryEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/resolver/dns/DatagramDnsQueryContext.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/internal/svm/PlatformDependent0Substitution.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/MaxCapacityQueue.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/mqtt/heartBeat/MqttHeartBeatBrokerHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/Http2SettingsAckFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/channel/ExtendedClosedChannelException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/resolver/dns/macOS/DnsResolver.java
```
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/unix/Buffer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/OpenSslTlsv13X509ExtendedTrustManager.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/AuthoritativeDnsServerCache.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/OpenSslPrivateKey.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/NoopAuthoritativeDnsServerCache.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DirContextUtils.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DnsAddressResolveContext.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DefaultAuthoritativeDnsServerCache.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DnsCnameCache.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/OpenSslSession.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DnsRecordResolveContext.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DefaultAuthoritativeDnsServerCache.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/SimpleUserEventChannelHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/channel/socket/nio/NioChannelOption.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/SslCertificate.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/SignatureAlgorithmConverter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslKeyMaterial.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslX509TrustManagerWrapper.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslKeyMaterialProvider.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslKeyMaterial.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/ExtendedOpenSslSession.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/resolver/dns/Cache.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslCachingKeyMaterialProvider.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/resolver/dns/NoopDnsCnameCache.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslCachingX509KeyManagerFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/resolver/dns/DefaultDnsCnameCache.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/internal/SuppressJava6Requirement.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/resolver/dns/DnsAddressDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/channel/unix/PreferredDirectByteBufAllocator.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslX509KeyManagerFactory.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 */
* under the License.
*/
/**
 * Set the `{code TCP_MD5SIG}` option on the socket. See `{code linux/tcp.h}` for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/epoll/EpollServerSocketChannelConfig.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *  https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/extensions/compression/WebSocketServerCompressionHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/mqtt/MqttDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/JdkSslEngine.java
jar/io/netty/example/spdy/server/SpdyServer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/websockets/extensions/WebSocketClientExtensionHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/PemReader.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/epoll/EpollTcpInfo.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanStageEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/JettyAlpnSslEngine.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/haproxy/HAPerMessageDeflateClientExtensionHandshaker.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateClientExtensionHandshaker.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/util/SimpleTrustManagerFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/spdy/server/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/websockets/extensions/compression/PerMessageDeflateServerExtensionHandshaker.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/mqtt/MqttEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/xml/XmlElementStart.java

Open Source Used In DNAC Platform Antelope 2597
Open Source Used In DNAC Platform Antelope 2598
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

Open Source Used In DNAC Platform Antelope 2604
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/PooledHeapByteBuf.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

// (BSD License: https://www.opensource.org/licenses/bsd-license)
// All rights reserved.
// Redistribution and use in source and binary forms, with or
// Redistributions of source code must retain the above
// copyright notice, this list of conditions and the
// following disclaimer.
// * Redistributions in binary form must reproduce the above
// following disclaimer in the documentation and/or other
// * Neither the name of the Webbit nor the names of

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket13FrameEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket07FrameEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket08FrameEncoder.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/HttpServerUpgradeHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/Http2ConnectionHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/DefaultHttp2ConnectionEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/collection/ShortObjectMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/Http2DataWriter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/collection/CharObjectMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/DefaultHeaders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/collection/LongObjectHashMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/DefaultHttpHeaders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/collection/IntObjectHashMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/collection/ShortObjectHashMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/ByteObjectMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/Http2FrameAdapter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2ClientUpgradeCodec.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http2/helloworld/client/Http2Client.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/collection/LongCollections.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/collection/IntCollections.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/collection/CharObjectHashMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2Exception.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2FrameReader.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2FrameSizePolicy.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2Stream.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2LocalFlowController.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/HttpToHttp2ConnectionHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2FrameReader.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http2/Http2ExampleUtil.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/collection/CharCollections.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/collection/ShortCollections.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/EmptyHeaders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2FrameListener.java
Open Source Used In DNAC Platform Antelope

- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2FrameTypes.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionAdapter.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http2/helloworld/server/Http2OrHttpHandler.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2Error.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2Settings.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionDecoder.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2Flags.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2SettingsHandler.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2ClientInitializer.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2EventAdapter.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionDecoder.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2Error.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2Settings.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionDecoder.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2Flags.java
- /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2SettingsHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
/*
 * Copyright 2020 The Netty Project
 */
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/pcap/TCPPacket.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/pcap/IPPacket.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/pcap/UDPPacket.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/pcap/EthernetPacket.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/pcap/PcapHeaders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/pcap/PcapWriter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/pcap/PcapWriteHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/pcap/package-info.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2012 The Netty Project
* *
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* https://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/GlobalTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/GlobalTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/traffic/GlobalChannelTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

/*
   Stomp Over WebSocket http://www.jmesnil.net/stomp-websocket/doc/ | Apache License V2.0
   Copyright (C) 2010-2013 [Jeff Mesnil](http://jmesnil.net/)
   Copyright (C) 2012 [FuseSource, Inc.](https://fusesource.com)
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/example/stomp/websocket/stomp.js
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/traffic/AbstractTrafficShapingHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/channel/sctp/SctpServerChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/channel/sctp/NioSctpChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/channel/sctp/NioSctpServerChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/channel/sctp/DefaultSctpChannelConfig.java

Open Source Used In DNAC Platform  Antelope  2612
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/sctp/SctpServerChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/sctp/SctpChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/sctp/SctpNotificationHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/sctp/SctpMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/sctp/SctpChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/worldclock/WorldClockProtocol.proto
jar/io/netty/example/redis/RedisClient.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/RedisMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/RedisConstants.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http2/helloworld/multiplex/server/HelloWorldHttp2Handler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http2/helloworld/frame/server/HelloWorldHttp2Handler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/RedisCodecException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/DefaultBulkStringRedisContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/BulkStringHeaderRedisMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/IntegerRedisMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http2/helloworld/multiplex/server/Http2OrHttpHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/ErrorRedisMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http2/helloworld/frame/server/Http2OrHttpHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/LastBulkStringRedisContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/ArrayRedisMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/RedisEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/AbstractStringRedisMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/RedisMessagePool.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/SimpleStringRedisMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/FullBulkStringRedisMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/BulkStringRedisContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/RedisDecoder.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2004-2011 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):

/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/logging/FormattingTuple.java

/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/logging/MessageFormatter.java

No license file was found, but licenses were detected in source scan.

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/channel/DefaultChannelHandlerContext.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/internal/CleanerJava6.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/UnsafeHeapSwappedByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/concurrent/FastThreadLocalThread.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/UnsafeDirectSwappedByteBuf.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

/* Adaptation of https://bjoern.hoehrmann.de/utf-8/decoder/dfa/
*
* Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>
*
* Permission is hereby granted, free of charge, to any person obtaining a copy of this software
* and associated documentation files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge, publish, distribute,
* sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
* The above copyright notice and this permission notice shall be included in all copies or
* substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING
* BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE
FOR ANY CLAIM,
* DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4.1.56-final-sources-jar/io/netty/handler/codec/http/websocketx/Utf8Validator.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/SSL.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/SSLContext.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/Buffer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/Library.java

No license file was found, but licenses were detected in source scan.

*/

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/shaded/org/jctools/maps/NonBlockingSetInt.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/shaded/org/jctools/util/UnsafeLongArrayAccess.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/shaded/org/jctools/queues/IndexedQueueSizeUtil.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/AtomicReferenceArrayQueue.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SpscGrowableAtomicArrayQueue.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/SpscAtomicArrayQueue.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/MpscUnboundedAtomicArrayQueue.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/shaded/org/jctools/queues/atomic/MpscArrayQueue.java
Open Source Used In DNAC Platform Antelope 2620
/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 */
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* Copyright (c) 2004-2011 QOS.ch
* All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/internal/logging/JdkLogger.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/internal/logging/Log4JLogger.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/internal/logging/CommonsLogger.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/internal/logging/InternalLogger.java
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/META-INF/maven/io.netty/netty-all/pom.xml

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/sctp/DefaultSctpServerChannelConfig.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/resolver/dns/DnsServerAddressStreamProvider.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SniCompletionEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SslCompletionEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SslCloseCompletionEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/OcspClientHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/OcspResponse.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/SSLHandshakeCompleteEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handle...
jar/io/netty/resolver/dns/MultiDnsServerAddressStreamProvider.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/NettyRuntime.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/ObjectCleaner.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/ConscryptAlpnSslEngine.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DefaultDnsServerAddressStreamProvider.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DnsCacheEntry.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/nio/SelectedSelectionKeySetSelector.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/socket/oio/OioDatagramChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/logging/LocationAwareSlf4JLogger.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2StreamChannelBootstrap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/Ocsp/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/HttpServerExpectContinueHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/HttpBuffers.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/SslClientHelloHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DnsServerAddressStreamProviders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/DnsQueryLifecycleObserverFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/internal/ChannelUtils.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/unix/UnixChannelUtil.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2StreamChannelId.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/NoopDnsQueryLifecycleObserver.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/MultiplexCodecBuilder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/ReadOnlyHttpHeaders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/mqtt/MqttMessageBuilders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/resolver/dns/LoggingDnsQueryLifecycleObserver.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/socket/ChannelInputShutdownReadComplete.java
No license file was found, but licenses were detected in source scan.

*/
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/PriorityQueue.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/bootstrap/FailedChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/SSLPrivateKeyMethodTask.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/SSLTask.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/SSLPrivateKeyMethodDecryptTask.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/CertificateCallbackTask.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/CertificateVerifierTask.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/SSLPrivateKeyMethodSignTask.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/internal/tcnative/SSLPrivateKeyMethod.java

No license file was found, but licenses were detected in source scan.

*/
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/proxy/HexDumpProxyBackendHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/qotm/QuoteOfTheMomentServer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/CorruptedFrameException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/DefaultEventExecutor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/timeout/ReadTimeoutHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/socket/ChannelInputShutdownEvent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/rtsp/RtspRequestDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/util/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/serialization/WeakReferenceMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/ssl/util/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/multipart/FileUpload.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/snoop/HttpSnoopClientInitializer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/compression/DecompressionException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/base64/package-info.java
Open Source Used In DNAC Platform Antelope 2630
Open Source Used In DNAC Platform Antelope 2631
Open Source Used In DNAC Platform Antelope

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/IllegalReferenceCountException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/securechat/SecureChatClientInitializer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/ThreadPerChannelEventLoopGroup.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/udt/echo/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/socksproxy/SocksServerConnectHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http/file/HttpStaticFileServerHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/sctp/oio/OioSctpServerChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/PooledDirectByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/socket/ServerSocketChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostStandardRequestDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/PooledByteBufAllocator.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/udt/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/DefaultChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/rtsp/RtspResponseStatuses.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/MessageToMessageCodec.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/HttpMethod.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/CodecException.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/bytes/ByteArrayEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/marshalling/DefaultMarshallerProvider.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5InitialRequest.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/echo/EchoClientHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/EventExecutor.java
Open Source Used In DNAC Platform Antelope 2633

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/serialization/ObjectEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/factorial/BigIntegerDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/echo/EchoClient.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/CompleteChannelFuture.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http/upload/HttpUploadServerInitializer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/websocketx/PingWebSocketFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/FixedLengthFrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/ReplayingDecoderByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/spdy/spdy/SpdySession.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/socket/DatagramChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/local/LocalChannelRegistry.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/socket/ DatagramChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/ChannelPromise.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/file/FileServer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/udt/echo/bytes/ByteEchoClient.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/factorial/FactorialServerHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/channel/epoll/EpollDatagramChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/QueryStringDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/telnet/TelnetClientHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/marshalling/ThreadLocalMarshallerProvider.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/concurrent/MultithreadEventExecutorGroup.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker00.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/channel/socket/ServerSocketChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/Delimiters.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/socks/SocksInitRequest.java
jar/jo/netty/channel/ServerChannel.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/marshalling/package-info.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/channel/DefaultEventLoop.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/buffer/ByteBufInputStream.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/http/multipart/AbstractDiskHttpData.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/sctp/ScpOutboundByteStreamHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/rtsp/package-info.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/http/DefaultHttpContent.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksx/v5/Socks5PasswordAuthRequest.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/http/HttpContentDecoder.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksx/v5/Socks5PasswordAuthRequest.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/string/StringEncoder.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerInitializer.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/marshalling/MarshallingDecoder.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/http/HttpContentDecoder.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerInitializer.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
* /opt/cola/perms/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources- 
  jar/jo/netty/handler/codec/socksproxy/SocksServerHandler.java
Open Source Used in DNAC Platform Antelope 2638
Open Source Used In DNAC Platform Antelope 2639
Open Source Used In DNAC Platform Antelope 2642

jar/io/netty/handler/codec/compression/JZlibEncoder.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http/WebSocketUtil.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/example/securechat/SecureChatServerInitializer.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http/package-info.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/example/http/websocketx/server/WebSocketServer.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/marshalling/MarshallerProvider.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/example/http/websocketx/server/WebSocketIndexPageHandler.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/UnsuportedMessageTypeException.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/channel/io/oio/AbstractOioMessageChannel.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/UnsupportedMessageTypeError.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/example/http/snoop/HttpSnoopServerInitializer.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/channel/ChannelPipelineException.java

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/channel/oio/AbstractOioMessageChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/concurrent/GlobalEventExecutor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/local/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/AttributeMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/proxy/HexDumpProxyInitializer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/localecho/LocalEchoClientHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/HttpContentEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/socks/SocksMessage.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/udt/nio/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/DelimiterBasedFrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshakerFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/LastHttpContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/udt/echo/rendezvousBytes/ByteEchoPeerHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/worldclock/WorldClockServer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpHeaders.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/MessageAggregator.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandResponse.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/PooledByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/MessageToMessageEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/http/upload/HttpUploadServer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/socket/SocketChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/socks/SocksCmdResponseDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/multipart/MemoryFileUpload.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/Constant.java
Open Source Used In DNAC Platform Antelope 2645

* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/sctp/SctpEchoClient.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/DefaultCookie.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/DefaultAttributeMap.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/compression/ZlibEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/rtsp/RtspRequestEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/socks/SocksCmdResponse.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/nio/AbstractNioChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/group/ChannelGroupFutureListener.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/marshalling/ChannelBufferByteInput.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/buffer/CompositeByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/compression/SnappyFrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/timeout/WriteTimeoutHandler.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthResponse.java
Open Source Used In DNAC Platform Antelope 2648
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
 * A decoder that splits the received {@link ByteBuf}s dynamically by the
 * value of the length field in the message. It is particularly useful when you
 * decode a binary message which has an integer header field that represents the
 * length of the message body or the whole message.
 * <p>
 * {@link LengthFieldBasedFrameDecoder} has many configuration parameters so
 * that it can decode any message with a length field, which is often seen in
 * proprietary client-server protocols. Here are some example that will give
 * you the basic idea on which option does what.
 * <h3>2 bytes length field at offset 0, do not strip header</h3>
 * <p>
 * The value of the length field in this example is <tt>12 (0x0C)</tt> which
 * represents the length of "HELLO, WORLD". By default, the decoder assumes
 * that the length field represents the number of the bytes that follows the
 * length field. Therefore, it can be decoded with the simplistic parameter
 * combination.
 * <pre>
 * <b>lengthFieldOffset</b>   = <b>0</b>
 * <b>lengthFieldLength</b>   = <b>2</b>
 * lengthAdjustment    = 0
 * initialBytesToStrip = 0 (= do not strip header)
 * </pre>
 * 
 * BEFORE DECODE (14 bytes)      AFTER DECODE (14 bytes)
 * +--------+----------------+----->+--------+----------------+
 * | Length | Actual Content |---->| Length | Actual Content |
 * | 0x000C | "HELLO, WORLD" |   | 0x000C | "HELLO, WORLD" |
 * +--------+----------------+----->+--------+----------------+
 * 
 * <h3>2 bytes length field at offset 0, strip header</h3>
 * <p>
 * Because we can get the length of the content by calling
 * {@link ByteBuf#readableBytes()}, you might want to strip the length
 * field by specifying <tt>initialBytesToStrip</tt>. In this example, we
 * specified <tt>2</tt>, that is same with the length of the length field, to
 * strip the first two bytes.
 * <pre>
 * lengthFieldOffset = 0
 * lengthFieldLength = 2
In most cases, the length field represents the length of the message body only, as shown in the previous examples. However, in some protocols, the length field represents the length of the whole message, including the message header. In such a case, we specify a non-zero lengthAdjustment. Because the length value in this example message is always greater than the body length by 2, we specify -2 as compensation.

```
lengthFieldOffset   =  0
lengthFieldLength   =  2
lengthAdjustment    = 0
initialBytesToStrip = 0
```

```
BEFORE DECODE (14 bytes)      AFTER DECODE (14 bytes)
+--------+----------------+      +--------+----------------+
| Length | Actual Content |----->| Length | Actual Content |
| 0x000C | "HELLO, WORLD" |      | 0x000E | "HELLO, WORLD" |
+--------+----------------+      +--------+----------------+
</pre>

3 bytes length field at the end of 5 bytes header, do not strip header

```
<tt>lengthFieldOffset</tt>   = <b>2</b> (= the length of Header 1)
<tt>lengthFieldLength</tt>   = <b>3</b>
<tt>lengthAdjustment</tt>    = 0
<tt>initialBytesToStrip</tt> = 0
```

```
BEFORE DECODE (17 bytes)      AFTER DECODE (17 bytes)
+----------+----------+----------------+      +----------+----------+----------------+
| Header 1 |  Length  | Actual Content |----->| Header 1 |  Length  | Actual Content |
+----------+----------+----------------+      +----------+----------+----------------+
</pre>

The following message is a simple variation of the first example. An extra header value is prepended to the message. <tt>lengthAdjustment</tt> is zero again because the decoder always takes the length of the prepended data into account during frame length calculation.
This is an advanced example that shows the case where there is an extra header between the length field and the message body. You have to specify a positive \texttt{lengthAdjustment} so that the decoder counts the extra header into the frame length calculation.

This is a combination of all the examples above. There are the prepended header before the length field and the extra header after the length field. The prepended header affects the \texttt{lengthFieldOffset} and the extra header affects the \texttt{lengthAdjustment}. We also specified a non-zero \texttt{initialBytesToStrip} to strip the length field and the prepended header from the frame. If you don't want to strip the prepended header, you could specify \texttt{0} for \texttt{initialBytesToSkip}.

This is a combination of all the examples above. There are the prepended header before the length field and the extra header after the length field. The prepended header affects the \texttt{lengthFieldOffset} and the extra header affects the \texttt{lengthAdjustment}. We also specified a non-zero \texttt{initialBytesToStrip} to strip the length field and the prepended header from the frame. If you don't want to strip the prepended header, you could specify \texttt{0} for \texttt{initialBytesToSkip}.
Let's give another twist to the previous example. The only difference from the previous example is that the length field represents the length of the whole message instead of the message body, just like the third example. We have to count the length of HDR1 and Length into \texttt{lengthAdjustment}. Please note that we don't need to take the length of HDR2 into account because the length field already includes the whole header length.

| lengthFieldOffset | 1 |
| lengthFieldLength | 2 |
| \texttt{lengthAdjustment} | \texttt{-3} (= the length of HDR1 + LEN, negative) |
| \texttt{initialBytesToStrip} | 3 |

\begin{verbatim}
\textbf{BEFORE DECODE (16 bytes)}
\textbf{AFTER DECODE (13 bytes)}
\end{verbatim}

| HDR1 | Length | HDR2 | Actual Content |
| 0xCA | 0x0010 | 0xFE | "HELLO, WORLD" |
| 0xCE | 0x0010 | 0xFE | "HELLO, WORLD" |

*/

---

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

---

No license file was found, but licenses were detected in source scan.
Open Source Used In DNAC Platform Antelope 2656
jar/io/netty/channel/unix/ServerDomainSocketChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/protobuf/ProtobufVarint32FrameDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/HttpToHttp2ConnectionHandlerBuilder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/example/sctp/multihoming/SctpMultiHomingEchoServer.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/dns/DnsRawRecord.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/dns/DnsQuery.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/dns/DnsSection.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/ApplicationProtocolAccessor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslJavaxX509Certificate.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslJavaxX509Certificate.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslJavaxX509Certificate.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslJavaxX509Certificate.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslJavaxX509Certificate.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/OpenSslJavaxX509Certificate.java
Open Source Used In DNAC Platform Antelope 2659
/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
/**
 * Set the { @code TCP_MD5SIG} option on the socket. See { @code linux/tcp.h} for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
 */
/**
 * Set the { @code TCP_QUICKACK} option on the socket.
 * See <a href="https://linux.die.net//man/7/tcp">TCP_QUICKACK</a>
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/epoll/EpollSocketChannelConfig.java
No license file was found, but licenses were detected in source scan.

*/

* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
* *
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2EmptyDataFrameListener.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2ControlFrameLimitEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2EmptyDataFrameConnectionDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http2/Http2SettingsReceivedConsumer.java
No license file was found, but licenses were detected in source scan.

*/

* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

* Handles an I/O event or intercepts an I/O operation, and forwards it to its next handler in
* its [@link ChannelPipeline].

* <h3>Sub-types</h3>
* <p>
* [@link ChannelHandler] itself does not provide many methods, but you usually have to implement one of its
* subtypes:
* <ul>
* <li>[@link ChannelInboundHandler] to handle inbound I/O events, and</li>
* <li>[@link ChannelOutboundHandler] to handle outbound I/O operations.</li>
* </ul>
* </p>
* <p>
* Alternatively, the following adapter classes are provided for your convenience:
* <ul>
* <li>[@link ChannelInboundHandlerAdapter] to handle inbound I/O events,</li>
* <li>[@link ChannelOutboundHandlerAdapter] to handle outbound I/O operations, and</li>
* <li>[@link ChannelDuplexHandler] to handle both inbound and outbound events</li>
* </ul>
* </p>
* <p>
* For more information, please refer to the documentation of each subtype.
* </p>

* * <h3>The context object</h3>
* <p>
* A [@link ChannelHandler] is provided with a [@link ChannelHandlerContext] object. A [@link ChannelHandler] is supposed to interact with the
* [@link ChannelPipeline] it belongs to via a context object. Using the context object, the [@link ChannelHandler] can pass events upstream or
* downstream, modify the pipeline dynamically, or store the information
* (using [@link AttributeKey]s) which is specific to the handler.
* </p>

* * <h3>State management</h3>
* 
* A [@link ChannelHandler] often needs to store some stateful information.
* The simplest and recommended approach is to use member variables:
* <pre>
* public interface Message {
*     // your methods here
* }
* </pre>

* public class DataServerHandler extends [@link SimpleChannelInboundHandler]&lt;Message&gt; { 
* 
*     <b>private boolean loggedIn;</b>
* 
* }
Because the handler instance has a state variable which is dedicated to
one connection, you have to create a new handler instance for each new
channel to avoid a race condition where a unauthenticated client can get
the confidential information:

```java
public class DataServerInitializer extends ChannelInitializer<Channel> {
    public void initChannel(Channel channel) {
        channel.pipeline().addLast("handler", new DataServerHandler());
    }
}
```

**Using @link AttributeKey**s

Although it's recommended to use member variables to store the state of a
handler, for some reason you might not want to create many handler instances.
In such a case, you can use `AttributeKey` s which is provided by

```java
public interface Message {
    // your methods here
}
```

```java
public class DataServerHandler extends SimpleChannelInboundHandler<Message> {
    private final AttributeKey<Thread> auth = AttributeKey.valueOf("auth");
}
```
* { @code @Override}
* public void channelRead({@link ChannelHandlerContext} ctx, Message message) {
*     { @link Attribute} &lt;{ @link Boolean} &gt; attr = ctx.attr(auth);
*     if (message instanceof LoginMessage) {
*         authenticate((LoginMessage) o);
*         &lt;b&gt;attr.set(true)&lt;/b&gt;;
*     } else (message instanceof GetDataMessage) {
*         if (&lt;b&gt;Boolean.TRUE.equals(attr.get())&lt;/b&gt;) {
*             ctx.writeAndFlush(fetchSecret((GetDataMessage) o));
*         } else {
*             fail();
*         }
*     }
*     ...
* }
* &lt;/pre&gt;
* Now that the state of the handler is attached to the { @link ChannelHandlerContext}, you can add the
* same handler instance to different pipelines:
* &lt;pre&gt;
* public class DataServerInitializer extends { @link ChannelInitializer} &lt;{ @link Channel} &gt; {
*     
*     private static final DataServerHandler &lt;b&gt;SHARED&lt;/b&gt; = new DataServerHandler();
*     
*     { @code @Override}
*     public void initChannel({ @link Channel} channel) {
*         channel.pipeline().addLast("handler", &lt;b&gt;SHARED&lt;/b&gt;);
*     }
* &lt;/pre&gt;
* 
* 
* &lt;h4&gt;The { @code @Sharable} annotation&lt;/h4&gt;
* &lt;p&gt;
* In the example above which used an { @link AttributeKey},
* you might have noticed the { @code @Sharable} annotation.
* &lt;p&gt;
* If a { @link ChannelHandler} is annotated with the { @code @Sharable}
* annotation, it means you can create an instance of the handler just once and
* add it to one or more { @link ChannelPipeline}s multiple times without
* a race condition.
* &lt;p&gt;
* If this annotation is not specified, you have to create a new handler
* instance every time you add it to a pipeline because it has unshared state
* such as member variables.
* &lt;p&gt;
* This annotation is provided for documentation purpose, just like
* 
* <h3>Additional resources worth reading</h3>
* <p>
* Please refer to the [@link ChannelHandler], and
* [@link ChannelPipeline] to find out more about inbound and outbound operations,
* what fundamental differences they have, how they flow in a pipeline, and how to handle
* the operation in your application.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/channel/ChannelHandler.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

/*
* Written by Robert Harder and released to the public domain, as explained at
* https://creativecommons.org/licenses/publicdomain
*/

/**
* Utility class for [@link ByteBuf] that encodes and decodes to and from
* <a href="https://en.wikipedia.org/wiki/Base64">Base64</a> notation.
* 
* The encoding and decoding algorithm in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/"">Robert Harder's Public Domain
* Base64 Encoder/Decoder</a>.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/base64/Base64.java

No license file was found, but licenses were detected in source scan.
/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version
 * 2.0 (the "License"); you may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/ocsp/OcspRequestBuilder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/ocsp/OcspClientExample.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/ocsp/OcspServerExample.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/ocsp/Digester.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/example/ocsp/OcspUtils.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

/*
 * Written by Robert Harder and released to the public domain, as explained at
 * https://creativecommons.org/licenses/publicdomain
 */
/**
 * Enumeration of supported Base64 dialects.
 * <p>
 * The internal lookup tables in this class has been derived from
 * <a href="http://iharder.sourceforge.net/current/java/base64/"">Robert Harder's Public Domain
 * Base64 Encoder/Decoder</a>
 * */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/base64/Base64Dialect.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version
 * 2.0 (the "License"); you may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/flow/FlowControlHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/ByteProcessorUtils.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/util/internal/ResourcesUtil.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/redis/InlineCommandRedisMessage.java
No license file was found, but licenses were detected in source scan.

*/

* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version
* 2.0 (the "License"); you may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-jar/io/netty/handler/codec/http/cors/CorsConfigBuilder.java
No license file was found, but licenses were detected in source scan.

*/

* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
Open Source Used In DNAC Platform Antelope 2670

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/internal/tcnative/CertificateRequestedCallback.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/internal/tcnative/SessionTicketKey.java

No license file was found, but licenses were detected in source scan.

/
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/ByteProcessor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/UniformStreamByteDistributor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/DecoratingHttp2ConnectionDecoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/DefaultHeadersImpl.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/StreamByteDistributor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/AbstractInboundHttp2ToHttpAdapterBuilder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/StreamBufferingEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/WeightedFairQueueByteDistributor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/DecoratingHttp2FrameWriter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/Http2PromisedRequestVerifier.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http2/DecoratingHttp2ConnectionEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/ValueConverter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http2/InboundHttp2ToHttpAdapterBuilder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http2/Http2StreamVisitor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/CharSequenceValueConverter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/channel/CoalescingBufferQueue.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/MathUtil.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/CleanerJava9.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/concurrent/FastThreadLocalRunnable.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/Cleaner.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2020 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/internal/tcnative/SSLSessionCache.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/internal/tcnative/SSLSession.java
No license file was found, but licenses were detected in source scan.

*/
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/http/websocketx/WebSocketChunkedInput.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/concurrent/PromiseCombiner.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/ssl/PemEncoded.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/util/IntSupplier.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/smtp/package-info.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/smtp/package-info.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/kqueue/AcceptFilter.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/haproxy/HAProxySSLTLV.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/DatagramPacketDecoder.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/channel/ChannelOutboundInvoker.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/channel/kqueue/AcceptFilter.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/haproxy/HAProxySSLTLV.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/handler/codec/DatagramPacketDecoder.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources.jar/io/netty/channel/ChannelOutboundInvoker.java
*/opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
Open Source Used In DNAC Platform Antelope 2673
jar/io/netty/handler/codec/dns/DefaultDnsPtrRecord.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/DomainNameMappingBuilder.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/channel/kqueue/KQueueServerChannelConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/bootstrap/ServerBootstrapConfig.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/Http2MultiplexCodec.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/InboundHttpToHttp2Adapter.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/concurrent/OrderedEventExecutor.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/OpenSslKeyMaterialManager.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/PemPrivateKey.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/Http2DataFrame.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/internal/logging/Log4J2LoggerFactory.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/channel/unix/SocketWritableByteChannel.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/internal/MacAddressUtil.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/ReferenceCountedOpenSslServerContext.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/flush/package-info.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/ssl/PemX509Certificate.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/channel/kqueue/KQueueStaticallyReferencedJniMethods.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/ReadOnlyHttp2Headers.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/buffer/WrappedUnpooledUnsafeDirectByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/util/concurrent/UnaryPromiseNotifier.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/smtp/DefaultLastSmtpContent.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/smtp/SmtpCommand.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/resolve/RoundRobinInetAddressResolver.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/Http2FrameStream.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/smtp/SmtpCommand.java
* /opt/cola/permits/1119092601_1614145555.72/0/netty-all-4-1-56-final-sources-
  jar/io/netty/handler/codec/http2/Http2FrameStream.java
Open Source Used In DNAC Platform Antelope 2674
jar/io/netty/handler/codec/dns/DnsOptPseudoRecord.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/ssl/ReferenceCountedOpenSslContext.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/channel/kqueue/AbstractKQueueChannel.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/SocketUtils.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/util/concurrent/NonStickyEventExecutorGroup.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http2/DefaultHttp2SettingsFrame.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/ThrowableUtil.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/logging/Log4J2Logger.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/smtp/SmtpRequestEncoder.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/smtp/SmtpResponse.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/ssl/Java8SslUtils.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/UnstableApi.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/util/internal/OutOfDirectMemoryError.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/channel/kqueue/Native.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/channel/socket/DuplexChannel.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/channel/PreferHeapByteBufAllocator.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http2/Http2PingFrame.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http2/DefaultHttp2WindowUpdateFrame.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/haproxy/HAProxyTLV.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/channel/socket/DuplexChannel.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/channel/PreferHeapByteBufAllocator.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http2/Http2PingFrame.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/resolver/dns/DnsCache.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/buffer/AbstractPooledDerivedByteBuf.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
jar/io/netty/handler/codec/http2/DefaultHttp2WindowUpdateFrame.java
* /opt/cola/permits/1119092601_1614145555.72/netty-all-4-1-56-final-sources-
1.795 metrics-jetty 4.1.0

1.796 jansi 1.18

1.796.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark,
and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute
must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices
that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text
file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the
Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices
normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the
License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an
addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license
terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works
as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions
stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for
inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any
additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of
any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product
names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and
reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work
(and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR
PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and
assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or
otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing,
shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or
consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of
such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may
choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.797 parquet-encoding 1.10.0

1.797.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

------------------------------------------------------------------------------------------------------------------------

This project includes code from Daniel Lemire's JavaFastPFOR project in this binary artifact. The "Lemire" bit packing classes produced by parquet-generator are derived from the JavaFastPFOR project.
1.798 bson 1.0.9 1.0.9

1.798.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate.
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include...
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.799 chai-http 3.0.0
1.799.1 Available under license :
MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
1.800 snapdragon-util 3.0.1

1.800.1 Available under license:
The MIT License (MIT)

Copyright (c) 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.801 type-detect 4.0.8

1.801.1 Available under license:
Copyright (c) 2013 Jake Luer <jake@alogicalparadox.com> (http://alogicalparadox.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.802 is-descriptor 1.0.2
1.802.1 Available under license :
   The MIT License (MIT)

   Copyright (c) 2015-2017, Jon Schlinkert.

   Permission is hereby granted, free of charge, to any person obtaining a copy
   of this software and associated documentation files (the "Software"), to deal
   in the Software without restriction, including without limitation the rights
   to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
   copies of the Software, and to permit persons to whom the Software is
   furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in
   all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
   IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
   FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
   AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
   LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
   OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
   THE SOFTWARE.

1.803 unpipe 1.0.0 1.0.0
1.803.1 Available under license :
   (The MIT License)

   Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

   Permission is hereby granted, free of charge, to any person obtaining
   a copy of this software and associated documentation files (the
   'Software'), to deal in the Software without restriction, including
   without limitation the rights to use, copy, modify, merge, publish,
   distribute, sublicense, and/or sell copies of the Software, and to
   permit persons to whom the Software is furnished to do so, subject to
   the following conditions:

   The above copyright notice and this permission notice shall be
   included in all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND.
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

**1.804 path-root-regex 0.1.2**

**1.804.1 Available under license:**
The MIT License (MIT)

Copyright (c) 2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

---

**1.805 semver 5.7.1**

**1.805.1 Available under license:**
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR

---

Open Source Used In DNAC Platform Antelope 2693
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.806 mime-db 1.47.0 1.47.0
1.806.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.807 reflectasm 1.10.1
1.807.1 Available under license:

No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2008, Nathan Sweet
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0. To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.809 mimic-fn 2.1.0
1.809.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.810 jetty-webapp 9.3.27.v20190418

1.811 negotiator 0.5.3
1.811.1 Available under license :
(The MIT License)

Copyright (c) 2012-2014 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schlueter
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.812 request 2.88.2

1.812.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source
code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.813 open-csv 2.3

1.813.1 Available under license:
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2007 Kyle Miller.
 * <p/>
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * <p/>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p/>
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-jar/au/com/bytecode/opencsv/bean/HeaderColumnNameMappingStrategy.java
No license file was found, but licenses were detected in source scan.

A very simple CSV parser for Java released under a commercial-friendly license.

Found in path(s):
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-jar/au/com/bytecode/opencsv/package.html
No license file was found, but licenses were detected in source scan.

/**
Copyright 2005 Bytecode Pty Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/**

Found in path(s):
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-jar/au/com/bytecode/opencsv/ResultSetHelperService.java
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-jar/au/com/bytecode/opencsv/ResultSetHelper.java
No license file was found, but licenses were detected in source scan.

/**

Copyright 2005 Bytecode Pty Ltd.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/**

* A very simple CSV writer released under a commercial-friendly license.
*
* @author Glen Smith
*
*
*/

Found in path(s):
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-jar/au/com/bytecode/opencsv/CSVWriter.java
No license file was found, but licenses were detected in source scan.

/**

Copyright 2005 Bytecode Pty Ltd.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Open Source Used In DNAC Platform Antelope 2705
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

/**
**
* A very simple CSV parser released under a commercial-friendly license.
* This just implements splitting a single line into fields.
*
* @author Glen Smith
* @author Rainer Pruy
*/

Found in path(s):
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-
doc/au/com/bytecode/opencsv/CSVParser.java
No license file was found, but licenses were detected in source scan.

/**
Copyright 2007 Kyle Miller.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-
doc/au/com/bytecode/opencsv/bean/MappingStrategy.java
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-
doc/au/com/bytecode/opencsv/bean/ColumnPositionMappingStrategy.java
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-
doc/au/com/bytecode/opencsv/bean/CsvToBean.java
No license file was found, but licenses were detected in source scan.

/**
Copyright 2005 Bytecode Pty Ltd.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

A very simple CSV reader released under a commercial-friendly license.
* @author Glen Smith

Found in path(s):
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-
jar/au/com/bytecode/opencsv/CSVReader.java
No license file was found, but licenses were detected in source scan.

Copyright 2007,2010 Kyle Miller.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136206716_1613708914.54/0/opencsv-2-3-sources-3-
jar/au/com/bytecode/opencsv/bean/HeaderColumnNameTranslateMappingStrategy.java

1.814 babel-helper-regex 6.26.0
1.814.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The "Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.815 safe-buffer 5.1.2 5.1.2
1.815.1 Available under license:
The MIT License (MIT)

Copyright (c) Feross Aboukhadieh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.816 has-flag 3.0.0
1.816.1 Available under license:
MIT License

Copyright (c) Sindresorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.817 arraybuffer.slice 0.0.6 0.0.6

1.817.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.818 has-binary 0.1.7 0.1.7

1.818.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Kevin Roark

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
1.819 minimist 0.0.8 0.0.8
1.819.1 Available under license :
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.820 jpam 1.1
1.820.1 Available under license :
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2003-2007 Greg Luck
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
Found in path(s):
* /opt/cola/permits/1126356109_1611624421.8/0/jpam-1-1-src-zip/JPam-1.1/java/net/sf/jpam/Pam.java
* /opt/cola/permits/1126356109_1611624421.8/0/jpam-1-1-src-zip/JPam-1.1/java/net/sf/jpam/jaas/JpamLoginModule.java
* /opt/cola/permits/1126356109_1611624421.8/0/jpam-1-1-src-zip/JPam-1.1/java/net/sf/jpam/PamReturnValue.java
* /opt/cola/permits/1126356109_1611624421.8/0/jpam-1-1-src-zip/JPam-1.1/java/net/sf/jpam/PamException.java
No license file was found, but licenses were detected in source scan.

License: Apache Software License, v. 1.1

Found in path(s):
* /opt/cola/permits/1126356109_1611624421.8/0/jpam-1-1-src-zip/JPam-1.1/rpm/jpam.spec

1.821 jsesc 0.5.0 0.5.0
1.821.1 Available under license:
Copyright Mathias Bynens <http://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.822 babel-plugin-proposal-dynamic-import
7.13.8
1.822.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.823 moment-timezone 0.5.31

1.823.1 Available under license:

The MIT License (MIT)

Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.824 protobuf-java 3.11.1
1.824.1 Available under license:

<OWNER> = Regents of the University of California
<ORGANIZATION> = University of California, Berkeley
<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.825 is-stream 1.1.0

1.825.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.826 cli-width 2.2.1

1.826.1 Available under license:
Copyright (c) 2015, Ilya Radchenko <ilya@burstcreations.com>

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.827 globals 9.18.0 9.18.0
1.827.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.828 async 1.4.1
1.828.1 Available under license:

The MIT License (MIT)

Copyright (c) Spatie bvba <info@spatie.be>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.829 xml-apis 1.4.01

1.829.1 Available under license:

xml-commons/java/external/LICENSE.dom-software.txt $Id: LICENSE.dom-software.txt 734314 2009-01-14 03:33:27Z mrglavas $

This license came from: http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/java-binding.zip
(COPYRIGHT.html)

W3C SOFTWARE NOTICE AND LICENSE

All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires "Notice of any changes or modifications to the W3C files, including the date changes were made." Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL definitions, the pragma prefix can no longer be 'w3c.org'; in the case of the Java language binding, the package names can no longer be in the 'org.w3c' package.

Note: The original version of the W3C Software Copyright Notice and License could be found at http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)
THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission.
Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=========================================================================  
==  NOTICE file corresponding to section 4(d) of the Apache License,  
==  Version 2.0, in this case for the Apache xml-commons xml-apis  
==  distribution.                                                      
=========================================================================  

Apache XML Commons XML APIs
Copyright 1999-2009 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- $Id: LICENSE.sac.html 477037 2006-11-20 04:30:53Z mrglavas $ -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" 
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>W3C IPR SOFTWARE NOTICE</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css">


Note: The original version of the W3C Software Copyright Notice and License could be found at http://www.w3.org/Consortium/Legal/copyright-software-19980720


This W3C work (including software, documents, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, and modify this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code: "Copyright 2002 World Wide Web Consortium," Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University. All Rights Reserved.

Notice of any changes or modifications to the W3C files, including the date changes were made. (We recommend you provide URIs to the location
This license came from: http://www.w3.org/Consortium/Legal/copyright-documents-20021231

W3C DOCUMENT LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231

Public documents on the W3C site are provided by the copyright holders under the following license. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright [Sdate-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered
under http://saxproject.org/?selected=pd

This page is now out of date -- see the new SAX site at
http://www.saxproject.org/ for more up-to-date
releases and other information. Please change your bookmarks.

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for
XML), and release all of the SAX 2.0 source code, compiled code, and
documentation contained in this distribution into the Public Domain.
SAX comes with NO WARRANTY or guarantee of fitness for any
purpose.

David Megginson, david@megginson.com
2000-05-05

1.830 perfmark-api 0.19.0
1.830.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of
fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source
code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICEx file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Open Source Used In DNAC Platform Antelope 2730
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.831 hbase-shaded-jetty 3.4.1

1.831.1 Available under license:

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

==============================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
==============================================================

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and
* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
  * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
  * org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
  * org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
  https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

  * javax.servlet:javax.servlet-api
  * javax.annotation:javax.annotation-api
  * javax.transaction:javax.transaction-api
  * javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following
artifacts may be included in the distribution or downloaded when ALPN
module is selected.

  * java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications
are hosted at github and both modified and original are under GPL v2 with
classpath exceptions.
  http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the
terms of http://asm.ow2.org/license.html
The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is
1.8.32 javassist 3.27.0-GA

1.8.32.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Javassist, a Java-bytecode translator toolkit.
 * Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.
 * 
 * The contents of this file are subject to the Mozilla Public License Version
 * 1.1 (the "License"); you may not use this file except in compliance with
 * the License. Alternatively, the contents of this file may be used under
 * the terms of the GNU Lesser General Public License Version 2.1 or later,
 * or the Apache License Version 2.0.
 * 
 * Software distributed under the License is distributed on an "AS IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
 * for the specific language governing rights and limitations under the
 * License.
 */

Found in path(s):
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/rmi/Proxy.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/Visitor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/Member.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/reflect/CannotInvokeException.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtNewWrappedConstructor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/LoaderClassPath.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/Mnemonic.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/ArrayInit.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/SignatureAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtArray.java

grant provided that the copyright notice appears in all copies.
Open Source Used In DNAC Platform Antelope
jar/javassist/tools/web/BadHttpRequest.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/BadBytecode.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/ClassPool.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/proxy/Proxy.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtNewWrappedMethod.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/analysis/Type.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/SyntheticAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/rmi/ObjectImporter.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/ExprEditor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ProceedHandler.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/ExceptionsAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/convert/TransformAccessArrayField.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/AnnotationDefaultAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/scopedpool/SoftValueHashMap.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/reflect/Loader.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtField.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/Dump.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/MethodDecl.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/CompileError.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/web/Viewer.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/stackmap/Tracer.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/util/HotSwapAgent.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/reflect/HotSwapAgent.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/bytecode/InnerClassesAttribute.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/web/FieldAccess.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/web/ProxyObject.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/convert/TransformCall.java

Open Source Used In DNAC Platform Antelope  2739
Open Source Used In DNAC Platform Antelope  2740
jar/javassist/tools/web/Webserver.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/LongVector.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/util/proxy/ProxyObjectOutputStream.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/ CastExpr.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/DeprecateAttribute.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/CtClass.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/Translator.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/tools/rmi/Sample.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/CodeAttribute.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/NewArray.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/ ClassClassPath.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/IntConst.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/ast/MethodHandler.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/TransformNew.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/scopedpool/ScopedClassPoolRepository.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/expr/ Cast.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/NoFieldException.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/SyntaxError.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/TokenId.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/compiler/Util/RunTimeSupport.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-jar/javassist/compiler/ast/Pair.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/convert/TransformWriteField.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/CtNewConstructor.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/ConstPool.java
* /opt/ cola/ permits/ 1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1-
jar/javassist/bytecode/analysis/MultiType.java

No license file was found, but licenses were detected in source scan.

/*
 * Javassist, a Java-bytecode translator toolkit.
 * Copyright (C) 2004 Bill Burke. All Rights Reserved.
 */
The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. Alternatively, the contents of this file may be used under the terms of the GNU Lesser General Public License Version 2.1 or later, or the Apache License Version 2.0.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

Found in path(s):
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/MemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/EnumMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/ByteMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/ClassMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/MemberValueVisitor.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/FloatMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/IntegerMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/CharMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/Annotation.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/DoubleMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/AnnotationMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/BooleanMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/LongMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/AnnotationMemberValue.java
* /opt/cola/permits/1135872664_1613621010.52/0/javassist-3-27-0-ga-sources-1.jar/javassist/bytecode/annotation/AnnotationMemberValue.java
1.833 netty-codec-http 4.1.42.Final

1.833.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/DefaultHttpObject.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/HttpContent.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/LastHttpContent.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/ReadOnlyHttpHeaders.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/HttpServerExpectContinueHandler.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/websocketx/WebSocketScheme.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/websocketx/ContinuationWebSocketFrame.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/websocketx/WebSocketClientHandshaker07.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/RtspRequestEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/websocketx/RtspMethods.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/websocketx/HttpVersion.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/websocketx/DiskFileUpload.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/websocketx/HttpContentCompressor.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/multipart/FileUpload.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/multipart/DiskFileUpload.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codehttp/RtspRequestEncoder.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019 The Netty Project
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionFilter.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/http/HttpServerKeepAliveHandler.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/http/multipart/FileUploadUtil.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/http/websocketx/WebSocketChunkedInput.java
No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/META-INF/native-image/io.netty/codec-http/native-image.properties
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

// (BSD License: http://www.opensource.org/licenses/bsd-license)
// All rights reserved.
// Redistribution and use in source and binary forms, with or
// * Redistributions of source code must retain the above
// * copyright notice, this list of conditions and the
// * following disclaimer.
// * Redistributions in binary form must reproduce the above
// * following disclaimer in the documentation and/or other
// * Neither the name of the Webbit nor the names of

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/http/websocketx/WebSocket07FrameEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/http/websocketx/WebSocket08FrameEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/http/websocketx/WebSocket13FrameEncoder.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/HttpClientUpgradeHandler.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/HttpServerUpgradeHandler.java

No license file was found, but licenses were detected in source scan.
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

// (BSD License: http://www.opensource.org/licenses/bsd-license)
// All rights reserved.
// Redistribution and use in source and binary forms, with or
// Redistributions of source code must retain the above
// copyright notice, this list of conditions and the
// following disclaimer.
// Redistributions in binary form must reproduce the above
// following disclaimer in the documentation and/or other
// Neither the name of the Webbit nor the names of

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/CombinedHttpHeaders.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/cookie/ClientCookieDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/cookie/CookieCodec.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/cookie/CookieHeaderNames.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/cookie/DefaultCookie.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/cookie/EmptyHttpHeaders.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/cookie/SpdyProtocolException.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/http/ClientCookieEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/http/CombinedHttpHeaders.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/http/cookie/ClientCookieDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/http/cookie/CookieCodec.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/http/cookie/CookieHeaderNames.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/http/rtsp/RtspDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/http/cookie/ServerCookieEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/http/cookie/ServerCookieEncoder.java
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version
* 2.0 (the "License"); you may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871240_1595214773.59/0/netty-codec-http-4-1-42-final-sources-1-
jar/io/netty/handler/codec/http/cors/CorsConfigBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/*
* Adaptation of http://bjoern.hoehrmann.de/utf-8/decoder/dfa/
*
* Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>
*
* Permission is hereby granted, free of charge, to any person obtaining a copy of this software
* and associated documentation files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge, publish, distribute,
* sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in all copies or
* substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
* BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE
FOR ANY CLAIM,
* DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
*/

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE
2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit

1.834 babel-plugin-syntax-async-functions

6.13.0 6.13.0

1.834.1 Available under license :
The MIT License (MIT)
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.835 request-local 1.0.5 1.0.5
1.835.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   You must give any other recipients of the Work or Derivative Works a copy of this License; and
   You must cause any modified files to carry prominent notices stating that You changed the files; and
   You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
   If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
1.836 expat 2.1.0
1.836.1 Available under license :
Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper
Copyright (c) 2001-2019 Expat maintainers

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.837 fs.realpath 1.0.0
1.837.1 Available under license :
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
This library bundles a version of the `fs.realpath` and `fs.realpathSync` methods from Node.js v0.10 under the terms of the Node.js MIT license.

Node's license follows, also included at the header of `old.js` which contains the licensed code:

Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.838 twill-api 0.6.0-incubating

1.838.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
Found in path(s):
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/internal/DefaultTwillSpecification.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/Racks.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/ClassAcceptor.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/ServiceController.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/SecureStore.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/internal/DefaultEventHandlerSpecification.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/ServiceAnnouncer.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/internal/RunIds.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/internal/DefaultLocalFile.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/internal/DefaultTwillRunnableSpecification.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/TwillRunResources.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/Hosts.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/EventHandler.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/TwillSpecification.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/TwillPreparer.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/TwillRunner.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/LocalFile.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/internal/package-info.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/AbstractTwillRunnable.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/api/TwillController.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-jar/org/apache/twill/internal/DefaultResourceSpecification.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/logging/LogHandler.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/TwillContext.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/EventHandlerSpecification.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/SecureStoreUpdater.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/Command.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/RuntimeSpecification.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/package-info.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/ResourceReport.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/logging/PrinterLogHandler.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/logging/LogEntry.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/EventHandlerContext.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/TwillRunnerService.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/logging/package-info.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/TwillRunnableSpecification.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/ElectionHandler.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/api/TwillApplication.java
* /opt/cola/permits/1137746702_1614121526.26/0/twill-api-0-6-0-incubating-sources-3-
jar/org/apache/twill/internal/DefaultTwillRunResources.java

1.839 i 0.3.6
1.839.1 Available under license:
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.840 is-equal-shallow 0.1.3
1.840.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.841 xtend 4.0.1 4.0.1

1.841.1 Available under license:

Copyright (c) 2012-2014 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.842 node-readfiles 0.2.0 0.2.0

1.842.1 Available under license:

(The MIT License)

Copyright (c) 2016 Alejandro Gonzalez <guatedude2@hotmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
1.843 yargs 13.3.2

1.843.1 Available under license:

Copyright 2010 James Halliday (mail@substack.net)
Modified work Copyright 2014 Contributors (ben@npmjs.com)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.844 @babel/helper-skip-transparent-expression-wrappers 7.12.1

1.844.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.845 @babel/plugin-syntax-async-generators 7.8.4
1.845.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.846 debug 3.1.0 3.1.0
1.846.1 Available under license :
(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.847 jol-core 0.9

1.847.1 Available under license:
The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change
the software or use pieces of it in new free programs; and that you know you
can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny
you these rights or to ask you to surrender the rights. These restrictions
translate to certain responsibilities for you if you distribute copies of the
software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for
a fee, you must give the recipients all the rights that you have. You must
make sure that they, too, receive or can get the source code. And you must
show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2)
offer you this license which gives you legal permission to copy, distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain that
everyone understands that there is no warranty for this free software. If the
software is modified by someone else and passed on, we want its recipients to
know that what they have is not the original, so that any problems introduced
by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We
wish to avoid the danger that redistributors of a free program will
individually obtain patent licenses, in effect making the program proprietary.
To prevent this, we have made it clear that any patent must be licensed for
everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification
follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice
placed by the copyright holder saying it may be distributed under the terms of
this General Public License. The "Program", below, refers to any such program
or work, and a "work based on the Program" means either the Program or any
derivative work under copyright law: that is to say, a work containing the
Program or a portion of it, either verbatim or with modifications and/or
translated into another language. (Hereinafter, translation is included
without limitation in the term "modification".) Each licensee is addressed as
"you".

Activities other than copying, distribution and modification are not covered by
this License; they are outside its scope. The act of running the Program is
not restricted, and the output from the Program is covered only if its contents
constitute a work based on the Program (independent of having been made by
running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that
distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs
If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL

Certain source files distributed by Oracle America and/or its affiliates are subject to the following clarification and special exception to the GPL, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.848 jsonfile 2.4.0

1.848.1 Available under license:
(The MIT License)

Copyright (c) 2012-2015, JP Richardson <jprichardson@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the
Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS
OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

1.849 eyes 0.1.8
1.849.1 Available under license :
Copyright (c) 2009 cloudhead

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.850 babel-plugin-transform-regenerator
6.26.0 6.26.0
1.850.1 Available under license:
   The MIT License (MIT)

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.851 hadoop-aws 3.2.2
1.851.1 Available under license:
   This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

   The binary distribution of this product bundles binaries of
org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the
following notices:
   * Copyright 2011 Dain Sundstrom <dain@iq80.com>
   * Copyright 2011 FuseSource Corp. http://fusesource.com

   The binary distribution of this product bundles binaries of
AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java),
which has the following notices:

   This software includes third party software subject to the following copyrights:
   - XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
   - PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-
2010 AOL Inc.

   The binary distribution of this product bundles binaries of
Gson 2.2.4,
which has the following notices:

   The Netty Project
   =================
Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of Webbit, an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit
This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
* https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)
===============================================================================
The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors. Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:

* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices:

* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
  Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
  TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of "Java Concurrency in Practice" book annotations 1.0, which has the following notices:

* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5) Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of Jetty :: Http Utility 9.3.19, Jetty :: IO Utility 9.3.19, Jetty :: Security 9.3.19, Jetty :: Server Core 9.3.19, Jetty :: Servlet Handling 9.3.19, Jetty :: Utilities 9.3.19, Jetty :: Utilities :: Ajax, Jetty :: Webapp Application Support 9.3.19, Jetty :: XML utilities 9.3.19,
which has the following notices:

Jetty Web Container
Copyright 1995-2016 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK
If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html
org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by
Unix systems for simple password protection. Copyright 1996 Aki Yoshida,
modified April 2001 by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt
for non-commercial or commercial purposes and without fee is
granted provided that the copyright notice appears in all copies./

The binary distribution of this product bundles binaries of
Snappy for Java 1.0.4.1,
which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
(Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by
"GCC RUNtime Library Exception"

== Contributors ==
* Tatu Saloranta
  * Providing benchmark suite
* Alec Wysoker
  * Performance and memory usage improvement

The binary distribution of this product bundles binaries of
Xerces2 Java Parser 2.9.1,
which has the following notices:
* =================================================================================================
Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

The binary distribution of this product bundles binaries of
Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
  * Logback: the reliable, generic, fast and flexible logging framework.
    Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
  * Apache HBase
    Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0
--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.
Licensed under the Apache License v2.0
This product uses the Glyphicons Halflings icon set.
http://glyphicons.com/

Copyright Jan Kovak
Licensed under the Apache License v2.0 as a part of the Bootstrap project.

This product includes portions of the Guava project v14, specifically 'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors
Licensed under the Apache License, Version 2.0
The binary distribution of this product bundles binaries of Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This also includes:
The phoenix-spark module has been adapted from the phoenix-spark library distributed under the terms of the Apache 2 license. Original source copyright: Copyright 2014 Simply Measured, Inc.
The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5 (https://pypi.python.org/pypi/python-daemon/). Original source copyright:

# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
  which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1,
which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9
which has the following notices:
* =--------------------------------------------------------------------------------------------=
  = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
  =--------------------------------------------------------------------------------------------=
  This product includes software developed by
  Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 *
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the distribution.
* - Neither the name of the University Catholique de Louvain - UCL
* nor the names of its contributors may be used to endorse or
* promote products derived from this software without specific prior
* written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
* ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

For portions of the native implementation of slicing-by-8 CRC calculation
in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor
  the names of its contributors may be used to endorse or promote
  products derived from this software without specific prior written
  permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*++
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 * Abstract: The main routine
 * ...
 */

For src/main/native/src/org/apache/hadoop/io/compress/lz4/{lz4.h,lz4.c,lz4hc.h,lz4hc.c},

/*
LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

   * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :
- LZ4 source repository : http://code.google.com/p/lz4/
- LZ4 public forum : https://groups.google.com/forum/#!forum/lz4c

/*

For hadoop-common-project/hadoop-common/src/main/native/gtest

****************************************************************************
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
    copyright notice, this list of conditions and the following disclaimer
    in the documentation and/or other materials provided with the
    distribution.
  * Neither the name of Google Inc. nor the names of its
    contributors may be used to endorse or promote products derived from
    this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the
following license:
re2j 1.1

****************************************************************************

(GO license)
This is a work derived from Russ Cox's RE2 in Go, whose license
http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

---------------------------------------------------------------------

Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
The binary distribution of this product bundles binaries of leveldbjni
(https://github.com/fusesource/leveldbjni), which is available under the
following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of FuseSource Corp. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
nodemanager/src/main/native/container-executor/impl/compat/\{fstatat|openat|unlinkat\}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from
the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===============

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

--------------------------------------------------------------------------------

Copyright (C) 2008-2016, SpryMedia Ltd.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

--------------------------------------------------------------------------------
Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------
Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
em/helpers v0.5.13
em-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)
--------------------------------------------------------------------------------
Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this
file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the
License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
express or implied. See the License for the specific language governing permissions and
limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js
--------------------------------------------------------------------------------
The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software
and associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT
LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE
OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js
The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery

Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
oj! Algorithms - version 43.0

--------------------------------------------------------------------------------


This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js
--------------------------------------------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js
--------------------------------------------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

D3 is available under a 3-clause BSD license. For details, see:
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
HSQLDB Database 2.3.4

(HSQL License)
"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.

The binary distribution of this product bundles these dependencies under the following license:

Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this
1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions
(b) under Patent Claims infringed by the making, using or selling of
Original Software, to make, have made, use, practice, sell, and offer for sale,
and/or otherwise dispose of the Original Software (or portions
thereof);

(c) The licenses granted in Sections2.1(a) and (b) are
effective on the date Initial Developer first distributes or otherwise makes the
Original Software available to a third party under the terms of this
License;

(d) Notwithstanding Section2.1(b) above, no patent license is
granted: (1) for code that You delete from the Original Software, or (2) for
infringements caused by: (i) the modification of the Original Software, or
(ii) the combination of the Original Software with other software or
devices.

2.2. Contributor Grant. Conditioned upon Your compliance with
Section 3.1 below and subject to third party intellectual property claims, each
Contributor hereby grants You a world-wide, royalty-free, non-exclusive
license:

(a) under intellectual property rights (other than patent or
trademark) Licensable by Contributor to use, reproduce, modify, display,
perform, sublicense and distribute the Modifications created by such Contributor
(or portions thereof), either on an unmodified basis, with other Modifications,
as Covered Software and/or as part of a Larger Work; and

(b) under Patent
Claims infringed by the making, using, or selling of Modifications made by that
Contributor either alone and/or in combination with its Contributor Version (or
portions of such combination), to make, use, sell, offer for sale, have made,
and/or otherwise dispose of: (1) Modifications made by that Contributor (or
portions thereof); and (2) the combination of Modifications made by that
Contributor with its Contributor Version (or portions of such
combination).

(c) The licenses granted in Sections2.2(a) and 2.2(b) are
effective on the date Contributor first distributes or otherwise makes the
Modifications available to a third party.

(d) Notwithstanding Section2.2(b)
above, no patent license is granted: (1) for any code that Contributor has
deleted from the Contributor Version; (2) for infringements caused by:
(i) third party modifications of Contributor Version, or (ii) the combination
of Modifications made by that Contributor with other software (except as part of
the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this
License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY
COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER
OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR
CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights
granted hereunder will terminate automatically if You fail to comply with terms
herein and fail to cure such breach within 30 days of becoming aware of the
breach. Provisions which, by their nature, must remain in effect beyond the
termination of this License shall survive.

6.2. If You assert a patent
infringement claim (excluding declaratory judgment actions) against Initial
Developer or a Contributor (the Initial Developer or Contributor against whom
You assert such claim is referred to as Participant) alleging that the
Participant Software (meaning the Contributor Version where the Participant is a
Contributor or the Original Software where the Participant is the Initial
Developer) directly or indirectly infringes any patent, then any and all rights
granted directly or indirectly to You by such Participant, the Initial Developer
(if the Initial Developer is not the Participant) and all Contributors under
Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from
Participant terminate prospectively and automatically at the expiration of such
60 day notice period, unless if within such 60 day period You withdraw Your
claim with respect to the Participant Software against such Participant either
unilaterally or pursuant to a written agreement with Participant.

6.3. In
the event of termination under Sections 6.1 or 6.2 above, all end user licenses
that have been validly granted by You or any distributor hereunder prior to
termination (excluding licenses granted to You by any distributor) shall survive
termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER
NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE,
SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF
COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY
PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY
CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF
GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER
COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE
POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO
LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO
THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT
ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO
THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
gizzly-framework 2.2.21
gizzly-http 2.2.21
gizzly-http-server 2.2.21
gizzly-http-servlet 2.2.21
gizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or
deposits contributions to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software,
prior Modifications used by a Contributor (if any), and the Modifications made
by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b)
Modifications, or (c) the combination of files containing Original Software with
files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source
Code.
1.5. Initial Developer means the individual or entity that first makes
Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions
thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent
possible, whether at the time of the initial grant or subsequently acquired, any
and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the
following:
A. Any file that results from an addition to, deletion from or modification of
the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous
Modification; or
C. Any new file that is contributed or otherwise made available under the terms
of this License.
1.10. Original Software means the Source Code and Executable form of
computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter
acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof),
either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for,
warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an
Initial Developer and You want to create a new license for Your Original Software. You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.
6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The
application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URLs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of
intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable
law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the
  names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the
following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

(2-clause BSD)
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice” book annotations 1.0

(CCAL v2.5)

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work” means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work” means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor” means the individual or entity that offers the Work under the terms of this License.

"Original Author” means the individual or entity who created the Work.

"Work” means the copyrightable work of authorship offered under the terms of
this License.
"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:
You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICensor OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH
6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the
following license:
jamron-runtime 2.4.1

(MPL 2.0)
Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means
that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.
1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.
2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in
Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.
3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.
6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License
10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:

JDOM 1.1

-----------------------------------------------------------------------------
/*--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions, and the disclaimer that follows
these conditions in the documentation and/or other materials
provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without prior written permission
from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:

"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4

--------------------------------------------------------------------------------
This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical
work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this
License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any
exercise by You of the rights granted under this License;
ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied,
and/or if the Original Author and/or Licensor designate another party
or parties (e.g., a sponsor institute, publishing entity, journal) for
attribution ("Attribution Parties") in Licensor's copyright notice,
terms of service or by other reasonable means, the name of such party
or parties; (ii) the title of the Work if supplied; (iii) to the
extent reasonably practicable, the URL, if any, that Licensor
specifies to be associated with the Work, unless such URI does not
refer to the copyright notice or licensing information for the Work;
and (iv) , consistent with Section 3(b), in the case of an Adaptation,
a credit identifying the use of the Work in the Adaptation (e.g.,
"French translation of the Work by Original Author," or "Screenplay
based on original Work by Original Author"). The credit required by
this Section 4 (b) may be implemented in any reasonable manner;
provided, however, that in the case of a Adaptation or Collection, at
a minimum such credit will appear, if a credit for all contributing
authors of the Adaptation or Collection appears, then as part of these
credits and in a manner at least as prominent as the credits for the
other contributing authors. For the avoidance of doubt, You may only
use the credit required by this Section for the purpose of attribution
in the manner set out above and, by exercising Your rights under this
License, You may not implicitly or explicitly assert or imply any
connection with, sponsorship or endorsement by the Original Author,
Licensor and/or Attribution Parties, as appropriate, of You or Your
use of the Work, without the separate, express prior written
permission of the Original Author, Licensor and/or Attribution
Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be
otherwise permitted by applicable law, if You Reproduce, Distribute or
Publicly Perform the Work either by itself or as part of any
Adaptations or Collections, You must not distort, mutilate, modify or
take other derogatory action in relation to the Work which would be
prejudicial to the Original Author's honor or reputation. Licensor
agrees that in those jurisdictions (e.g. Japan), in which any exercise
of the right granted in Section 3(b) of this License (the right to
make Adaptations) would be deemed to be a distortion, mutilation,
modification or other derogatory action prejudicial to the Original
Author's honor and reputation, the Licensor will waive or not assert,
as appropriate, this Section, to the fullest extent permitted by the
applicable national law, to enable You to reasonably exercise Your
right under Section 3(b) of this License (right to make Adaptations)
but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR
OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY
KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE,
INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF
LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS,
WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION
OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE
LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR
ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES
ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS
BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate
automatically upon any breach by You of the terms of this License.
Individuals or entities who have received Adaptations or Collections
from You under this License, however, will not have their licenses
terminated provided such individuals or entities remain in full
compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will
survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is
perpetual (for the duration of the applicable copyright in the Work).
Notwithstanding the above, Licensor reserves the right to release the
Work under different license terms or to stop distributing the Work at
any time; provided, however that any such election will not serve to
withdraw this License (or any other license that has been, or is
required to be, granted under the terms of this License), and this
License will continue in full force and effect unless terminated as
stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection,
the Licensor offers to the recipient a license to the Work on the same
terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor
offers to the recipient a license to the original Work on the same
terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of
the remainder of the terms of this License, and without further action
by the parties to this agreement, such provision shall be reformed to
the minimum extent necessary to make such provision valid and
enforceable.
d. No term or provision of this License shall be deemed waived and no
breach consented to unless such waiver or consent shall be in writing
and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with
respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.

For: hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/datanode/checker/AbstractFuture.java and
Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.852 chalk 1.1.3 1.1.3
1.852.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.853 nan 2.10.0 2.10.0
1.853.1 Available under license:
The MIT License (MIT)

Copyright (c) 2018 NAN contributors

*NAN contributors listed at <https://github.com/nodejs/nan#contributors>*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.854 @babel/plugin-transform-modules-commonjs 7.13.8

1.854.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.855 flat-cache 1.3.4
1.855.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Roy Riojas

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.856 guava 30.0-jre
1.856.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 * *
 * Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/AtomicLongMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/GwtTransient.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2013 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/hash/HashingInputStream.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/Verify.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/Runnables.java
*/
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/io/CharSequenceReader.java
No license file was found, but licenses were detected in source scan.

/
* Copyright (C) 2005 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

/
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/ClosingFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/AbstractBaseGraph.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/Traverser.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/primitives/Platform.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/Internal.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/net/InternetDomainName.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collect/RegularImmutableTable.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/hash/Striped64.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/cache/Striped64.java

No license file was found, but licenses were detected in source scan.

/
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/util/concurrent/AtomicDoubleArray.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/hash/Striped64.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/cache/Striped64.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/**
 * Holder for web specializations of methods of { @code Doubles }. Intended to be empty for regular
 * version.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2010 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

No license file was found, but licenses were detected in source scan.
jar/com/google/common/collection/SortedMapDifference.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingImmutableCollection.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/MinMaxPriorityQueue.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractSequentialIterator.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingListMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/RowSortedTable.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/CompactHashing.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/graph/IncidentEdgeSet.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of { @code Floats}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/primitives/FloatsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of { @code Objects} only in web. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/Hashing.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This following method is a modified version of one found in
 * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
 * which contained the following notice:
 *
 * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
 * the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
 *
 * Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/MoreExecutors.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the
 * License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * either
 * express or implied. See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractRangeSet.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Count.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/RangeSet.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingSortedMultiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/SortedIterables.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/SortedIterable.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ImmutableSortedMultiset.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/graph/package-info.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Collections.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Collectors.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/CollectSpliterators.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/ImmutableLongArray.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/ForwardingCondition.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/ForwardingLock.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Outer class that exists solely to let us write [@code Partially.GwtIncompatible] instead of plain
* [@code GwtIncompatible]. This is more accurate for [@link Futures#catching], which is available
* under GWT but with a slightly different signature.
* 
* <p>We can't use [@code PartiallyGwtIncompatible] because then the GWT compiler wouldn't recognize
* it as a [@code GwtIncompatible] annotation. And for [@code Futures.catching], we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT
* version.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/RegularImmutableMultiset.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ImmutableTable.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ImmutableSetMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/SingletonImmutableTable.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ImmutableEnumSet.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/AbstractIndexedListIterator.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ImmutableClassToInstanceMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ImmutableSortedSetFauxvrideShim.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/SingletonImmutableList.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ComputationException.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ComparisonChain.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ImmutableAsList.java
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Returns an array containing all of the elements in the specified collection. This method
 * returns the elements in the order they are returned by the collection's iterator. The returned
 * array is "safe" in that no references to it are maintained by the collection. The caller is
 * thus free to modify the returned array.
 *
 * <p>This method assumes that the collection size doesn't change while the method is running.
 *
 * <p>TODo(kevinb): support concurrently modified collections?
 *
 * @param c the collection for which to return an array of elements
 */
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/hash/ImmutableSupplier.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/ExecutionSequencer.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/JdkBackedImmutableSet.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/JdkBackedImmutableMultiset.java
No license file was found, but licenses were detected in source scan.
/*
 * Copyright (C) 2020 The Guava Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
 * Holder for web specializations of methods of {Ints. Intended to be empty for regular
 * version.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/primitives/IntsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/RangeMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/TreeTraverser.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/RegularImmutableAsList.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/AbstractNavigableMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/CompactLinkedHashSet.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/AbstractMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/FilteredMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/ForwardingBlockingDeque.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the 
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, 
either 
* express or implied. See the License for the specific language governing permissions and 
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ImmutableSortedAsList.java
No license file was found, but licenses were detected in source scan.

/--* 
* Copyright (C) 2007 The Guava Authors 
* 
* Licensed under the Apache License, Version 2.0 (the "License"); 
* you may not use this file except in compliance with the License. 
* You may obtain a copy of the License at 
* 
* http://www.apache.org/licenses/LICENSE-2.0 
* 
* Unless required by applicable law or agreed to in writing, software 
* distributed under the License is distributed on an "AS IS" BASIS, 
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
* See the License for the specific language governing permissions and 
* limitations under the License. 
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/LinkedHashMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Multimaps.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Multisets.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ReverseNaturalOrdering.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ClassToInstanceMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractSetMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractSortedSetMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/MutableClassToInstanceMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/LinkedListMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingConcurrentMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/EnumBiMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/NaturalOrdering.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractMapEntry.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ImmutableList.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingObject.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Iterables.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Lists.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/SortedSetMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ListMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ConcurrentHashMultiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingMultiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/NullsLastOrdering.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractMultiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractBiMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Multiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ReverseOrdering.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/BitMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingSortedSet.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ByFunctionOrdering.java
Open Source Used In DNAC Platform Antelope  2879

* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/SetMultimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/TreeMultiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Multimap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/EnumHashBiMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractIterator.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/Ordering.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingSortedMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/MapDifference.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/CompoundOrdering.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/AbstractMapBasedMultiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/ForwardingMapEntry.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2010 The Guava Authors
 * *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/annotations/package-info.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/Equivalence.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/annotations/Beta.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/Atomics.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/net/package-info.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/Strings.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/ListeningExecutorService.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/package-info.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2014 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collection/TopKSelector.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collection/ImmutableEntry.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collection/StandardRowSortedTable.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collection/Streams.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collection/JdkBackedImmutableBiMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collection/IndexedImmutableSet.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/JdkBackedImmutableMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collection/BaseImmutableMultimap.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2006 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ImmediateFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/GwtFluentFutureCatchingSpecialization.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/FuturesGetChecked.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/AggregateFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/annotations/VisibleForTesting.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/MoreObjects.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/escape/CharEscaperBuilder.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2014 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/MoreObjects.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/math/Quantiles.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/ListenerCallQueue.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/**
 * Holder for web specializations of methods of [@code Shorts]. Intended to be empty for regular
 * version.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/primitives/ShortsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/SortedMultisets.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/SortedMultiset.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
 * and released to the public domain, as explained at
 * http://creativecommons.org/licenses/publicdomain
 *
 * As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
 * java.util.HashMap class.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
  jar/com/google/common/util/concurrent/Striped.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
  jar/com/google/common/primitives/Doubles.java
Open Source Used In DNAC Platform Antelope 2890

* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/Chars.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/Converter.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/primitives/Bytes.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 */
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}</b>
 * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
 * version.
 *
 * @throws UnsupportedOperationException always
 * @deprecated <b>Pass a key of type '{@code Comparable}' to use '{@link
 *     ImmutableSortedMap#of(Comparable, Object)}'. </b>
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (C) 2013 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collect/MultimapBuilder.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources.jar/com/google/common/collect/ImmutableMapEntry.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/CommonPattern.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/CommonMatcher.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/PatternCompiler.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/JdkPattern.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/GraphConstants.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/EndpointPair.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/MoreCollectors.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/Comparators.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/CollectCollectors.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/LinkedHashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/StandardMutableGraph.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/ImmutableValueGraph.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/ValueGraphBuilder.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/EndpointPairIterator.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/DirectedGraphConnections.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/ForwardingGraph.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/ForwardingValueGraph.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/MapIteratorCache.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/AbstractValueGraph.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/ElementOrder.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/DirectedNetworkConnections.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/GraphBuilder.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/GraphConnections.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/MultiEdgesConnecting.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/NetworkBuilder.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/UndirectedGraphConnections.java
*/opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/graph/StandardMutableValueGraph.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/math/BigDecimalMath.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/hash/Java8Compatibility.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/hash/Java8Compatibility.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/math/ToDoubleRounder.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/io/MultiInputStream.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/collect/HashBiMap.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/io/Resources.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/io/MultiInputStream.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/eventbus/AllowConcurrentEvents.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/collect/EnumMultiset.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/io/package-
info.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/FinalizableReference.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/Charsets.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/Objects.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/io/Closeables.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/Function.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/Predicate.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/eventbus/Subscribe.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/Supplier.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/io/Files.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/util/concurrent/ListenerableFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/Functions.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/eventbus/EventBus.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/io/LittleEndianDataOutputStream.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
jar/com/google/common/base/Defaults.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
*/
the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/IgnoreJRERequirement.java
No license file was found, but licenses were detected in source scan.

*/
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/ServiceManagerBridge.java
No license file was found, but licenses were detected in source scan.

*/
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/cola/permits/1102081303_1603450661.12/0/guava-30-0-jre-sources-
1.857 sindresorhus-is 0.14.0

1.857.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.858 nimbus-jose-jwt 7.9

1.858.1 Available under license :
Nimbus JOSE + JWT

Copyright 2012 - 2018, Connect2id Ltd.
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.859 jackson-annotations 2.11.1

1.859.1 Available under license:

This copy of Jackson JSON processor annotations is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.860 object-assign 4.1.1

1.860.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.861 has-own-deep 1.1.0

1.861.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.862 commons-pool 1.6

1.862.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Pool
Copyright 2001-2012 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.863 cassandra-driver-core 3.2.0

1.864 jmespath-java 1.11.837
1.864.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources.jar/com/amazonaws/jmespath/JmesPathSubExpression.java
* /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources.jar/com/amazonaws/jmespath/JmesPathLiteral.java
* /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources.jar/com/amazonaws/jmespath/OpEquals.java
* /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources.jar/com/amazonaws/jmespath/OpGreaterThanOrEqualTo.java
* /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources.jar/com/amazonaws/jmespath/OpNotEquals.java
*/
jar/com/amazonaws/jmespath/Comparator.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/ObjectMapperSingleton.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/OpLessThan.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/NumericComparator.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathField.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/InvalidTypeException.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathContainsFunction.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathFlatten.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathFunction.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/CamelCaseUtils.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathAndExpression.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathValueProjection.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathExpression.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/OpLessThanOrEqualTo.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathMultiSelectList.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathProjection.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/OpGreaterThan.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathVisitor.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathLengthFunction.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathNotExpression.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathEvaluationVisitor.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathIdentity.java
  * /opt/cola/permits/1137760046_1614121472.51/0/jmespath-java-1-11-837-sources-
jar/com/amazonaws/jmespath/JmesPathFilter.java

No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>
1.865 silent-spawn 0.4.0

1.865.1 Available under license:
MIT License

Copyright (c) 2017 Jan Blaha

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.866 stream-shift 1.0.1 1.0.1

1.866.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.867 extend 3.0.2 3.0.2

1.867.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Stefan Thomas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.868 component-bind 1.0.0

1.868.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.869 performance-now 2.1.0

1.869.1 Available under license:
Copyright (c) 2013 Braveg1rl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.870 glob 7.1.6

1.870.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

## Glob Logo

Glob's logo created by Tanya Brassie <http://tanyabrassie.com/>, licensed under a Creative Commons Attribution-ShareAlike 4.0 International License https://creativecommons.org/licenses/by-sa/4.0/

### 1.871 path-parse 1.0.6
#### 1.871.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Javier Blanco

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.872 drill-logical 1.19.0
1.872.1 Available under license:

Drill : Logical Plan
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.873 netty 4.0.48.Final

1.873.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.
When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in
Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)
Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the
10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, 5th Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of
the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or
Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2011 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
* license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
* http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
* license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
* http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:
This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/protobuf

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/
This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/snappy

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/caliper

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

(BSD License: http://www.opensource.org/licenses/bsd-license)
Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Webbit nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*
 * Copyright (c) 2004-2007 QOS.ch
 * All rights reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
 * LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
 * OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
 * WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 */

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions.
for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
1.874 babel-core 6.26.3

1.874.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.875 jsreport-pdfjs 0.5.1

1.875.1 Available under license:

The MIT License (MIT)
Copyright (c) 2013-2017 Markus Ast

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.876 array-flatten 1.1.1
1.876.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.877 hikaricp 3.4.2
1.877.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.878 netty-codec 4.1.42.Final

1.878.1 Available under license:

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/DateFormatter.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4.1.42-final-sources-1-jar/io/netty/handler/codec/CharSequenceValueConverter.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/compression/Lz4XXHash32.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/bytes/ByteArrayEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/MessageToMessageEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/MessageToByteEncoder.java
Open Source Used In DNAC Platform Antelope 2964

* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/ByteToMessageDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/marshalling/MarshallingEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/marshalling/MarshallingDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/string/StringDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/string/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/marshalling/LimitingByteInput.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/marshalling/ClassLoader механизмом ClassResolvers.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/marshalling/LimitingByteInput.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/ReplayingDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/ReplayingDecoderByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/serialization/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/bytes/ByteArrayDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/DecoderException.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/MessageAggregator.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1.jar/io/netty/handler/codec/DelimiterBasedFrameDecoder.java

No license file was found, but licenses were detected in source scan.

/ * Copyright 2014 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
* */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/EmptyHeaders.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/DefaultHeaders.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/Headers.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 * */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/compression/Crc32c.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/compression/JdkZlibDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/xml/XmlFrameDecoder.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
*

---

Open Source Used In DNAC Platform Antelope 2957
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* A decoder that splits the received @link ByteBuf}s dynamically by the
* value of the length field in the message. It is particularly useful when you
* decode a binary message which has an integer header field that represents the
* length of the message body or the whole message.
* <p>
* @link LengthFieldBasedFrameDecoder} has many configuration parameters so
* that it can decode any message with a length field, which is often seen in
* proprietary client-server protocols. Here are some example that will give
* you the basic idea on which option does what.
* 
* <h3>2 bytes length field at offset 0, do not strip header</h3>
* 
* The value of the length field in this example is \texttt{12 (0x0C)} which
* represents the length of "HELLO, WORLD". By default, the decoder assumes
* that the length field represents the number of the bytes that follows the
* length field. Therefore, it can be decoded with the simplistic parameter
* combination.
* 
* BEFORE DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content |----->| Length | Actual Content |
* +--------+----------------+      +--------+----------------+
* \texttt{0x000C | "HELLO, WORLD" | 0x000C | "HELLO, WORLD" |
* +--------+----------------+      +--------+----------------+}
* 
* AFTER DECODE (14 bytes)
* 
* <h3>2 bytes length field at offset 0, strip header</h3>
* 
* Because we can get the length of the content by calling
* @link ByteBuf#readableBytes\}}, you might want to strip the length
* field by specifying `<tt>initialBytesToStrip</tt>`. In this example, we
* specified `<tt>2</tt>`, that is same with the length of the length field, to
* strip the first two bytes.
* `<pre>
* lengthFieldOffset  = 0
* lengthFieldLength  = 2
* lengthAdjustment   = 0
* `<b>initialBytesToStrip</b>` = `<b>2</b>` (= the length of the Length field)
* 
* BEFORE DECODE (14 bytes) AFTER DECODE (12 bytes)
* +--------+----------------+      +----------------+
* | Length | Actual Content |----->| Actual Content |
* | 0x000C | "HELLO, WORLD" |      | "HELLO, WORLD" |
* +--------+----------------+      +----------------+
* </pre>
* *
* <h3>2 bytes length field at offset 0, do not strip header, the length field
* represents the length of the whole message</h3>
* *
* * In most cases, the length field represents the length of the message body
* only, as shown in the previous examples. However, in some protocols, the
* length field represents the length of the whole message, including the
* message header. In such a case, we specify a non-zero
* `<tt>lengthAdjustment</tt>`. Because the length value in this example message
* is always greater than the body length by `<tt>2</tt>`, we specify `<tt>-2</tt>`
* as `<tt>lengthAdjustment</tt>` for compensation.
* `<pre>
* lengthFieldOffset  = 0
* lengthFieldLength  = 2
* `<b>lengthAdjustment</b>` = `<b>-2</b>` (= the length of the Length field)
* initialBytesToStrip = 0
* 
* BEFORE DECODE (14 bytes) AFTER DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content |----->| Length | Actual Content |
* | 0x000E | "HELLO, WORLD" |      | 0x000E | "HELLO, WORLD" |
* +--------+----------------+      +--------+----------------+
* </pre>
* *
* <h3>3 bytes length field at the end of 5 bytes header, do not strip header</h3>
* *
* The following message is a simple variation of the first example. An extra
* header value is prepended to the message. `<tt>lengthAdjustment</tt>` is zero
* again because the decoder always takes the length of the prepended data into
* account during frame length calculation.
* `<pre>
* `<b>lengthFieldOffset</b>` = `<b>2</b>` (= the length of Header 1)
* `<b>lengthFieldLength</b>` = `<b>3</b>`
* lengthAdjustment = 0
* initialBytesToStrip = 0

* BEFORE DECODE (17 bytes) AFTER DECODE (17 bytes)
* +-------------------------------------+-------------------------------------+
* | Header 1 | Length | Actual Content |----->| Header 1 | Length | Actual Content |
* | 0xCAFE  | 0x00000C | "HELLO, WORLD" |      | 0xCAFE  | 0x00000C | "HELLO, WORLD" |
* +-------------------------------------+-------------------------------------+
* </pre>

* <h3>3 bytes length field at the beginning of 5 bytes header, do not strip header</h3>

* This is an advanced example that shows the case where there is an extra
* header between the length field and the message body. You have to specify a
* positive <tt>lengthAdjustment</tt> so that the decoder counts the extra
* header into the frame length calculation.

* lengthFieldOffset = 0
* lengthFieldLength = 3
* <b>lengthAdjustment</b> = <b>2</b> (= the length of Header 1)
* initialBytesToStrip = 0

* BEFORE DECODE (17 bytes) AFTER DECODE (17 bytes)
* +-------------------------------------+-------------------------------------+
* |  Length | Header 1 | Actual Content |----->|  Length | Header 1 | Actual Content |
* | 0x00000C | 0xCAFE  | "HELLO, WORLD" |      | 0x00000C | 0xCAFE  | "HELLO, WORLD" |
* +-------------------------------------+-------------------------------------+
* </pre>

* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,
     strip the first header field and the length field</h3>

* This is a combination of all the examples above. There are the prepended
* header before the length field and the extra header after the length field.
* The prepended header affects the <tt>lengthFieldOffset</tt> and the extra
* header affects the <tt>lengthAdjustment</tt>. We also specified a non-zero
* <tt>initialBytesToStrip</tt> to strip the length field and the prepended
* header from the frame. If you don't want to strip the prepended header, you
* could specify <tt>0</tt> for <tt>initialBytesToSkip</tt>.

* lengthFieldOffset = 1 (= the length of HDR1)
* lengthFieldLength = 2
* <b>lengthAdjustment</b> = <b>1</b> (= the length of HDR2)
* <b>initialBytesToStrip</b> = <b>3</b> (= the length of HDR1 + LEN)

* BEFORE DECODE (16 bytes) AFTER DECODE (13 bytes)
* +-------------------------------+------------------+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Content |
* +-------------------------------+------------------+
2 bytes length field at offset 1 in the middle of 4 bytes header,
strip the first header field and the length field, the length field
represents the length of the whole message

Let's give another twist to the previous example. The only difference from
the previous example is that the length field represents the length of the
whole message instead of the message body, just like the third example.
We have to count the length of HDR1 and Length into <tt>lengthAdjustment</tt>.
Please note that we don't need to take the length of HDR2 into account
because the length field already includes the whole header length.

lengthFieldOffset = 1
lengthFieldLength = 2
<br>lengthAdjustment = <b>-3</b> (= the length of HDR1 + LEN, negative)
initialBytesToStrip = <b>3</b>

BEFORE DECODE (16 bytes) AFTER DECODE (13 bytes)
+----------+--------+----------+----------------+------+
| HDR1 | Length | HDR2 | Actual Content | HDR2 | Actual Content |
| 0xCA | 0x10   | 0xFE  | "HELLO, WORLD" | 0xFE | "HELLO, WORLD" |
+----------+--------+----------+----------------+------+
<br>initialBytesToStrip = <b>3</b>

@see LengthFieldPrepender
Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/protobuf/ProtobufVarint32LengthFieldPrepender.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/HeadersUtils.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/protobuf/ProtobufVarint32FrameDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/ProtocolDetectionState.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/ProtocolDetectionResult.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/protobuf/ProtobufDecoderNano.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/protobuf/ProtobufEncoderNano.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/protobuf/ProtobufDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/protobuf/ProtobufEncoder.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

/*
 * Written by Robert Harder and released to the public domain, as explained at
 * http://creativecommons.org/licenses/publicdomain
 */

** Enumeration of supported Base64 dialects.
  * <p>
  * The internal lookup tables in this class has been derived from
*

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/base64/Base64Dialect.java

No license file was found, but licenses were detected in source scan.

/*
 *
Copyright 2014 The Netty Project
 *
The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* http://www.apache.org/licenses/LICENSE-2.0
 *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/compression/MessageAggregationException.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/compression/LzmaFrameEncoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/compression/LzmaFrameDecoder.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/compression/Crc32.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/compression/Bzip2BlockCompressor.java
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-jar/io/netty/handler/codec/compression/Lz4FrameDecoder.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 */
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/
* Written by Robert Harder and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
*/
/**
** Utility class for {@link ByteBuf} that encodes and decodes to and from
* <a href="http://en.wikipedia.org/wiki/Base64">Base64</a> notation.
* <p>
* The encoding and decoding algorithm in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/">Robert Harder's Public Domain
* Base64 Encoder/Decoder</a>.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871266_1595214869.21/0/netty-codec-4-1-42-final-sources-1-
  jar/io/netty/handler/codec/base64/Base64.java

1.879 to-regex-range 2.1.1
1.879.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

Open Source Used In DNAC Platform Antelope 2965
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.880 boost 1.60

1.880.1 Available under license:

Use of this software is granted under one of the following two licenses, to be chosen freely by the user.


================================================================================
Copyright (c) 2006, 2007 Marcin Kalicinski

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2. The MIT License

================================================================================
Copyright (c) 2006, 2007 Marcin Kalicinski

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===========================================================================

Copyright (c) 2007-2011 Barend Gehrels, Amsterdam, the Netherlands.
Copyright (c) 2008-2011 Bruno Lalande, Paris, France.
Copyright (c) 2009-2011 Mateusz Loskot, London, UK.

Use, modification and distribution is subject to the Boost Software License,
Version 1.0. (See accompanying file LICENSE_1_0.txt or copy at
http://www.boost.org/LICENSE_1_0.txt)

============================================================================

The default copyright note for C++ source files reads:

// Boost.Geometry (aka GGL, Generic Geometry Library)

// Copyright (c) 2007-2011 Barend Gehrels, Amsterdam, the Netherlands.
// Copyright (c) 2008-2011 Bruno Lalande, Paris, France.
// Copyright (c) 2009-2011 Mateusz Loskot, London, UK.

// Parts of Boost.Geometry are redesigned from Geodan's Geographic Library
// (geolib/GGL), copyright (c) 1995-2010 Geodan, Amsterdam, the Netherlands.

// Use, modification and distribution is subject to the Boost Software License,
// Version 1.0. (See accompanying file LICENSE_1_0.txt or copy at
// http://www.boost.org/LICENSE_1_0.txt)

Exceptions:
1) Major work of any author -> might change order, change date
2) Exclusive work of one author including design -> might take sole copyright
Examples:
- adapted geometry types e.g. array, Boost.Polygon
- implemented file-format e.g. WKB
- implemented specific strategy or algorithm e.g. intersections
3) Examples -> might take sole copyright
4) Tests -> might take sole copyright
5) Docs -> might take sole copyright
6) Utilities e.g. converters -> might take sole copyright
The copyright note for other (source) files as .py, Jamfiles, etc is similar and comments are changed accordingly.

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE article PUBLIC "-//Boost//DTD BoostBook XML V1.0//EN" "http://www.boost.org/tools/boostbook.dtd">
<article id="copyright_test" last-revision="DEBUG MODE Date: 2000/12/20 12:00:00 $" xmlns:xi="http://www.w3.org/2001/XInclude">
<title>Copyright Test</title>
<articleinfo>
<copyright>
<year>1963</year> <year>1964</year> <year>1965</year> <holder>Jane Doe</holder>
</copyright>
<copyright>
<year>2018</year> <holder>Joe Blow, John Coe</holder>
</copyright>
<copyright>
<year>1977</year> <year>1985</year> <holder>Someone else</holder>
</copyright>
</articleinfo>
<article>
This package was debianized by Vladimir Prus <ghost@cs.msu.su> on Wed, 17 July 2002, 19:27:00 +0400.

Copyright:

/+\n+\Copyright 1993-2002 Christopher Seiwald and Perforce Software, Inc.\+/

This is Release 2.4 of Jam/MR, a make-like program.

License is hereby granted to use this software and distribute it freely, as long as this copyright notice is retained and modifications are clearly marked.

ALL WARRANTIES ARE HEREBY DISCLAIMED.

Some portions are also:

Copyright 2001-2006 David Abrahams.
Copyright 2002-2006 Rene Rivera.
Copyright 2003-2006 Vladimir Prus.

Distributed under the Boost Software License, Version 1.0.
(See accompanying file LICENSE_1_0.txt or http://www.boost.org/LICENSE_1_0.txt)
This is an attempt to acknowledge early contributions to the garbage collector. Later contributions should instead be mentioned in README.changes.

HISTORY -

Early versions of this collector were developed as a part of research projects supported in part by the National Science Foundation and the Defense Advance Research Projects Agency.

The garbage collector originated as part of the run-time system for the Russell programming language implementation. The first version of the garbage collector was written primarily by Al Demers. It was then refined and mostly rewritten, primarily by Hans-J. Boehm, at Cornell U., the University of Washington, Rice University (where it was first used for C and assembly code), Xerox PARC, SGI, and HP Labs. However, significant contributions have also been made by many others.

Some other contributors:

More recent contributors are mentioned in the modification history in README.changes. My apologies for any omissions.

The SPARC specific code was originally contributed by Mark Weiser. The Encore Multimax modifications were supplied by Kevin Kenny (kenny@m.cs.uiuc.edu). The adaptation to the IBM PC/RT is largely due to Vernon Lee, on machines made available to Rice by IBM. Much of the HP specific code and a number of good suggestions for improving the generic code are due to Walter Underwood. Robert Brazile (brazile@diamond.bbn.com) originally supplied the ULTRIX code. Al Dosser (dosser@src.dec.com) and Regis Cridlig (Regis.Cridlig@cl.cam.ac.uk) subsequently provided updates and information on variation between ULTRIX systems. Parag Patel (parag@netcom.com) supplied the A/UX code. Jesper Peterson (jep@mitami.mtia.oz.au), Michel Schinz, and Martin Tauchmann (martintauchmann@bigfoot.com) supplied the Amiga port. Thomas Funke (thf@zelator.in-berlin.de(?) and Brian D.Carlstrom (bdc@clark.lcs.mit.edu) supplied the NeXT ports. Douglas Steel (doug@wg.icl.co.uk) provided ICL DRS6000 code. Bill Janssen (janssen@parc.xerox.com) supplied the SunOS dynamic loader specific code. Manuel Serrano (serrano@cornas.inria.fr) supplied linux and Sony News specific code. Al Dosser provided Alpha/OSF/1 code. He and Dave Detlefs(detlefs@src.dec.com) also provided several generic bug fixes. Alistair G. Crooks(agic@uts.amdahl.com) supplied the NetBSD and 386BSD ports. Jeffrey Hsu (hsu@soda.berkeley.edu) provided the FreeBSD port. Brent Benson (brent@jade.ssd.csd.harris.com) ported the collector to a Motorola 88K processor running CX/UX (Harris NightHawk). Ari Huttunen (Ari.Huttunen@hut.fi) generalized the OS/2 port to nonIBM development environments (a nontrivial task).
Patrick Beard (beard@cs.ucdavis.edu) provided the initial MacOS port. David Chase, then at Olivetti Research, suggested several improvements. Scott Schwartz (schwartz@groucho.cse.psu.edu) supplied some of the code to save and print call stacks for leak detection on a SPARC. Jesse Hull and John Ellis supplied the C++ interface code. Zhong Shao performed much of the experimentation that led to the current typed allocation facility. (His dynamic type inference code hasn’t made it into the released version of the collector, yet.)

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Software License, Version 1.0

Copyright 2002-2003, Trustees of Indiana University.
Copyright 2000-2001, University of Notre Dame.
All rights reserved.

Indiana University has the exclusive rights to license this product under the following license.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* All redistributions of source code must retain the above copyright notice, the list of authors in the original source code, this list of conditions and the disclaimer listed in this license;

* All redistributions in binary form must reproduce the above copyright notice.
notice, this list of conditions and the disclaimer listed in this license in the documentation and/or other materials provided with the distribution;

* Any documentation included with all redistributions must include the following acknowledgement:

"This product includes software developed at the University of Notre Dame and the Pervasive Technology Labs at Indiana University. For technical information contact Andrew Lumsdaine at the Pervasive Technology Labs at Indiana University. For administrative and license questions contact the Advanced Research and Technology Institute at 351 West 10th Street. Indianapolis, Indiana 46202, phone 317-278-4100, fax 317-274-5902."

Alternatively, this acknowledgement may appear in the software itself, and wherever such third-party acknowledgments normally appear.

* The name Indiana University, the University of Notre Dame or "Caramel" shall not be used to endorse or promote products derived from this software without prior written permission from Indiana University. For written permission, please contact Indiana University Advanced Research & Technology Institute.

* Products derived from this software may not be called "Caramel", nor may Indiana University, the University of Notre Dame or "Caramel" appear in their name, without prior written permission of Indiana University Advanced Research & Technology Institute.

Indiana University provides no reassurances that the source code provided does not infringe the patent or any other intellectual property rights of any other entity. Indiana University disclaims any liability to any recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise.

LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR OTHER PROPRIETARY RIGHTS. INDIANA UNIVERSITY MAKES NO WARRANTIES THAT SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP DOORS", "WORMS", OR OTHER HARMFUL CODE. LICENSEE ASSUMES THE ENTIRE RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS, AND TO THE PERFORMANCE AND VALIDITY OF INFORMATION GENERATED USING SOFTWARE. The following people hereby grant permission to replace all existing licenses on their contributions to Boost with the Boost Software License, Version 1.0. (boostinspect:nolicense boostinspect:nocopyright)

Aleksey Gurtovoy (agurtovoy@meta-comm.com)
Andrei Alexandrescu (andrewalex - at - hotmail.com) (See Boost list message of August 12, 2004 11:06:58 AM EST)
Open Source Used in DNAC Platform Antelope

Andrew Lumsdaine ()
Anthony Williams (anthony -at- justsofwareobolutions.co.uk)
Beman Dawes (bdawes@acm.org)
Brad King (brad.king -at- kitware.com) (See Boost list message of Wed, 21 Jul 2004 11:15:46 -0400)
Brian Osman (osman -at- vvisions.com) (See CVS log)
Bruce Barr (schmoost -at- yahoo.com) (See Boost list of Mon, 16 Aug 2004 15:06:43 -0500)
Bruno da Silva de Oliveira (bruno - at - esss.com.br)
Christain Engstrom (christian.engstrom -at- glindra.org) (See Boost list message of Mon, 30 Aug 2004 14:31:49 +0200)
Cromwell D Enage (sponage -at- yahoo.com) (See Boost list message of August 12, 2004 11:49:13 AM EST)
Dan Gohman (dig -at- cray.com) (See Boost list message of Sat, 21 Aug 2004 10:54:59 +0100)
Tom Nuffer (dan -at- nuffer.name)
Daniel Frey (d.frey -at- gmx.de, daniel.frey -at- aixigo.de)
Daniel Nuffer (dan -at- nuffer.name)
Darin Adler (darin -at- bentspoon.com) (Email to Andreas Huber, see change log)
Daryle Walker (darylew - at - hotmail.com)
Dave Abrahams (dave@boost-consulting.com)
Dave Moore (dmoore -at- viefinancial.com) (See Boost list message of 18 Dec 2003 15:35:50 -0500)
David Abrahams (dave@boost-consulting.com)
Dietmar Kuehl (dietmar_kuehl -at- yahoo.com) (Email to Andreas Huber, see change log)
Douglas Gregor (gregod -at- cs.rpi.edu, dgregor -at- cs.indiana.edu, doug.gregor -at- gmail.com)
Dr John Maddock (john - at - johnmaddock.co.uk)
Edward D. Brey (brey -at- ductape.net) (Email to Andreas Huber, see change log)
Eric Ford (un5o6n902 -at- sneakemail.com) (See Boost list message of Sun, 15 Aug 2004 10:29:13 +0100)
Eric Friedman (efb@users.sourceforge.net)
Eric Niebler (eric@boost-consulting.com)
Fernando Cacciola (fernando_cacciola@ciudad.com.ar)
Fernando Luis Cacciola Carballal (fernando_cacciola@ciudad.com.ar)
Francois Faure (Francois.Faure -at- imag.fr) (See CVS log)
Gary Powell (powellg - at - amazon.com) (See Boost list message of 10 Feb 2004 14:22:46 -0800)
Gennadiy Rozental (rogeeff -at- mail.com) (Email to Andreas Huber, see change log)
Gottfried Ganssauge (Gottfried.Ganssauge -at- HAUF.DE) (See Boost List message of Mon, 16 Aug 2004 10:09:19 +0200)
Gottfried Ganzaige (Gottfried.Ganssauge -at- HAUF.DE) (Alternative spelling of Gottfried Ganssauge)
Greg Colvin (gregory.colvin -at- oracle.com) (See Boost list message of Sat, 14 Aug 2004 10:57:00 +0100)
Gregory Colvin (gregory.colvin -at- oracle.com) (See Boost list message of Sat, 14 Aug 2004 10:57:00 +0100)
Gunter Winkler (gunter.winkler -at- unibw-muenchen.de) (See Boost List message of Mon, 16 Aug 2004 10:24:17 +0200)
Hartmut Kaiser (hartmut.kaiser -at- gmail.com)
Herve Bronnimann (hbr -at- poly.edu)
Herv Brnnimann (hbr -at- poly.edu)
Housemarque Oy (ilari.kuittinen ilari.kuittinen -at- housemarque.fi)
Howard Himnant (hinnant -at- twcy.rr.com) (See Boost list message of July 25, 2004 3:44:49 PM EST)
Hubert Holin (hubert_holin -at- users.sourceforge.net)
Indiana University ()
Itay Maman (imaman -at- users.sourceforge.net)
Jaakko Jrvi (jjarvi -at- osl.iu.edu)
Jaap Suter (j.suter -at- student.utwente.nl) (See Boost list message of Thu, 16 Sep 2004 09:32:43 -0700)
Open Source Used In DNAC Platform Antelope 2973
1.881 websocket-common 9.3.27.v20190418

1.882 babel-plugin-syntax-object-rest-spread
6.13.0

1.882.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The "Software identified by reference to the Ecma Standard* ("Software")" is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.883 aws-java-sdk-sts 1.11.375
1.883.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * 
 * http://aws.amazon.com/apache2.0
 * 
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
/*opt/cola/permissions/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/auth/SessionCredentialsHolder.java
*/opt/cola/permissions/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/auth/RefreshableTask.java
*/opt/cola/permissions/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/auth/ShouldDoAsyncSessionRefresh.java
*/opt/cola/permissions/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/auth/DaemonThreadFactory.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
*/

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013-2018 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
*/

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012-2018 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 *
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* http://aws.amazon.com/apache2.0
*
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
  jar/com/amazonaws/auth/SessionCredentialsProviderFactory.java
No license file was found, but licenses were detected in source scan.

/
* Copyright 2011-2018 Amazon Technologies, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* http://aws.amazon.com/apache2.0
*
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
  jar/com/amazonaws/auth/STSSessionCredentials.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
  jar/com/amazonaws/auth/STSSessionCredentialsProvider.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
  jar/com/amazonaws/auth/NoSessionSupportCredentials.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
  jar/com/amazonaws/auth/STSAssumeRoleSessionCredentialsProvider.java
No license file was found, but licenses were detected in source scan.

/
* Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
with
* the License. A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR
* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
permissions
* and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/transform/DecodeAuthorizationMessageResultStaxUnmarshaller.ja
va
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceAsyncClientBuilder.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/transform/IDPRejectedClaimExceptionUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/GetFederationTokenRequest.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/AssumeRoleWithWebIdentityRequest.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/RegionDisabledException.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleResultStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/GetCallerIdentityResult.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/GetSessionTokenRequest.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/package-info.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/transform/CredentialsStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/package-info.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleResultStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/GetCallerIdentityResult.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/GetSessionTokenRequest.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/GetCallerIdentityResult.java
* /opt/cola/permits/1180370927_1626285908.9/0/aws-java-sdk-sts-1-11-375-sources-1-
jar/com/amazonaws/services/securitytoken/model/DecodeAuthorizationMessageResult.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleWithSAMLResultStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceClient.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/GetCallerIdentityResultStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceAsyncClient.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/GetSessionTokenResultStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/AbstractAWSSecurityTokenServiceAsync.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/FederatedUserStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/ExpiredTokenExceptionUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/InvalidIdentityTokenExceptionUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/PackedPolicyTooLargeException.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/AssumedRoleUserStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/DelegateSessionResultStaxUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/ExpiredTokenExceptionUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/InvalidIdentityTokenExceptionUnmarshaller.java
* /opt/cola/permits/1180370927_1626285908.90/aws-java-sdk-sts-1-11-375-sources-1.jar/com/amazonaws/services/securitytoken/model/transform/PolicyTooLargeException.java

1.884 har-schema 2.0.0

Open Source Used In DNAC Platform Antelope 2882
1.884.1 Available under license:
Copyright (c) 2015, Ahmad Nassri <ahmad@ahmadnassri.com>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.885 stream-to-array 2.3.0
1.885.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.886 jgrapht-core 1.3.0
1.886.1 Available under license:

Copyright (c) 2001-2014, JGraph Ltd
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the JGraph nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JGRAPH BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Contributors ##
Copyright Notice: The JGraphT project source code is a composite of contributions by multiple authors. The copyright for each contribution is owned by the corresponding author. For details, please see [HISTORY.md](HISTORY.md) as well as the complete git history. All authors have agreed to license their contributions to the public as open source under the specific terms noted in [README.md](README.md).

JGraphT wouldn't be the library it is today without the source contributions and suggestions made by the authors:

- [Barak Naveh](http://sourceforge.net/users/barak_naveh/) (project founder)
- [John V Sichi](https://github.com/jsichi) (current project administrator)
- [Joris Kinable](https://github.com/jkinable) (JGraphT Project Reviewer/Committer and Release Manager)
- [Dimitrios Michail](https://github.com/d-michail) (JGraphT Project Reviewer/Committer)
- [Liviu Rau](http://sourceforge.net/users/liviu_aurelian/)
- [Nathan Fiedler](http://www.bluemarsh.com/personal/index.html)
- [Michael Behrisch](http://sourceforge.net/users/behrisch/)
- [Linda Buisman](http://sourceforge.net/users/linda_buisman/)
- Erik Postma
- Mikael Hansen
- Avner Linder
- Marden Neubert
- [Christian Soltenborn](http://sourceforge.net/users/csoltenborn/)
- [Christian Hammer](http://sourceforge.net/users/hammerc/)
- Ewgenij Proschak
- [Hartmut Benz](http://sourceforge.net/users/ivins/)
- [Charles Fry](http://frogcircus.org/)
- Guillaume Boulmier
- Carl Anderson
- Khanh Vu
- Aaron Harnly
- Welson Sun
- Trevor Harmon
- David Black-Schaffer
- Vinayak Borkar
- Andrew Berman
- Lucas Scharenbroich
- Hookahey
- Tim Shearouse
- Holger Brandl
- Ilya Razenshtein
- Peter Giles
- Andrew Newell
- Tim Engler
- Tom Larkworthy
- Soren Davidsen
- Andrea Pagani
- Tom Conerly
- Michele Mancioppi
- Adrian Marte
- [Assaf Mizrachi](https://github.com/assimiz)
- Harshal Vora
- Matt Sarjent
- Robby McKilliam
- Yuriy Nakonechnyy
- Andreas Schnaiter
- Owen Jacobson
- Alejandro R. Lopez del Huerto
- Vladimir Kostyukov
- Ernst de Ridder
- Michal Pasieka
- Alexey Kudinkin
- Adam Gouge
- Nikolay Ognyanov
- Graham Hill (AzrgExplorers)
- Leo Crawford
- Isaac Kleinman
- Sebastian Hubenschmid
- JeanYves Tinevez
- Oliver Kopp
- Javier Gutierrez (javierj)
- Nicolas Fortin
- Peter Goldstein
- Rodrigo López Dato
- Anders Wallgren
- Siarhei
- Jan Altenbernd
- Andrew Chen
- Florian Buenzli
- Thomas Tschager
- Tomas Hruz
- Philipp Hoppen
- Chris Wensel
- Wil Selwood
- Mihhail Verhovtsov
- Fabian Sph
- Rita Dobler
- Szabolcs Besenyei
- Luiz Kill
- Christophe Thiebaud
- Jon Robinson
- Thomas Breitbart
- Sarah Komla-Elbri
- Graeme Ahokas
- Christoph Zauner
- [Andrew Gainer-Dewar](https://github.com/agdphd)
- Benedikt Waldvogel
- Victor Mikhailov
- Nils Olberg
- [Daniel Gomez-Sanchez](https://github.com/magicDGS)
- [Skuratovich Sergey](https://github.com/SSNikolaevich)
- [Martin Sturm](https://github.com/WorstCase00)
- [Patrick Sharp](https://github.com/sharpTrick)
- [Piotr Turski](https://github.com/piotrturski)
- [Alexandru Vleanu](https://github.com/AlexandruVleanu)
- [Davide Cavestro](https://github.com/davidecavestro)
- [Mark Raynsford](https://github.com/io7m)
- [Mariusz Smykua](https://github.com/mariuszs)
- [Pratik Tibrewal](https://github.com/tibrewalpratik17)
- [Chen Kui](https://github.com/Yimismi)
- [Timofey Chudakov](https://github.com/Toptachamann)
- [Konstantinos Karatsenidis](https://github.com/gate2k1)
- [Kirill Vishnyakov](https://github.com/LightnessOfBeing)
- [Emilio Cruciani](https://github.com/ioemilio)
- [Vivek Talreja](https://github.com/Vivek1012)
- [Gilles Gosuin](https://github.com/gilles-gosuin)
- [Viktor Volkov](https://github.com/chupacabra007)
- [Philipp Kaesgen](https://github.com/PhilippKaesgen)
- [Lukas Harzenetter](https://github.com/lharzenetter)
- [Christoph Grune](https://github.com/christophgruene)

---

**Open Source Used In DNAC Platform Antelope** 2016
- [Daniel Mock](https://github.com/danielmock)
- [Oliver Feith](https://github.com/Watercrystal)
- [Abdallah Atouani](https://github.com/AbdallahAt)
- [Peter Harman](https://github.com/harmanpa)
- [Nikhil Sharma](https://github.com/nks1558)
- [Dennis Fischer](https://github.com/pdelvo)
- [PHaroZ](https://github.com/PHaroZ)

(if we have missed your name on this list, please email us to get it fixed).

Other people have also helped in different ways: reporting bugs, requesting features, commenting, and by merely asking very good questions.

Many thanks to all of you.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.
To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that,
in the event an application does not supply such function or
table, the facility still operates, and performs whatever part of
its purpose remains meaningful.

(For example, a function in a library to compute square roots has
a purpose that is entirely well-defined independent of the
application. Therefore, Subsection 2d requires that any
application-supplied function or table used by this function must
be optional: if the application does not supply it, the square
root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.
4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).
To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
   Distributed under this Agreement,
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.
2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:
a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) does not attempt to limit or alter the recipients’ rights in the Source Code under section 3.2; and
   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (“notices”) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities
with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.
Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

[The "BSD 3-clause license"]
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=======

MIT License for codepointat.js from https://git.io/codepointat
MIT License for fromcodepoint.js from https://git.io/vDW1m

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.887 for-own 0.1.5

1.887.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2015, 2017, Jon Schlinkert
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.888 vary 1.1.2

1.888.1 Available under license:
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.889 twill-discovery-api 0.6.0-incubating

1.889.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137747599_1614121902.52/0/twill-discovery-api-0-6-0-incubating-sources-2-
  jar/org/apache/twill/discovery/DiscoveryServiceClient.java
* /opt/cola/permits/1137747599_1614121902.52/0/twill-discovery-api-0-6-0-incubating-sources-2-
  jar/org/apache/twill/discovery/ServiceDiscovered.java
* /opt/cola/permits/1137747599_1614121902.52/0/twill-discovery-api-0-6-0-incubating-sources-2-
  jar/org/apache/twill/discovery/DiscoveryService.java
* /opt/cola/permits/1137747599_1614121902.52/0/twill-discovery-api-0-6-0-incubating-sources-2-
  jar/org/apache/twill/discovery/Discoverable.java

1.890 libdrm 2.4.91 2~16.04.1

1.891 jackson-jaxrs-base 2.9.10

1.891.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:
1.892 js-tokens 3.0.2

1.892.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014, 2015, 2016, 2017 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.893 p-try 1.0.0

1.893.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derived Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derived Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.895 glib 2.48.2-0ubuntu1

1.895.1 Available under license:

This work may be reproduced and distributed in whole or in part, in any medium, physical or electronic, so as long as this copyright notice remains intact and unchanged on all copies. Commercial redistribution is permitted and encouraged, but you may not redistribute, in whole or in part, under terms more restrictive than those under which you received it. If you redistribute a modified or translated version of this work, you must also make the source code to the modified or translated version available in electronic form without charge. However, mere aggregation as part of a larger work shall not count as a modification for this purpose.

All code examples in this work are placed into the public domain, and may be used, modified and redistributed without restriction.

BECAUSE THIS WORK IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE WORK, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE WORK "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SHOULD THE WORK PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.
IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE WORK AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE WORK, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

PCRE LICENCE

Please see the file LICENCE in the PCRE distribution for licensing details.

End

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library’s complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or
table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).
To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.896 normalize-package-data 2.5.0

1.896.1 Available under license :

This package contains code originally written by Isaac Z. Schlueter. Used with permission.

Copyright (c) Meryn Stol ("Author")  
All rights reserved.

The BSD License
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.897 chai 4.3.4

1.897.1 Available under license :

MIT License

Copyright (c) 2017 Chai.js Assertion Library

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.898 object-visit 1.0.1

1.898.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015, 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.899 backport-util-concurrent 3.1

1.899.1 Available under license:
The backport-util-concurrent software has been released to the public domain, as explained at: http://creativecommons.org/licenses/publicdomain.

Acknowledgements: backport-util-concurrent is based in large part on the public domain sources from:
1) JSR166,
2) package dl.util.concurrent,
3) Doug Lea's "collections" package.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html><head><title>Creative Commons Public Domain</title></head>
<body>
<p align="center"><em>Copyright-Only Dedication (based on United States law) or Public Domain Certification</em></p>
<p>The
person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.</p>

<p>A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.</p>

<p>Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.</p>

<p>Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.</p>

1.900 set-blocking 2.0.0

1.900.1 Available under license:

Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.901 ws 1.1.1 1.1.1
1.901.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.902 iceberg-common 0.8.0-incubating
1.902.1 Available under license:

Apache Iceberg (incubating)
Copyright 2017-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--------------------------------------------------------------------------------
This project includes code from Kite, developed at Cloudera, Inc. with
the following copyright notice:

| Copyright 2013 Cloudera Inc. |
| Licensed under the Apache License, Version 2.0 (the "License"); |
| you may not use this file except in compliance with the License. |
| You may obtain a copy of the License at |
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes a gradle wrapper.

* gradlew and gradle/wrapper/gradle-wrapper.properties

Copyright: 2010-2019 Gradle Authors.
Home page: https://github.com/gradle/gradle
License: http://www.apache.org/licenses/LICENSE-2.0

This product includes code from Apache Avro.

* Conversion in DecimalWriter is based on Avro's Conversions.DecimalConversion.

Home page: https://parquet.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

This product includes code from Apache Parquet.

* DynMethods.java
* DynConstructors.java
* AssertHelpers.java

This product includes code from Cloudera Kite.

* SchemaVisitor and visit methods

Copyright: 2013-2017 Cloudera Inc.
Home page: https://kitesdk.org/
License: http://www.apache.org/licenses/LICENSE-2.0

This product includes code from Presto.

* Retry wait and jitter logic in Tasks.java

Copyright: 2016 Facebook and contributors
Home page: https://prestodb.io/
License: http://www.apache.org/licenses/LICENSE-2.0

This product includes code from Apache iBATIS.

* Hive ScriptRunner.java

Copyright: 2004 Clinton Begin
Home page: http://ibatis.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

This product includes code from Apache Hive.

* Hive metastore derby schema in hive-schema-3.1.0.derby.sql

Copyright: 2011-2018 The Apache Software Foundation
Home page: http://hive.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

This product includes code from Apache Spark.

* dev/check-license script
1.903 strip-eof 1.0.0

1.903.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindresorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.904 shared-mime-info 1.5 2ubuntu0.2

1.904.1 Available under license:
This package was debianized by Filip Van Raemdonck (mechanix@debian.org) on Thu, 4 Jul 2002 20:59:23 +0200.

It was downloaded from http://freedesktop.org/wiki/Software/shared-mime-info.

Copyright:
Copyright (C) 2002, 2003 Thomas Leonard.

License:
This package is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This package is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this package; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

On Debian systems, the complete text of the GNU General
Public License can be found in `/usr/share/common-licenses/GPL'.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.905 zlib 1.2.5

1.905.1 Available under license:
/* zlib.h -- interface of the 'zlib' general purpose compression library version 1.2.11, January 15th, 2017

Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu
*/
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.906 jersey-media-jaxb 2.32
1.906.1 Available under license:

# Notice for Jersey
This content is produced and maintained by the Eclipse Jersey project.

* Project home: https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such
availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

Angular JS, v1.6.6
* License MIT (http://www.opensource.org/licenses/mit-license.php)
* Project: http://angularjs.org
* Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1
* License: all the source code provided by AOP Alliance is Public Domain.
* Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2
* License: Apache License, 2.0
* Project: http://beanvalidation.org/1.1/
* Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag.

Hibernate Validator CDI, 6.1.2.Final
* License: Apache License, 2.0
* Project: https://beanvalidation.org/
* Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

Bootstrap v3.3.7
* License: MIT license (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* Project: http://getbootstrap.com
* Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0
* License: Apache License, 2.0
* Copyright (C) 2009 The Guava Authors

javax.inject Version: 1
* License: Apache License, 2.0
* Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA
Jackson JAX-RS Providers Version 2.10.1
* License: Apache License, 2.0
* Project: https://github.com/FasterXML/jackson-jaxrs-providers
* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4
* License: jquery.org/license
* Project: jquery.org
* Copyright: (c) jQuery Foundation

jQuery Barcode plugin 0.3
* Project: http://www.pasella.it/projects/jQuery/barcode
* Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266
* License: CC0
* No copyright
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

KineticJS, v4.7.1
* License: MIT license (http://www.opensource.org/licenses/mit-license.php)
* Copyright: Eric Rowell

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0
* License: Apache License, 2.0
* Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents
* License: W3C License
* Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
   Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program.
or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient’s receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and
b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT
PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement,
but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.
The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

  c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and
If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the
author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

---

## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is
making a combined work based on this library. Thus, the terms and
conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.907 @babel/plugin-syntax-json-strings 7.8.3

1.907.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.908 asm 5.0.3
1.908.1 Available under license:

No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2011 INRIA, France Telecom
 * All rights reserved.
 * * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 * contributors may be used to endorse or promote products derived from
 * this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */
/**
 * Creates a new {@link AnalyzerAdapter}. <i>Subclasses must not use this
 * constructor</i>. Instead, they must use the
 * { @link #AnalyzerAdapter(int, String, int, String, String, MethodVisitor) }
 * version.
 *
 * @param owner
 * the owner's class name.
 *
 */
* be <tt>null</tt>.
* @throws IllegalStateException
* If a subclass calls this constructor.
*/

Found in path(s):
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-
  jar/org/objectweb/asm/commons/AnalyzerAdapter.java
No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
 * Creates a new [@link GeneratorAdapter]. <i>Subclasses must not use this
 * constructor</i>. Instead, they must use the
 * [@link #GeneratorAdapter(int, MethodVisitor, int, String, String)]
 * version.
 *
 * @param mv
 * the method visitor to which this adapter delegates calls.
 * @param access
* the method's access flags (see [@link Opcodes]).
* @param name
* the method's name.
* @param desc
* the method's descriptor (see [@link Type Type]).
* @throws IllegalStateException
* If a subclass calls this constructor.
*/

Found in path(s):
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/commons/GeneratorAdapter.java
No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2013 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*    
*    THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
*    AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
*    IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
*    ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
*    LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
*    CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
*    SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
*    INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
*    CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
*    ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
*    THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/TypePath.java
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/TypeReference.java
No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2011 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */
/**
 * Creates a new JSRInliner. <i>Subclasses must not use this
 * constructor</i>. Instead, they must use the
 * [@link JSRInlinerAdapter(int, MethodVisitor, int, String, String, String, String[])]
 * version.
 *
 * @param mv
 * the <code>MethodVisitor</code> to send the resulting inlined
 * method code to (use <code>null</code> for none).
 *
 * @param access
 * the method's access flags (see [@link Opcodes]). This
 * parameter also indicates if the method is synthetic and/or
 * deprecated.
 *
 * @param name
 * the method's name.
 *
 * @param desc
 * the method's descriptor (see [@link Type]).
* @param signature
* the method's signature. May be null.
* @param exceptions
* the internal names of the method's exception classes (see
* { @link Type#getInternalName() getInternalName }). May be
* null.
* @throws IllegalStateException
* If a subclass calls this constructor.
*/

/**
 * Constructs a new { @link LocalVariableAnnotationNode }. Subclasses must
 * not use this constructor. Instead, they must use the
 * { @link #LocalVariableAnnotationNode(int, TypePath, LabelNode[], LabelNode[], int[], String)}
 */
* version.
* 
* @param typeRef
*    a reference to the annotated type. See {@link TypeReference}.
* @param typePath
*    the path to the annotated type argument, wildcard bound, array
*    element type, or static inner type within 'typeRef'. May be
*    <tt>null</tt> if the annotation targets 'typeRef' as a whole.
* @param start
*    the first instructions corresponding to the continuous ranges
*    that make the scope of this local variable (inclusive).
* @param end
*    the last instructions corresponding to the continuous ranges
*    that make the scope of this local variable (exclusive). This
*    array must have the same size as the 'start' array.
* @param index
*    the local variable's index in each range. This array must have
*    the same size as the 'start' array.
* @param desc
*    the class descriptor of the annotation class.
*/

Found in path(s):
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-
  jar/org/objectweb/asm/tree/LocalVariableAnnotationNode.java
No license file was found, but licenses were detected in source scan.

2011, Eugene Kuleshov
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

2011 INRIA, France Telecom
* All rights reserved.

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb.asm/xml/asm-xml.dtd
No license file was found, but licenses were detected in source scan.
No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2011 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 * contributors may be used to endorse or promote products derived from
 * this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */
/**
 * Constructs a new {@link Textifier}. <i>Subclasses must not use this
 * constructor</i>. Instead, they must use the {@link #Textifier(int)}
 * version.
 *
 * @throws IllegalStateException
 *     If a subclass calls this constructor.
 */

Found in path(s):
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/util/Textifier.java

No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2011 INRIA, France Telecom
 * All rights reserved.
 */
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Constructs a new \texttt{ClassNode}. <i>Subclasses must not use this constructor</i>. Instead, they must use the \texttt{ClassNode(int)} version.

@throws IllegalStateException
If a subclass calls this constructor.

ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

No license file was found, but licenses were detected in source scan.

2011, Eugene Kuleshov
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
*/opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/xml/package.html
No license file was found, but licenses were detected in source scan.

/***
* ASM XML Adapter
* Copyright (c) 2004-2011, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
*/opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/xml/Processor.java
*/opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/xml/SAXClassAdapter.java
*/opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/xml/SAXAdapter.java
*/opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/xml/SAXCodeAdapter.java
*/opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/xml/ASMContentHandler.java

Open Source Used In DNAC Platform Antelope 3074
No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2011 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */

Found in path(s):
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/xml/SAXFieldAdapter.java
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/xml/SAXAnnotationAdapter.java
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/optimizer/ClassOptimizer.java
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/tree/analysis/Value.java
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/Label.java
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/Item.java
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/commons/StaticInitMerger.java
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/optimizer/MethodConstantsCollector.java
  * /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/optimizer/ClassOptimizer.java
No license file was found, but licenses were detected in source scan.

#All rights reserved.
#Redistribution and use in source and binary forms, with or without
#modification, are permitted provided that the following conditions
#are met:
#1. Redistributions of source code must retain the above copyright
#   notice, this list of conditions and the following disclaimer.
#2. Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in the
#   documentation and/or other materials provided with the distribution.
#3. Neither the name of the copyright holders nor the names of its
#   this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/optimizer/shrink.properties
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/optimizer/shrink-writer.properties
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/optimizer/shrink-resize.properties
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/optimizer/shrink-signatures.properties
* /opt/cola/permits/1135840046_1613613019.19/0/asm-5-0-3-sources-7-jar/org/objectweb/asm/optimizer/shrink-annotations.properties
1.909 os-homedir 1.0.2 1.0.2

1.909.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.910 tephra-core 0.6.0

1.910.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
1.911 regenerator-runtime 0.13.3 0.13.3
1.911.1 Available under license :
MIT License

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.912 bcel 2.7.2
1.912.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
package org.apache.xalan;

/**
 * Administrative class to keep track of the version number of
 * the Xalan release.
 * <p>This class implements the upcoming standard of having
 * org.apache.project-name.Version.getVersion() be a standard way
 * to get version information. This class will replace the older
 * org.apache.xalan.processor.Version class.</p>
 * <p>See also: org/apache/xalan/res/XSLTInfo.properties for
 * information about the version of the XSLT spec we support.</p>
 * @xsl.usage general
 */

public class Version {

/**
 * Get the basic version string for the current Xalan release.
 * Version String formatted like
 * Futurework: have this read version info from jar manifest.
 * @return String denoting our current version
 */

public static String getVersion() {
    return getProduct() + " " + getImplementationLanguage() + " "
            + getMajorVersionNum() + "." + getReleaseVersionNum() + ".";
    +( (getDevelopmentVersionNum() > 0) ?
       ("D" + getDevelopmentVersionNum()) : ("" + getMaintenanceVersionNum()) );
}

/**
 * Print the processor version to the command line.
 * @param argv command line arguments, unused.
 */

public static void main(String argv[]) {
    System.out.println(getVersion());
}
/**
 * Name of product: Xalan.
 */
public static String getProduct()
{
    return "Xalan";
}

/**
 * Implementation Language: Java.
 */
public static String getImplementationLanguage()
{
    return "Java";
}

/**
 * Major version number.
 * Version number. This changes only when there is a
 * significant, externally apparent enhancement from
 * the previous release. 'n' represents the n'th
 * version.
 * *
 * Clients should carefully consider the implications
 * of new versions as external interfaces and behaviour
 * may have changed.
 */
public static int getMajorVersionNum()
{
    return @version.VERSION@;
}

/**
 * Release Number.
 * Release number. This changes when:
 * - a new set of functionality is to be added, eg,
 *   implementation of a new W3C specification.
 * - API or behaviour change.
 * - its designated as a reference release.
 */
public static int getReleaseVersionNum()
{
    return @version.RELEASE@;
}
/**
 * Maintenance Drop Number.
 * Optional identifier used to designate maintenance
 * drop applied to a specific release and contains
 * fixes for defects reported. It maintains compatibility
 * with the release and contains no API changes.
 * When missing, it designates the final and complete
 * development drop for a release.
 */
public static int getMaintenanceVersionNum()
{
    return @version.MINOR@;
}

/**
 * Development Drop Number.
 * Optional identifier designates development drop of
 * a specific release. D01 is the first development drop
 * of a new release.
 * Development drops are works in progress towards a
 * completed, final release. A specific development drop
 * may not completely implement all aspects of a new
 * feature, which may take several development drops to
 * complete. At the point of the final drop for the
 * release, the D suffix will be omitted.
 * Each 'D' drops can contain functional enhancements as
 * well as defect fixes. 'D' drops may not be as stable as
 * the final releases.
 */
public static int getDevelopmentVersionNum()
{
    try {
        if (((new String("@version.DEVELOPER@"))).length() == 0)
            return 0;
        else
            return Integer.parseInt("@version.DEVELOPER@");
    } catch (NumberFormatException nfe) {
        return 0;
    }
}

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/Version.src
No license file was found, but licenses were detected in source scan.
/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/**
 * This is a special exception that is used to stop parsing when
 * search for an element. For instance, when searching for xml:stylesheet
 * PIs, it is used to stop the parse once the document element is found.
 * @see StylesheetPIHandler
 * @xsl.usage internal
 */

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
/*
 * /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/StopParseException.java
 * No license file was found, but licenses were detected in source scan.
 */
/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/*
 * $Id: Version.src 468654 2006-10-28 07:09:23Z minchau $
 */

package org.apache.xml.serializer;

/**
 * Administrative class to keep track of the version number of
 * the Serializer release.
 * This class implements the upcoming standard of having
 * org.apache.project-name.Version.getVersion() be a standard way
 * to get version information.<P>
 * @xsl.usage general
 */

public final class Version
{

/**
 * Get the basic version string for the current Serializer.
 * Version String formatted like
 * <CODE>"<B>Serializer</B> <B>Java</B> v.r[.dd| <B>D</B>nn"]</CODE>.  
 * Futurework: have this read version info from jar manifest.
 *
 * @return String denoting our current version
 */

public static String getVersion()
{
}

/**
 * Print the processor version to the command line.
 * @param argv command line arguments, unused.
 */
public static void main(String argv[])
{
    System.out.println(getVersion());
}

/**
 * Name of product: Serializer.
 */
public static String getProduct()
{
    return "Serializer";
}

/**
 * Implementation Language: Java.
 */
public static String getImplementationLanguage()
{
    return "Java";
}

/**
 * Major version number.
 * Version number. This changes only when there is a
 * significant, externally apparent enhancement from
 * the previous release. 'n' represents the n'th
 * version.
 * Clients should carefully consider the implications
 * of new versions as external interfaces and behaviour
 * may have changed.
 */
public static int getMajorVersionNum()
{
    return @version.VERSION@;
/**
 * Release Number.
 * Release number. This changes when:
 * - a new set of functionality is to be added, eg,
 *   implementation of a new W3C specification.
 * - API or behaviour change.
 * - its designated as a reference release.
 */
public static int getReleaseVersionNum()
{
    return @version.RELEASE@;
}

/**
 * Maintenance Drop Number.
 * Optional identifier used to designate maintenance
 * drop applied to a specific release and contains
 * fixes for defects reported. It maintains compatibility
 * with the release and contains no API changes.
 * When missing, it designates the final and complete
 * development drop for a release.
 */
public static int getMaintenanceVersionNum()
{
    return @version.MINOR@;
}

/**
 * Development Drop Number.
 * Optional identifier designates development drop of
 * a specific release. D01 is the first development drop
 * of a new release.
 *
 * Development drops are works in progress towards a
 * completed, final release. A specific development drop
 * may not completely implement all aspects of a new
 * feature, which may take several development drops to
 * complete. At the point of the final drop for the
 * release, the D suffix will be omitted.
 *
 * Each 'D' drops can contain functional enhancements as
 * well as defect fixes. 'D' drops may not be as stable as
 * the final releases.
 */
public static int getDevelopmentVersionNum()
{
try {
    if ((new String("@version.DEVELOPER@")).length() == 0)
        return 0;
    else
        return Integer.parseInt("@version.DEVELOPER@");
} catch (NumberFormatException nfe) {
    return 0;
}

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/Version.src
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the "License);
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
/**
 * Major version number.
 * Version number. This changes only when there is a
 * significant, externally apparent enhancement from
 * the previous release. 'n' represents the n'th
 * version.
 *
 * Clients should carefully consider the implications
 * of new versions as external interfaces and behaviour
 * may have changed.
 */

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/Version.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/processor/XSLProcessorVersion.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/Version.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
// Attribution to: "Voytenko, Dimitry" <DVoytenko@SECTORBASE.COM>

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/ToHTMLStream.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
// Proprietary
/** The 'document-location()' id (Proprietary). */

Found in path(s):
Open Source Used In DNAC Platform Antelope 3094

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xpath/compiler/FunctionTable.java
No license file was found, but licenses were detected in source scan.

# to you under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/serializer/Encodings.properties
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/serializer/XMLEntities.properties
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/serializer/output_text.properties
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/res/XSLTInfo.properties
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/serializer/output_xml.properties
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/serializer/output_unknown.properties
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/serializer/HTMLEntities.properties
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/serializer/output_html.properties
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package org.apache.xalan.xslt.compiler;

import java.util.Stack;
import java.util.Vector;
import java.io.StringReader;
import java_cup.runtime.*;
import org.apache.xml.dtm.DTM;
import org.apache.xalan.xslt.DOM;
import org.apache.xml.dtm.Axis;
import org.apache.xalan.xslt.runtime.Operators;
import org.apache.xalan.xslt.compiler.util.ErrorMsg;

parser code {
/**
 * Used by function calls with no args.
 */
static public final Vector EmptyArgs = new Vector(0);

/**
 * Reference to non-existing variable.
 */
static public final VariableRef DummyVarRef = null;

/**
 * Reference to the Parser class.
 */
private Parser _parser;
private XSLTC _xsltc;

/**
 * String representation of the expression being parsed.
 */
private String _expression;

/**
 * Line number where this expression/pattern was declared.
 */
private int _lineNumber = 0;

/**
 * Reference to the symbol table.
 */
public SymbolTable _symbolTable;

public XPathParser(Parser parser) {
    _parser = parser;
    _xsltc = parser.getXSLTC();
    _symbolTable = parser.getSymbolTable();
}

public int getLineNumber() {
    return _lineNumber;
}

public QName getQNameIgnoreDefaultNs(String name) {
    return _parser.getQNameIgnoreDefaultNs(name);
}

public QName getQName(String namespace, String prefix, String localname) {
    return _parser.getQName(namespace, prefix, localname);
}

public void setMultiDocument(boolean flag) {
    _xsltc.setMultiDocument(flag);
}

public void setCallsNodeset(boolean flag) {
    _xsltc.setCallsNodeset(flag);
}

public void setHasIdCall(boolean flag) {
    _xsltc.setHasIdCall(flag);
}

/**
 * This method is similar to findNodeType(int, Object) except that it
 * creates a StepPattern instead of just returning a node type. It also
 * differs in the way it handles "*[uri]::*" and "*[uri]:@*". The last two
 * patterns are expanded as "*[namespace-uri() = 'uri']::*" and
 * "*[namespace-uri() = 'uri']:@*", respectively. This expansion considerably
 * simplifies the grouping of patterns in the Mode class. For this
 * expansion to be correct, the priority of the pattern/template must be
 * set to -0.25 (when no other predicates are present).
public StepPattern createStepPattern(int axis, Object test, Vector predicates) {
    int nodeType;
    if (test == null) { // ".*
        nodeType = (axis == Axis.ATTRIBUTE) ? NodeTest.ATTRIBUTE : 
            (axis == Axis.NAMESPACE) ? -1 : NodeTest.ELEMENT;
    }
    else if (test instanceof Integer) {
        nodeType = ((Integer) test).intValue();
    } else {
        QName name = (QName)test;
        boolean setPriority = false;
        if (axis == Axis.NAMESPACE) {
            nodeType = (name.toString().equals("*")) ? -1 
                : _xslt.registerNamespacePrefix(name);;
        } else {
            final String uri = name.getNamespace();
            final String local = name.getLocalPart();
            final QName namespace_uri = 
                _parser.getQNameIgnoreDefaultNs("namespace-uri");
            // Expand {uri}:* to *[namespace-uri() = 'uri'] - same for @*
            if (uri != null && (local.equals("*") || local.equals("@*"))) {
                if (predicates == null) {
                    predicates = new Vector(2);
                    // Priority is set by hand if no other predicates exist
                    setPriority = (predicates.size() == 0);
                }
                predicates.add(
                    new Predicate(
                        new EqualityExpr(Operators.EQ,
                            new NamespaceUriCall(namespace_uri),
                            new LiteralExpr(uri))));
            }
            if (local.equals("*")) {
                nodeType = (axis == Axis.ATTRIBUTE) ? NodeTest.ATTRIBUTE 
                    : NodeTest.ELEMENT;
            }
        }
    }
else if (local.equals("@*")) {
    nodeType = NodeTest.ATTRIBUTE;
}
else {
    nodeType = (axis == Axis.ATTRIBUTE) ? _xslt.registerAttribute(name) : _xslt.registerElement(name);
}

final StepPattern result = new StepPattern(axis, nodeType, predicates);

// Set priority for case prefix:* and prefix:@* (no predicates)
if (setPriority) {
    result.setPriority(-0.25);
}

return result;

public int findNodeType(int axis, Object test) {
if (test == null) { // *
    return (axis == Axis.ATTRIBUTE) ? NodeTest.ATTRIBUTE:
        (axis == Axis.NAMESPACE) ? -1 : NodeTest.ELEMENT;
}
else if (test instanceof Integer) {
    return ((Integer)test).intValue();
}
else {
    QName name = (QName)test;

    if (axis == Axis.NAMESPACE) {
        return (name.toString().equals("*") ? -1 : _xslt.registerNamespacePrefix(name));
    }
    if (name.getNamespace() == null) {
        final String local = name.getLocalPart();

        if (local.equals("*") {
            return (axis == Axis.ATTRIBUTE) ? NodeTest.ATTRIBUTE : NodeTest.ELEMENT;
        }
    }
    return NodeTest.ATTRIBUTE;
}
return (axis == Axis.ATTRIBUTE) ? _xslt.registerAttribute(name) : _xslt.registerElement(name);
}

/**
 * Parse the expression passed to the current scanner. If this
 * expression contains references to local variables and it will be
 * compiled in an external module (not in the main class) request
 * the current template to create a new variable stack frame.
 *
 * @param lineNumber Line where the current expression is defined.
 * @param external   Set to <tt>true</tt> if this expression is
 *                   compiled in a separate module.
 *
 */
public Symbol parse(String expression, int lineNumber) throws Exception {
  try {
    _expression = expression;
    _lineNumber = lineNumber;
    return super.parse();
  }
  catch (IllegalCharException e) {
    ErrorMsg err = new ErrorMsg(ErrorMsg.ILLEGAL_CHAR_ERR,
                                 lineNumber, e.getMessage());
    _parser.reportError(Constants.FATAL, err);
  }
  return null;
}

/**
 * Lookup a variable or parameter in the symbol table given its name.
 *
 * @param name Name of the symbol being looked up.
 *
 */
final SyntaxTreeNode lookupName(QName name) {
  // Is it a local var or param ?
  final SyntaxTreeNode result = _parser.lookupVariable(name);
  if (result != null)
    return(result);
  else
    return(_symbolTable.lookupName(name));
}

public final void addError(ErrorMsg error) {
  _parser.reportError(Constants.ERROR, error);
}
public void report_error(String message, Object info) {
    final ErrorMsg err = new ErrorMsg(ErrorMsg.SYNTAX_ERR, _lineNumber,
                                       _expression);
    _parser.reportError(Constants.FATAL, err);
}

public void report_fatal_error(String message, Object info) {
    // empty
}

public RelativeLocationPath insertStep(Step step, RelativeLocationPath rlp) {
    if (rlp instanceof Step) {
        return new ParentLocationPath(step, (Step) rlp);
    }
    else if (rlp instanceof ParentLocationPath) {
        final ParentLocationPath plp = (ParentLocationPath) rlp;
        final RelativeLocationPath newrlp = insertStep(step, plp.getPath());
        return new ParentLocationPath(newrlp, plp.getStep());
    }
    else {
        addError(new ErrorMsg(ErrorMsg.INTERNAL_ERR, "XPathParser.insertStep"));
        return rlp;
    }
}

/**
 * Returns true if the axis applies to elements only. The axes
 * child, attribute, namespace, descendant result in non-empty
 * nodesets only if the context node is of type element.
 */
public boolean isElementAxis(int axis) {
    return (axis == Axis.CHILD || axis == Axis.ATTRIBUTE ||
            axis == Axis.NAMESPACE || axis == Axis.DESCENDANT);
}

terminal SLASH, DOT, LBRACK, RBRACK, VBAR, LPAREN, RPAREN, STAR, COMMA;
terminal DOLLAR, ATSIGN;
terminal DDOT, DCOLON, DSLASH;
terminal EQ, NE;
terminal LT, GT, LE, GE;
terminal PLUS, MINUS, DIV, MOD, MULT;
terminal String Literal;
terminal String QNAME;
terminal ID, KEY, TEXT, NODE, OR, AND, COMMENT, PI, PIPARAM, PRECEDINGSIBLING;
terminal SELF, PARENT, CHILD, ATTRIBUTE, ANCESTER, ANCESTORORSELF, DESCENDANT;
terminal DESCendantOrSelf, FOLLOWING, FOLLOWINGSibling, Namespace, Preceding;
terminal Double REAL;
terminal Long INT;
terminal Pattern, Expression;

non terminal SyntaxTreeNode TopLevel;

non terminal Expression Expr, Argument, LocationPath;
non terminal Expression Predicate, FilterExpr, Step;
non terminal Expression OrExpr, AndExpr, EqualityExpr;
non terminal Expression RelationalExpr, AdditiveExpr;
non terminal Expression MultiplicativeExpr, UnaryExpr;
non terminal Expression VariableReference, FunctionCall;
non terminal Expression PrimaryExpr, UnionExpr, PathExpr, AbbreviatedStep;
non terminal Expression RelativeLocationPath, AbbreviatedRelativeLocationPath;
non terminal Expression AbsoluteLocationPath, AbbreviatedAbsoluteLocationPath;

non terminal Object NodeTest, NameTest;

non terminal IdKeyPattern IdKeyPattern;
non terminal Pattern Pattern;
non terminal Pattern LocationPathPattern;
non terminal StepPattern ProcessingInstructionPattern;
non terminal RelativePathPattern RelativePathPattern;
non terminal StepPattern StepPattern;
non terminal Object NodeTestPattern, NameTestPattern;

non terminal Vector Predicates, NonemptyArgumentList;
non terminal QName QName, FunctionName, VariableName;
non terminal Integer AxisName, AxisSpecifier;
non terminal Integer ChildOrAttributeAxisSpecifier;

precedence left VBAR;
precedence left OR;
precedence left AND;
precedence nonassoc EQ, NE;
precedence left LT, GT, LE, GE;

precedence left PLUS, MINUS;
precedence left DIV, MOD, MULT;
precedence left DOLLAR;
precedence left ATSIGN;
precedence right DCOlon;

start with TopLevel;

TopLevel ::= Pattern:pattern
            { : RESULT = pattern; : }
| EXPRESSION Expr:expr |
| : RESULT = expr; |

/* --------------------------- Patterns ----------------------------------- */

Pattern ::= LocationPathPattern:lpp |
{ : RESULT = lpp; : }

| LocationPathPattern:lpp VBAR Pattern:p |
{ : RESULT = new AlternativePattern(lpp, p); : }

LocationPathPattern ::= SLASH |
{ : RESULT = new AbsolutePathPattern(null); : }

| SLASH RelativePathPattern:rpp |
{ : RESULT = new AbsolutePathPattern(rpp); : }

| IdKeyPattern:ikp |
{ : RESULT = ikp; : }

| IdKeyPattern:ikp SLASH RelativePathPattern:rpp |
{ : RESULT = new ParentPattern(ikp, rpp); : }

| IdKeyPattern:ikp DSLASH RelativePathPattern:rpp |
{ : RESULT = new AncestorPattern(ikp, rpp); : }

| DSLASH RelativePathPattern:rpp |
{ : RESULT = new AncestorPattern(rpp); : }

| RelativePathPattern:rpp |
{ : RESULT = rpp; : }

IdKeyPattern ::= ID LPAREN Literal:l RPAREN |
{ : RESULT = new IdPattern(l); 
  parser.setHasIdCall(true); 
  : }

| KEY LPAREN Literal:l1 COMMA Literal:l2 RPAREN |
{ : RESULT = new KeyPattern(l1, l2); : }

ProcessingInstructionPattern ::= PIPARAM LPAREN Literal:l RPAREN |
{ : RESULT = new ProcessingInstructionPattern(l); : }

RelativePathPattern ::= StepPattern:sp |
{ : RESULT = sp; : }

| StepPattern:sp SLASH RelativePathPattern:rpp
\[ \text{RESULT} = \text{new ParentPattern}(\text{sp}, \text{rpp}); \]

\[ \text{StepPattern} : \text{sp DSLASH RelativePathPattern} : \text{rpp} \]
\[ \{ : \text{RESULT} = \text{new AncestorPattern}(\text{sp}, \text{rpp}); \} ; \]

\textbf{StepPattern ::= NodeTestPattern} : \text{nt}  
\[ : \]
\textbf{RESULT} = \text{parser.createStepPattern(Axis.CHILD, nt, null)};
\[ : \]

\[ \{ : \text{RESULT} = \text{parser.createStepPattern(Axis.CHILD, nt, pp)}; \} ; \]

\[ \{ : \text{RESULT} = \text{parser.createStepPattern(Axis.CHILD, nt, pp)}; \} ; \]

\[ \{ : \text{RESULT} = \text{pip}; \} ; \]

\[ \{ : \text{RESULT} = (\text{ProcessingInstructionPattern})\text{pip.setPredicates(pp)}; \} ; \]

\[ \{ : \text{RESULT} = \text{parser.createStepPattern(axis.intValue(), nt, null)}; \} ; \]

\[ \{ : \text{RESULT} = \text{parser.createStepPattern(axis.intValue(), nt, pp)}; \} ; \]

\[ \{ : \text{RESULT} = \text{pip}; // \text{TODO: report error if axis is attribute} \} ; \]

\[ \{ : \text{RESULT} = (\text{ProcessingInstructionPattern})\text{pip.setPredicates(pp)}; \} ; \]

\textbf{NodeTestPattern ::= NameTestPattern} : \text{nt}  
\[ : \]

\textbf{NODE}
{ : RESULT = new Integer(NodeTest.ANODE); : }

| TEXT
{ : RESULT = new Integer(NodeTest.TEXT); : }

| COMMENT
{ : RESULT = new Integer(NodeTest.COMMENT); : }

| PI
{ : RESULT = new Integer(NodeTest.PI); : }

NameTestPattern ::= STAR
{ : RESULT = null; : }

| QName:qn
{ : RESULT = qn; : }

ChildOrAttributeAxisSpecifier ::= ATSIGN
{ : RESULT = new Integer(Axis.ATTRIBUTE); : }

| CHILD DCOLON
{ : RESULT = new Integer(Axis.CHILD); : }

| ATTRIBUTE DCOLON
{ : RESULT = new Integer(Axis.ATTRIBUTE); : }

Predicates ::= Predicate:p
{ :
Vector temp = new Vector();
temp.addElement(p);
RESULT = temp;
: }

| Predicate:p Predicates:pp
{ : pp.insertElementAt(p, 0); RESULT = pp; : }

Predicate ::= LBRACK Expr:e RBRACK
{ :
RESULT = new Predicate(e);
: }

/* --------------------------- Expressions --------------------------------- */

Expr ::= OrExpr:ex
{ : RESULT = ex; : }

OrExpr ::= AndExpr:ae
{ : RESULT = ae; : }
<table>
<thead>
<tr>
<th>OrExpr:oe OR AndExpr:ae</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new LogicalExpr(LogicalExpr.OR, oe, ae);</td>
</tr>
</tbody>
</table>

AndExpr ::= EqualityExpr:e |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = e;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AndExpr:ae AND EqualityExpr:ee</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new LogicalExpr(LogicalExpr.AND, ae, ee);</td>
</tr>
</tbody>
</table>

EqualityExpr ::= RelationalExpr:re |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = re;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EqualityExpr:ee EQ RelationalExpr:re</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new EqualityExpr(Operators.EQ, ee, re);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EqualityExpr:ee NE RelationalExpr:re</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new EqualityExpr(Operators.NE, ee, re);</td>
</tr>
</tbody>
</table>

RelationalExpr ::= AdditiveExpr:ae |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = ae;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RelationalExpr:re LT AdditiveExpr:ae</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new RelationalExpr(Operators.LT, re, ae);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RelationalExpr:re GT AdditiveExpr:ae</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new RelationalExpr(Operators.GT, re, ae);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RelationalExpr:re LE AdditiveExpr:ae</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new RelationalExpr(Operators.LE, re, ae);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RelationalExpr:re GE AdditiveExpr:ae</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new RelationalExpr(Operators.GE, re, ae);</td>
</tr>
</tbody>
</table>

AdditiveExpr ::= MultiplicativeExpr:me |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = me;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AdditiveExpr:ae PLUS MultiplicativeExpr:me</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new BinOpExpr(BinOpExpr.PLUS, ae, me);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AdditiveExpr:ae MINUS MultiplicativeExpr:me</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new BinOpExpr(BinOpExpr.MINUS, ae, me);</td>
</tr>
</tbody>
</table>

MultiplicativeExpr ::= UnaryExpr:ue |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = ue;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MultiplicativeExpr:me MULT UnaryExpr:ue</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT = new BinOpExpr(BinOpExpr.TIMES, me, ue);</td>
</tr>
</tbody>
</table>
| MultiplicativeExpr:me DIV UnaryExpr:ue
{| : RESULT = new BinOpExpr(BinOpExpr.DIV, me, ue); : |

| MultiplicativeExpr:me MOD UnaryExpr:ue
{| : RESULT = new BinOpExpr(BinOpExpr.MOD, me, ue); : |}

UnaryExpr ::= UnionExpr:ue
{| : RESULT = ue; : |

| MINUS UnaryExpr:ue
{| : RESULT = new UnaryOpExpr(ue); : |

UnionExpr ::= PathExpr:pe
{| : RESULT = pe; : |

| PathExpr:pe VBAR UnionExpr:rest
{| : RESULT = new UnionPathExpr(pe, rest); : |

PathExpr ::= LocationPath:lp
{| : RESULT = lp; : |

| FilterExpr:fexp
{| : RESULT = fexp; : |

| FilterExpr:fexp SLASH RelativeLocationPath:rlp
{| : RESULT = new FilterParentPath(fexp, rlp); : |

| FilterExpr:fexp DSLASH RelativeLocationPath:rlp
{| :
   // Expand '//' into '/descendant-or-self::node()/' or
   // into /descendant-or-self::*/
   //
   int nodeType = DOM.NO_TYPE;
   if (rlp instanceof Step &&
       parser.isElementAxis(((Step) rlp).getAxis()))
   {
      nodeType = DTM.ELEMENT_NODE;
   }
   final Step step = new Step(Axis.DESCENDANTORSELF, nodeType, null);
   FilterParentPath fpp = new FilterParentPath(fexp, step);
   fpp = new FilterParentPath(fpp, rlp);
   if (fexp instanceof KeyCall == false) {
      fpp.setDescendantAxis();
   }
   RESULT = fpp;
| ; |
LocationPath ::= RelativeLocationPath:rlp
   { : RESULT = rlp; : }

| AbsoluteLocationPath:alp
   { : RESULT = alp; : }

RelativeLocationPath ::= Step:step
   { : RESULT = step; : }

| RelativeLocationPath:rlp SLASH Step:step
   { :
    if (rlp instanceof Step && ((Step) rlp).isAbbreviatedDot()) {
        RESULT = step;          // Remove './' from the middle
    }
    else if (((Step) step).isAbbreviatedDot()) {
        RESULT = rlp;           // Remove '/.' from the end
    } else {
        RESULT =
        new ParentLocationPath((RelativeLocationPath) rlp, step);
    }
   }

| AbbreviatedRelativeLocationPath:arlp
   { : RESULT = arlp; : }

AbsoluteLocationPath ::= SLASH
   { : RESULT = new AbsoluteLocationPath(); : }

| SLASH RelativeLocationPath:rlp
   { : RESULT = new AbsoluteLocationPath(rlp); : }

| AbbreviatedAbsoluteLocationPath:aalp
   { : RESULT = aalp; : }

AbbreviatedRelativeLocationPath ::= RelativeLocationPath:rlp DSLASH Step:step
   { :
    final Step right  = (Step)step;
    final int axis    = right.getAxis();
    final int type    = right.getNodeType();
    final Vector predicates = right.getPredicates();
    if ((axis == Axis.CHILD) && (type != NodeTest.ATTRIBUTE)) {
        // Compress '/child:E' into 'descendant::E' - if possible
        if (predicates == null) {
            right.setAxis(Axis.DESCENDANT);
            if (rlp instanceof Step && ((Step)rlp).isAbbreviatedDot()) {
RESULT = right;
}
else {
    // Expand 'rlp//E' into 'rlp/descendant::E'
    RelativeLocationPath left = (RelativeLocationPath)rlp;
    RESULT = new ParentLocationPath(left, right);
}
}
else {
    // Expand './/step' -> 'descendant-or-self::*/step'
    if (rlp instanceof Step && ((Step)rlp).isAbbreviatedDot()) {
        Step left = new Step(Axis.DESCENDANTORSELF,
                DTM.ELEMENT_NODE, null);
        RESULT = new ParentLocationPath(left, right);
    }
    else {
        // Expand 'rlp//step' -> 'rlp/descendant-or-self::*/step'
        RelativeLocationPath left = (RelativeLocationPath)rlp;
        Step mid = new Step(Axis.DESCENDANTORSELF,
                DTM.ELEMENT_NODE, null);
        ParentLocationPath ppl = new ParentLocationPath(mid, right);
        RESULT = new ParentLocationPath(left, ppl);
    }
}
}
else if ((axis == Axis.ATTRIBUTE) || (type == NodeTest.ATTRIBUTE)) {
    // Expand 'rlp//step' -> 'rlp/descendant-or-self::*/step'
    RelativeLocationPath left = (RelativeLocationPath)rlp;
    Step middle = new Step(Axis.DESCENDANTORSELF,
                DOM.NO_TYPE, null);
    ParentLocationPath ppl = new ParentLocationPath(middle, right);
    RESULT = new ParentLocationPath(left, ppl);
}
else {
    // Expand 'rlp//step' -> 'rlp/descendant-or-self::node()/step'
    RelativeLocationPath left = (RelativeLocationPath)rlp;
    Step middle = new Step(Axis.DESCENDANTORSELF,
                DOM.NO_TYPE, null);
    ParentLocationPath ppl = new ParentLocationPath(middle, right);
    RESULT = new ParentLocationPath(left, ppl);
}
};

AbbreviatedAbsoluteLocationPath ::= DSLASH RelativeLocationPath:rlp
    {:
        //
        // Expand '//' into '/descendant-or-self::node()' or
int nodeType = DOM.NO_TYPE;
if (rlp instanceof Step &&
    parser.isElementAxis(((Step) rlp).getAxis()))
{
    nodeType = DTM.ELEMENT_NODE;
}
final Step step = new Step(Axis.DESCENDANTORSELF, nodeType, null);
RESULT = new AbsoluteLocationPath(parser.insertStep(step,
    (RelativeLocationPath) rlp));
};

Step ::= NodeTest:ntest
  |
      :
      if (ntest instanceof Step) {
          RESULT = (Step)ntest;
      }
      else {
          RESULT = new Step(Axis.CHILD,
              parser.findNodeType(Axis.CHILD, ntest),
              null);
      }
  
  | NodeTest:ntest Predicates:pp
    |
    : if (ntest instanceof Step) {
        Step step = (Step)ntest;
        step.addPredicates(pp);
        RESULT = (Step)ntest;
    }
    else {
        RESULT = new Step(Axis.CHILD,
            parser.findNodeType(Axis.CHILD, ntest),
            pp);
    }
  
  | AxisSpecifier:axis NodeTest:ntest Predicates:pp
    |
    : RESULT = new Step(axis.intValue(),
        parser.findNodeType(axis.intValue(), ntest),
        pp);
    }

  | AxisSpecifier:axis NodeTest:ntest
    |
    : RESULT = new Step(axis.intValue(),
        parser.findNodeType(axis.intValue(), ntest),
        null);
AbbreviatedStep::abbrev
{: RESULT = abbrev; :};

AxisSpecifier ::= AxisName:an DCOLON
{: RESULT = an; :}

| ATSIGN
{: RESULT = new Integer(Axis.ATTRIBUTE); :};

AxisName ::= ANCESTOR
{: RESULT = new Integer(Axis.ANCESTOR); :}

| ANCESTORORSELF
{: RESULT = new Integer(Axis.ANCESTORORSELF); :}

| ATTRIBUTE
{: RESULT = new Integer(Axis.ATTRIBUTE); :}

| CHILD
{: RESULT = new Integer(Axis.CHILD); :}

| DESCENDANT
{: RESULT = new Integer(Axis.DESCENDANT); :}

| DESCENDANTORSELF
{: RESULT = new Integer(Axis.DESCENDANTORSELF); :}

| FOLLOWING
{: RESULT = new Integer(Axis.FOLLOWING); :}

| FOLLOWINGSIBLING
{: RESULT = new Integer(Axis.FOLLOWINGSIBLING); :}

| NAMESPACE
{: RESULT = new Integer(Axis.NAMESPACE); :}

| PARENT
{: RESULT = new Integer(Axis.PARENT); :}

| PRECEDING
{: RESULT = new Integer(Axis.PRECEDING); :}

| PRECEDINGSIBLING
{: RESULT = new Integer(Axis.PRECEDINGSIBLING); :}

| SELF
AbbreviatedStep ::= DOT
{ : RESULT = new Step(Axis.SELF, NodeTest.ANODE, null); : }

| DDOT
{ : RESULT = new Step(Axis.PARENT, NodeTest.ANODE, null); : };

FilterExpr ::= PrimaryExpr:primary
{ : RESULT = primary; : }

| PrimaryExpr:primary Predicates:pp
{ : RESULT = new FilterExpr(primary, pp); : };

PrimaryExpr ::= VariableReference:vr
{ : RESULT = vr; : }

| LPAREN Expr:ex RPAREN
{ : RESULT = ex; : }

| Literal:string
{ :
/*
* If the string appears to have the syntax of a QName, store
* namespace info in the literal expression. This is used for
* element-available and function-available functions, among
* others. Also, the default namespace must be ignored.
*/
String namespace = null;
final int index = string.lastIndexOf(':');

if (index > 0) {
    final String prefix = string.substring(0, index);
    namespace = parser._symbolTable.lookupNamespace(prefix);
}
RESULT = (namespace == null) ? new LiteralExpr(string)
    : new LiteralExpr(string, namespace);
}

| INT:num
{ :
    long value = num.longValue();
    if (value < Integer.MIN_VALUE || value > Integer.MAX_VALUE) {
        RESULT = new RealExpr(value);
    }
    else {
        if (num.doubleValue() == -0)
            RESULT = new RealExpr(num.doubleValue());
    }
else if (num.intValue() == 0)
    RESULT = new IntExpr(num.intValue());
else if (num.doubleValue() == 0.0)
    RESULT = new RealExpr(num.doubleValue());
else
    RESULT = new IntExpr(num.intValue());
}
}

| REAL:.num |
| : RESULT = new RealExpr(num.doubleValue()); |

| FunctionCall:fc |
| : RESULT = fc; |

VariableReference ::= DOLLAR VariableName:varName
 |
// An empty qname prefix for a variable or parameter reference
// should map to the null namespace and not the default URI.
SyntaxTreeNode node = parser.lookupName(varName);

if (node != null) {
    if (node instanceof Variable) {
        RESULT = new VariableRef((Variable)node);
    }
    else if (node instanceof Param) {
        RESULT = new ParameterRef((Param)node);
    }
    else {
        RESULT = new UnresolvedRef(varName);
    }
}

if (node == null) {
    RESULT = new UnresolvedRef(varName);
}

| FunctionCall ::= FunctionName:fname LPAREN RPAREN |

if (node != null) {
    if (node instanceof Variable) {
        RESULT = new VariableRef((Variable)node);
    }
    else if (node instanceof Param) {
        RESULT = new ParameterRef((Param)node);
    }
    else {
        RESULT = new UnresolvedRef(varName);
    }
}

if (node == null) {
    RESULT = new UnresolvedRef(varName);
}

| if (parser.getQNameIgnoreDefaultNs("current").equals(fname)) |
| : RESULT = new CurrentCall(fname); |

else if (parser.getQNameIgnoreDefaultNs("number").equals(fname)) {
    RESULT = new NumberCall(fname, parser.EmptyArgs);
}
else if (parser.getQNameIgnoreDefaultNs("string").equals(fname)) {

RESULT = new StringCall(fname, parser.EmptyArgs);
}

else if (parser.getQNameIgnoreDefaultNs("concat").equals(fname)) {
    RESULT = new ConcatCall(fname, parser.EmptyArgs);
}

else if (parser.getQNameIgnoreDefaultNs("true").equals(fname)) {
    RESULT = new BooleanExpr(true);
}

else if (parser.getQNameIgnoreDefaultNs("false").equals(fname)) {
    RESULT = new BooleanExpr(false);
}

else if (parser.getQNameIgnoreDefaultNs("name").equals(fname)) {
    RESULT = new NameCall(fname);
}

else if (parser.getQNameIgnoreDefaultNs("generate-id").equals(fname)) {
    RESULT = new GenerateIdCall(fname, parser.EmptyArgs);
}

else if (parser.getQNameIgnoreDefaultNs("string-length").equals(fname)) {
    RESULT = new StringLengthCall(fname, parser.EmptyArgs);
}

else if (parser.getQNameIgnoreDefaultNs("position").equals(fname)) {
    RESULT = new PositionCall(fname);
}

else if (parser.getQNameIgnoreDefaultNs("last").equals(fname)) {
    RESULT = new LastCall(fname);
}

else if (parser.getQNameIgnoreDefaultNs("local-name").equals(fname)) {
    RESULT = new LocalNameCall(fname);
}

else if (parser.getQNameIgnoreDefaultNs("namespace-uri").equals(fname)) {
    RESULT = new NamespaceUriCall(fname);
}

else {
    RESULT = new FunctionCall(fname, parser.EmptyArgs);
}

| FunctionName:fname LPAREN NonemptyArgumentList:argl RPAREN |
| {:
    if (parser.getQNameIgnoreDefaultNs("concat").equals(fname)) {
        RESULT = new ConcatCall(fname, argl);
    }
    else if (parser.getQNameIgnoreDefaultNs("number").equals(fname)) {
        RESULT = new NumberCall(fname, argl);
    }
    else if (parser.getQNameIgnoreDefaultNs("document").equals(fname)) {
        parser.setMultiDocument(true);
        RESULT = new DocumentCall(fname, argl);
    }
}
else if (parser.getQNameIgnoreDefaultNs("string").equals(fname)) {
    RESULT = new StringCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("boolean").equals(fname)) {
    RESULT = new BooleanCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("name").equals(fname)) {
    RESULT = new NameCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("generate-id").equals(fname)) {
    RESULT = new GenerateIdCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("not").equals(fname)) {
    RESULT = new NotCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("format-number").equals(fname)) {
    RESULT = new FormatNumberCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("unparsed-entity-uri").equals(fname)) {
    RESULT = new UnparsedEntityUriCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("key").equals(fname)) {
    RESULT = new KeyCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("id").equals(fname)) {
    RESULT = new KeyCall(fname, argl);
    parser.setHasIdCall(true);
}
else if (parser.getQNameIgnoreDefaultNs("ceiling").equals(fname)) {
    RESULT = new CeilingCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("round").equals(fname)) {
    RESULT = new RoundCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("floor").equals(fname)) {
    RESULT = new FloorCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("contains").equals(fname)) {
    RESULT = new ContainsCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("string-length").equals(fname)) {
    RESULT = new StringLengthCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("starts-with").equals(fname)) {
    RESULT = new StartsWithCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("function-available").equals(fname)) {
RESULT = new FunctionAvailableCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("element-available").equals(fname)) {
    RESULT = new ElementAvailableCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("local-name").equals(fname)) {
    RESULT = new LocalNameCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("lang").equals(fname)) {
    RESULT = new LangCall(fname, argl);
}
else if (parser.getQNameIgnoreDefaultNs("namespace-uri").equals(fname)) {
    RESULT = new NamespaceUriCall(fname, argl);
}
else if (parser.getQName(TRANSLET_URI, "xslt", "cast").equals(fname)) {
    RESULT = new CastCall(fname, argl);
}
// Special case for extension function nodeset()
else if (fname.getLocalPart().equals("nodeset") || fname.getLocalPart().equals("node-set")) {
    parser.setCallsNodeset(true);  // implies MultiDOM
    RESULT = new FunctionCall(true, argl);
}
else {
    RESULT = new FunctionCall(true, argl);
}
);

NonemptyArgumentList ::= Argument:arg
{|,
    Vector temp = new Vector();
    temp.addElement(arg);
    RESULT = temp;
|}
| Argument:arg COMMA NonemptyArgumentList:argl
{|: argl.insertElementAt(arg, 0); RESULT = argl; :|)

FunctionName ::= QName:fname
{|,
    RESULT = fname;
|};

VariableName ::= QName:vname
{|,
    RESULT = vname;
|};

Argument ::= Expr:ex
NodeTest ::= NameTest:nt
    { : RESULT = nt; : }

| NODE
    { : RESULT = new Integer(NodeTest.ANODE); : }

| TEXT
    { : RESULT = new Integer(NodeTest.TEXT); : }

| COMMENT
    { : RESULT = new Integer(NodeTest.COMMENT); : }

| PIPARAM LPAREN Literal:l RPAREN
    { :
        QName name = parser.getQNameIgnoreDefaultNs("name");
        Expression exp = new EqualityExpr(Operators.EQ,
            new NameCall(name),
            new LiteralExpr(l));
        Vector predicates = new Vector();
        predicates.addElement(new Predicate(exp));
        RESULT = new Step(Axis.CHILD, NodeTest.PI, predicates);
    }

| PI
    { : RESULT = new Integer(NodeTest.PI); : }

NameTest ::= STAR
    { : RESULT = null; : }

| QName:qn
    { : RESULT = qn; : );

QName ::= QNAME:qname
    { : RESULT = parser.getQNameIgnoreDefaultNs(qname); : }

| DIV
    { : RESULT = parser.getQNameIgnoreDefaultNs("div"); : }

| MOD
    { : RESULT = parser.getQNameIgnoreDefaultNs("mod"); : }

| KEY
    { : RESULT = parser.getQNameIgnoreDefaultNs("key"); : }

| ANCESTOR
    { : RESULT = parser.getQNameIgnoreDefaultNs("child"); : }
Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/compiler/xpath.cup
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
  
  Open Source Used In DNAC Platform Antelope 3117
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the "License);
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

// is found, then throw a special exception in order to terminate

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/processor/TransformerFactoryImpl.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002 World Wide Web Consortium,
 * (Massachusetts Institute of Technology, Institut National de
 * Recherche en Informatique et en Automatique, Keio University). All
 * Rights Reserved. This program is distributed under the W3C's Software
 * Intellectual Property License. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even
 * the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See W3C License http://www.w3.org/Consortium/Legal/ for more details.
 */

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/domapi/XPathStylesheetDOM3Exception.java
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
* /
/**
 * Execute the proprietary document-location() function, which returns
 * a node set of documents.
 * @xsl.usage advanced
 */

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-
  jar/org/apache/xpath/functions/FuncDoclocation.java
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 * *
 *    http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

$/Id: xpath.lex 1225752 2011-12-30 04:12:46Z mrglavas $ /

/*
 * @author Jacek Ambroziak
 * @author Santiago Pericas-Geertsen
 * @author Morten Jorgensen
 *
 */

package org.apache.xalan.xsltcompiler;

import java_cup.runtime.Symbol;
```java
void initialize() {
    last = -1;
}

static boolean isWhitespace(int c) {
    return (c == ' ' || c == '	' || c == '' || c == '
' || c == '');
}

/**
* If symbol is not followed by `::` or `('`, then treat it as a
* name instead of an axis or function (Jira-1912).
*/
Symbol disambiguateAxisOrFunction(int ss) throws Exception {
    // Peek in the input buffer without changing the internal state
    int index = yy_buffer_index;
    // Skip whitespace
    while (index < yy_buffer_read && isWhitespace(yy_buffer[index])) {
        index++;
    }
    // If end of buffer, can't disambiguate :
    if (index >= yy_buffer_read) {
        // Can't disambiguate, so return as symbol
        return new Symbol(ss);
    }
    // Return symbol if next token is `::` or `('`
    return (yy_buffer[index] == ':' && yy_buffer[index+1] == ':') ||
        (yy_buffer[index] == '(')
            ? newSymbol(ss) : newSymbol(sym.QNAME, yytext());
}

/**
* If symbol is first token or if it follows any of the operators
* listed in http://www.w3.org/TR/xpath#exprlex then treat as a
* name instead of a keyword (Jira-1912).
*/
Symbol disambiguateOperator(int ss) throws Exception {
```
switch (last) {
    case -1: // first token
    case sym.ATSIGN:
    case sym.DCOLOGIN:
    case sym.LPAREN:
    case sym.LBRACK:
    case sym.COMMA:
    case sym.AND:
    case sym.OR:
    case sym.MOD:
    case sym.DIV:
    case sym.MULT:
    case sym.SLASH:
    case sym.DSLASH:
    case sym.VBAR:
    case sym.PLUS:
    case sym.MINUS:
    case sym.EQ:
    case sym.NE:
    case sym.LT:
    case sym.LE:
    case sym.GT:
    case sym.GE:
        return newSymbol(sym.QNAME, yytext());
    }
    return newSymbol(ss);
}

/**
 * If symbol is first token or if it follows any of the operators
 * listed in http://www.w3.org/TR/xpath#exprlex then treat as a
 * wildcard instead of a multiplication operator
 */
Symbol disambiguateStar() throws Exception {
    switch (last) {
        case -1: // first token
        case sym.ATSIGN:
        case sym.DCOLOGIN:
        case sym.LPAREN:
        case sym.LBRACK:
        case sym.COMMA:
        case sym.AND:
        case sym.OR:
        case sym.MOD:
        case sym.DIV:
        case sym.MULT:
        case sym.SLASH:
        case sym.DSLASH:

case sym.VBAR:
case sym.PLUS:
case sym.MINUS:
case sym.EQ:
case sym.NE:
case sym.LT:
case sym.LE:
case sym.GT:
case sym.GE:
    return newSymbol(sym.STAR);
}
return newSymbol(sym.MULT);
}

Symbol newSymbol(int ss) {
    last = ss;
    return new Symbol(ss);
}

Symbol newSymbol(int ss, String value) {
    last = ss;
    return new Symbol(ss, value);
}

Symbol newSymbol(int ss, Long value) {
    last = ss;
    return new Symbol(ss, value);
}

Symbol newSymbol(int ss, Double value) {
    last = ss;
    return new Symbol(ss, value);
}
%
%
%eofval{
    return newSymbol(sym.EOF);
%eofval}

%yylexthrow{
    Exception
%yylexthrow}

Letter={BaseChar}|{Ideographic}

BaseChar=[\u0041-\u005A\u0061-\u007A\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u00FF\u0100-\u0131\u0134-\u013E\u0141-\u0149\u014A-\u017E\u0180-\u01C3\u01CD-\u01F4-\u01F5\u01FA-\u0217\u02A8-\u02BB-\u02C1-\u0386\u0388-\u038A\u038C\u038E-\u03A1\u03A3-\u03CE\u03D0-]
"ancestor-or-self" { return disambiguateAxisOrFunction(sym.ANCESTORORSELF); }
"descendant" { return disambiguateAxisOrFunction(sym.DESCENDANT); }
"descendant-or-self" { return disambiguateAxisOrFunction(sym.DESCENDANTORSELF); }
"following" { return disambiguateAxisOrFunction(sym.FOLLOWING); }
"following-sibling" { return disambiguateAxisOrFunction(sym.FOLLOWINGSIBLING); }
"namespace" { return disambiguateAxisOrFunction(sym.NAMESPACE); }
"parent" { return disambiguateAxisOrFunction(sym.PARENT); }
"preceding" { return disambiguateAxisOrFunction(sym.PRECEDING); }
"preceding-sibling" { return disambiguateAxisOrFunction(sym.PRECEDINGSIBLING); }
"self" { return disambiguateAxisOrFunction(sym.SELF); }
"[" { return newSymbol(sym.LBRACK); }
 "]" { return newSymbol(sym.RBRACK); }
 "}" { return newSymbol(sym.LPAREN); }
 ")" { return newSymbol(sym.RPAREN); }
"<PATTERN>" { initialize(); return new Symbol(sym.PATTERN); }
"<EXPRESSION>" { initialize(); return new Symbol(sym.EXPRESSION); }
\["[^\"]*" { return newSymbol(sym.Literal,
   yytext().substring(1, yytext().length() - 1)); }
\["[^\"]*\] { return newSymbol(sym.Literal,
   yytext().substring(1, yytext().length() - 1)); }
{Digit}+ { return newSymbol(sym.INT, new Long(yytext())); }
{Digit}+("[Digit]*)? { return newSymbol(sym.REAL, new Double(yytext())); }
"."{Digit}+ { return newSymbol(sym.REAL, new Double(yytext())); }
"." { return newSymbol(sym.DOT); }
({NCName}:)?[{NCName} { return newSymbol(sym.QNAME, yytext()); }
({NCName}:)?"*" { return newSymbol(sym.QNAME, yytext()); }
({NCName}:)?"@*" { return newSymbol(sym.QNAME, yytext()); }
\[\s\n\t\f\] { /* ignore white space. */ }
. { throw new Exception(yytext()); }

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-
  jar/org/apache/xalan/xslt/compiler/xpath.lex
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
public class XSLProcessorVersion
{

    public static void main(String argv[])
    {
        System.out.println(S_VERSION);
    }

    public static final String PRODUCT = "Xalan";

    public static final String LANGUAGE = "Java";

    public static final String S_VERSION = "version number of Xalan release.
    <P>See also: org/apache/xalan/res/XSLTInfo.properties</P>
    @deprecated To be replaced by org.apache.xalan.Version.getVersion()
    @xsl.usage general
*/
public static final int VERSION = @version.VERSION@;

/**
 * Release Number.
 * Release number. This changes when:
 * - a new set of functionality is to be added, eg,
 * - implementation of a new W3C specification.
 * - API or behaviour change.
 * - its designated as a reference release.
 */
public static final int RELEASE = @version.RELEASE@;

/**
 * Maintenance Drop Number.
 * Optional identifier used to designate maintenance
 * drop applied to a specific release and contains
 * fixes for defects reported. It maintains compatibility
 * with the release and contains no API changes.
 * When missing, it designates the final and complete
 * development drop for a release.
 */
public static final int MAINTENANCE = @version.MINOR@;

/**
 * Development Drop Number.
 * Optional identifier designates development drop of
 * a specific release. D01 is the first development drop
 * of a new release.
 * Development drops are works in progress towards a
 * completed, final release. A specific development drop
 * may not completely implement all aspects of a new
 * feature, which may take several development drops to
 * complete. At the point of the final drop for the
 * release, the D suffix will be omitted.
 * Each 'D' drops can contain functional enhancements as
 * well as defect fixes. 'D' drops may not be as stable as
 * the final releases.
 */
public static final int DEVELOPMENT = 0;

/**
 * Version String like <CODE>"<B>Xalan</B> <B>Language</B> v.r[.dd| <B>D</B>nn"]</CODE>.
 * <P>Semantics of the version string are identical to the Xerces project.</P>
/*
public static final String S_VERSION = PRODUCT+" "+LANGUAGE+" "+VERSION+" "+RELEASE+" "+(DEVELOPMENT > 0 ? ("D"+DEVELOPMENT) : (""+MAINTENANCE));
*/

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/processor/XSLProcessorVersion.src
No license file was found, but licenses were detected in source scan.

* Licensed to the Apache Software Foundation (ASF) under one
  or more contributor license agreements. See the NOTICE file
  distributed with this work for additional information
  regarding copyright ownership. The ASF licenses this file
  to you under the Apache License, Version 2.0 (the "License);
  you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
  
  http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
  limitations under the License.

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/trace/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/functions/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/transformer/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/client/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/lib/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/domapi/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/extensions/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/operations/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/compiler/package.html

Open Source Used In DNAC Platform Antelope 3128
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/patterns/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/lib/sql/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/res/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/objects/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/axes/package.html
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/res/package.html

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

// Proprietary, built in functions
/*** current function string (Proprietary). */

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/compiler/Keywords.java

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/transformer/QueuedEvents.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/XslAttribute.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/dtm/ref/NodeLocator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/ToStream.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/CurrentCall.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/trax/TransformerHandlerImpl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/When.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/SecuritySupport.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/NodeSetType.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/When.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/CurrentCall.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/When.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Util.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/When.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Util.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/When.java
Open Source Used In DNAC Platform Antelope 3131
Open Source Used in DNAC Platform: Antelope
jar/org/apache/xml/utils/res/XResources_he.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xpath/functions/FuncRound.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/compiler/utils/MethodGenerator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/compiler/IntExpr.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/compiler/Fallback.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/runtime/ErrorMessages_ja.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/utils/DOMOrder.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/compiler/CastExpr.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/compiler/LangCall.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/extensions/ExtensionsTable.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xpath/operations/Minus.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/runtime/ErrorMessages_pl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/utils/res/CharArrayWrapper.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/compiler/AVTPartXPath.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xslt/TransletException.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/utils/ObjectFactory.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/templates/WhiteSpaceInfo.java
Open Source Used In DNAC Platform Antelope 3134

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/trace/EndSelectionEvent.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/functions/FuncLang.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/trax/DOM2SAX.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/res/XResources_ka.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/IntVector.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/processor/ProcessorGlobalVariableDecl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/res/XMLErrorResources_tr.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/LiteralElement.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Step.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/ToXMLStream.java
jar/org/apache/xalan/xsltc/trax/SecuritySupport.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/FilterParentPath.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/Sort.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/CeilingCall.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/functions/Function3Args.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/utils/URI.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xpath/XPath.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xsltc/compiler/util/RealType.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/templates/DecimalFormatProperties.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/xsltc/compiler/AttributeValue.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/transformer/NodeSortKey.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/res/XMLErrorResources_ko.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/res/XMLErrorResources_zh_TW.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/res/XMLErrorResources_sv.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/utils/res/XResourceBundleBase.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xml/utils/res/LongArrayWrapper.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources.jar/org/apache/xalan/templates/ElemPI.java

Open Source Used In DNAC Platform Antelope 3135
jar/org/apache/xalan/transformer/ClonerToResultTree.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/templates/AVTPart.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/cmdline/getopt/GetOptException.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/SecuritySupport.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/templates/ElemIf.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/ParameterRef.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/XSLTBlacklistGenerator.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/XSLTBlacklist.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/XSLTBlacklistFactory.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/XSLTBlacklistManager.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/XSLTBlacklistSet.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/roundcall/DTMDOMException.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/roundcall/RoundCall.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/roundcall/TransformSnapshotImpl.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/roundcall/Util.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/roundcall/ProcessorStylesheetElement.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/roundcall/Template.java
* /opt/cola/permits/1134796662_1613422294.09/xalan-2-7-2-sources-jar/org/apache/xal
Open Source Used In DNAC Platform Antelope 3142

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/ObjectFactory.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/axes/UnionPathIterator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/ElemUnknown.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xpath/objects/XRTreeFragSelectWrapper.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/ReferenceType.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/lib/Redirect.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/XMLReaderManager.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/domapi/XPathExpressionImpl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/utils/SerializerMessages_sl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/transformer/MsgMgr.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/XslElement.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/ReferenceType.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/dom/Filter.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/UnImplNode.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/res/XSLTErrorResources_es.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/ElemOtherwise.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xml/serializer/ExtendedContentHandler.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/runtime/ErrorMessages_de.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/runtime/InternalRuntimeError.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/SourceTreeManager.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/dtm/ObjectFactory.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/utils/XML11Char.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/operations/Div.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/serialize/SerializerFactory.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/processor/ProcessorPreserveSpace.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/XsIElement.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/AttributesImplSerializer.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/UnImplNode.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/XslElement.java
Open Source Used In DNAC Platform Antelope 3144
Open Source Used In DNAC Platform Antelope

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/trace/TraceListenerEx3.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/res/XSLTErrorResources_pt_BR.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/XNumber.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/OutputPropertyUtils.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/TopLevelElement.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Copy.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/dtm/ref/ObjectFactory.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/functions/FunNumber.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/Constants.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/trace/TraceListener.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/dom3/NamespaceSupport.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/dom3/DOMErrorImpl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan/extensions/ExtensionHandlerJava.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/processor/ProcessorStripSpace.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/trace/TraceListenerEx2.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/RealExpr.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/axes/UnionChildIterator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/dtm/DTMException.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/CompareGenerator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/XMLNSDecl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Output.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Closure.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/utils/Utils.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/TemplateList.java

jar/org/apache/xpath/axes/SelfIteratorNoPredicate.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/res/XResources_sv.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/InternalError.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/util/SerializerMessages_ru.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/KeyCall.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/ElemDesc.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/dom/NodeIteratorBase.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/ObjectFactory.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/trace/TraceListenerEx.java

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/ResultTreeType.java
Open Source Used In DNAC Platform Antelope

* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/dom/DOMAdapter.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/dom/AnyNodeCounter.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/dom/getopt/GetOpt.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/dom/FilteredStepIterator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/dom/SingletonIterator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/dom/CountersTable.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/ErrorMessagesFr.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/WithParam.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/CountersTable.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/BooleanExpr.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/MatchGenerator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/compiler/OpMapVector.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/StringVector.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/runtime/StringValueHandler.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/operations/Operation.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/functions/FuncString.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/ElemUse.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/dtm/ref/DTMManagerDefault.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/lib/PipeDocument.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/lib/Extensions.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/TypeCheckError.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/dom/StepIterator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/trax/TemplatesHandlerImpl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/serializer/XSLOutputAttributes.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Text.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/transformer/StackGuard.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/util/IntType.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/CompilerException.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/functions/FuncSubstringAfter.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/dom/CachedNodeListIterator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/TemplateSubPatternAssociation.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xml/utils/res/XResources_zh_CN.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/runtime/ErrorMessages_sk.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/XPathContext.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Include.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/ElemWithParam.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/MethodType.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/WhitespaceStrippingElementMatcher.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/Attribute.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/CopyOf.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/runtime/Attributes.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/axes/MatchPatternIterator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/DecimalFormatting.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/NodeIterator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/util/ErrorMessages_hu.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/transformer/XalanTransformState.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/runtime/ErrorMessages_es.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/serializer/SerializerMessages_pl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/serializer/SerializerMessages_es.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/serializer/SerializerMessages_hu.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/res/XMLErrorResources_ru.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/serializer/SerializerMessages_pl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/serializer/XPathEvaluatorImpl.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/serializer/Expression.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/compiler/serializer/TransformStateSetter.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xalan/templates/ElemExtensionCall.java
jar/org/apache/xalan/processor/ProcessorOutputElem.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/Param.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/AttributeValueTemplate.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/extensions/ObjectFactory.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/res/XPATHErrorResources_ru.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/util/SlotAllocator.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xsltc/trax/DOM2TO.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/templates/Stylesheet.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/xslt/compiler/ConcatCall.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/res/XSLTErroryResources_zh_TW.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xalan/res/XResources_ja_JP_A.java
* /opt/cola/permits/1134796662_1613422294.09/0/xalan-2-7-2-sources-jar/org/apache/xpath/operations/Gt.java
1.913 natural-compare 1.4.0

1.913.1 Available under license:
No license file was found, but licenses were detected in source scan.

[The MIT License](http://lauri.rooden.ee/mit-license.txt)

Found in path(s):
* /opt/cola/permits/1110631610_1613674636.83/0/natural-compare-1.4.0-4-tgz/package/README.md
No license file was found, but licenses were detected in source scan.

/*
 * @version 1.4.0
 * @date 2015-10-26
 * @stability 3 - Stable
 * @author Lauri Rooden (https://github.com/litejs/natural-compare-lite)
 * @license MIT License
 */

Found in path(s):
* /opt/cola/permits/1110631610_1613674636.83/0/natural-compare-1.4.0-4-tgz/package/index.js
1.914 jsreport-express 2.8.1
1.914.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.

1.915 os-locale 1.4.0
1.915.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.916 apib2swagger 1.12.0 1.12.0
1.916.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Keisuke Minami

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.917 hsql-db 1.8.0.10
1.917.1 Available under license :
No license file was found, but licenses were detected in source scan.

(SqlFile processor v. %{1})
Distribution is permitted under the terms of the HSQldb license.
(c) 2004-2007 Blaine Simpson and the HSQldb Development Group.

\q  to Quit.
\?  lists Special Commands.
?:? lists Edit-Buffer/History commands.
*?  lists PL commands.

SPECIAL Commands begin with \\ and execute when you hit ENTER.
EDIT-BUFFER / HISTORY Commands begin with ':' and execute when you hit ENTER.
COMMENTS begin with '/\*\*' and end with the very next '\\*/'.

PROCEDURAL LANGUAGE commands begin with "*" and end when you hit ENTER. All other lines comprise SQL Statements. SQL Statements are terminated by either a blank line (which moves the statement into the edit buffer without executing) or a line ending with ';'. (which executes the statement). SQL Commands may begin with '/PLVARNANE' for a PL variable command alias. After turning on variable expansion with command "*" (or any other PL command), PL variables may be used in any command (other than : commands) like so: *{PLVARNANE}.

1.918 qs 6.10.1 6.10.1
1.918.1 Available under license :  
BSD 3-Clause License

Copyright (c) 2014, Nathan LaFreniere and other [contributors](https://github.com/ljharb/qs/graphs/contributors) All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.919 calcite-elasticsearch 1.21.0-drill-r2

1.919.1 Available under license:

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to you under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Database of contributors to Apache Calcite.
# Pages such as developer.md use this data.
# List must be sorted by first name, last name.
#
- name: Alan Gates
  emeritus: 2018/05/04
  apacheId: gates
  githubId: alanfgates
  org: Hortonworks
  role: PMC
- name: Aman Sinha
  apacheId: amansinha
  githubId: amansinha100
  org: MapR
  role: PMC
- name: Andrei Sereda
  apacheId: sereda
  githubId: asereda-gs
  org:
  role: Committer
- name: Ashutosh Chauhan
  apacheId: hashutosh
  githubId: ashutoshe
  org: Hortonworks
  role: PMC
- name: Chris Baynes
  apacheId: cbaynes
  githubId: chris-baynes
  org: Contiamo
role: Committer
- name: Christian Beikov
  apacheId: cbeikov
githubId: beikov
  org: Blazebit
role: Committer
- name: Chunwei Lei
  apacheId: chunwei
githubId: chunweilei
  org: Alibaba
role: Committer
- name: Danny Chan
  apacheId: danny0405
githubId: danny0405
  org: Alibaba
role: Committer
- name: Edmon Begoli
  apacheId: ebegoli
githubId: ebegoli
  org: Oak Ridge National Laboratory
role: Committer
- name: Francis Chuang
  apacheId: francischuang
githubId: F21
  org: Boostport
role: PMC Chair
- name: Gian Merlino
  apacheId: gian
githubId: gianm
  org: Imply
role: Committer
- name: Haisheng Yuan
  apacheId: hyuan
githubId: hsyuan
  org: Alibaba
role: Committer
- name: Hongze Zhang
  apacheId: hongze
githubId: zhztheplayer
  org: Tencent
role: Committer
- name: James R. Taylor
  apacheId: jamestaylor
githubId: JamesRTaylor
  org: Salesforce
role: PMC
- name: Jacques Nadeau
  apacheId: jacques
- name: Jess Balint
  apacheId: jbalint
  githubId: jbalint
  org: Stardog
  role: Committer

- name: Jess Camacho Rodriguez
  apacheId: jcamacho
  githubId: jcamachor
  org: Hortonworks
  role: PMC

- name: Jinfeng Ni
  apacheId: jni
  githubId: jinfengni
  org: MapR
  role: PMC

- name: John Pullokkaran
  apacheId: jpullokk
  githubId: jpullokkaran
  org: Hortonworks
  role: PMC

- name: Josh Elser
  apacheId: elserj
  githubId: joshelser
  org: Hortonworks
  role: PMC

- name: Julian Hyde
  apacheId: jhyde
  githubId: julianhyde
  org: Looker
  role: PMC
  homepage: http://people.apache.org/~jhyde

- name: Kevin Liew
  apacheId: kliew
  githubId: kliewkliew
  org:
  role: Committer

- name: Kevin Risden
  apacheId: krisden
  githubId: risdenk
  org:
  role: PMC

- name: Laurent Goujon
  apacheId: laurent
  githubId: laurentgo
  org: Dremio
- name: Maryann Xue
  apacheId: maryannxue
  githubId: maryannxue
  org: Intel
  role: PMC

- name: Michael Mior
  apacheId: mmior
  githubId: michaelmior
  org: Rochester Institute of Technology
  role: PMC
  homepage: https://michael.mior.ca/

- name: Milinda Pathirage
  apacheId: milinda
  githubId: milinda
  org: Indiana University
  role: Committer
  homepage: http://milinda.pathirage.org/

- name: MinJi Kim
  apacheId: minji
  githubId: minji-kim
  org: Oracle
  role: Committer

- name: Nick Dimiduk
  apacheId: ndimiduk
  githubId: ndimiduk
  org: Rochester Institute of Technology
  role: PMC
  homepage: http://www.n10k.com

- name: Nishant Bangarwa
  apacheId: nishant
  githubId: nishantmonu51
  org: Hortonworks
  role: Committer

- name: Ruben Quesada Lopez
  apacheId: rubenql
  githubId: rubenada
  org: TIBCO
  role: Committer

- name: Sergey Nuyanzin
  apacheId: snuyanzin
  githubId: snuyanzin
  org: EPAM
  role: Committer

- name: Shuyi Chen
  apacheId: shuyichen
  githubId: suez1224
  org: Uber
- name: Slim Bouguerra
  apacheId: bslim
  githubId: b-slim
  org: Hortonworks
  role: Committer

- name: Stamatis Zampetakis
  apacheId: zabetak
  githubId: zabetak
  org: TIBCO
  role: PMC

- name: Steven Noels
  apacheId: stevenn
  githubId: stevenn
  org: NGData
  role: PMC

- name: Ted Dunning
  apacheId: tdunning
  githubId: tdunning
  org: MapR
  role: PMC
  avatar: https://mapr.com/blog/author/ted-dunning/assets/tdunning-panama.jpg

- name: Vladimir Sitnikov
  apacheId: vladimir-sitnikov
  githubId: vlsi
  org: NetCracker
  role: PMC

- name: Volodymyr Vysotskyi
  apacheId: volodymyr
  githubId: vvyosotskyi
  org:
  role: PMC

- name: Zhen Wang
  apacheId: zhenw
  githubId: zinking
  org:
  role: Committer

- name: Zhiqiang He
  apacheId: zhiqianghe
  githubId: Zhiqiang-He
  org: Huawei
  role: Committer

- name: Zhiwei Peng
  apacheId: zhiwei
  githubId: pengzhiwei2018
  org: Ant Financial Group
  role: Committer

- name: Zoltan Haindrich
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-----------------------------------------------------------------------
APACHE CALCITE SUBCOMPONENTS:

The Apache Calcite project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

-----------------------------------------------------------------------
The MIT License

-----------------------------------------------------------------------

The Apache Calcite project bundles the following files under the MIT License:

- site
  Parts of the web site generated by Jekyll (http://jekyllrb.com/)
  Copyright (c) 2008-2015 Tom Preston-Werner
- site/_sass/_font-awesome.scss
  Font-awesome css files v4.1.0 (http://fortawesome.github.io/Font-Awesome/)
  Copyright (c) 2013 Dave Gandy
- site/_sass/_normalize.scss
  normalize.css v3.0.2 | git.io/normalize
  Copyright (c) Nicolas Gallagher and Jonathan Neal
- site/_sass/_gridism.scss
  Gridism: A simple, responsive, and handy CSS grid by @cobyism
  https://github.com/cobyism/gridism
  Copyright (c) 2013 Coby Chapple

-----------------------------------------------------------------------
Open Source Used In DNAC Platform Antelope 3181
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----------------------------------------------------------------------

The Open Font License
-----------------------------------------------------------------------

The Apache Calcite project bundles the following fonts under the
SIL Open Font License (OFL) - http://scripts.sil.org/OFL/

- site/fonts/fontawesome-webfont.*
  Font-awesome font files v4.0.3 (http://fortawesome.github.io/Font-Awesome/)
Apache Calcite
Copyright 2012-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product is based on source code originally developed
by DynamoBI Corporation, LucidEra Inc., SQLstream Inc. and others
under the auspices of the Eigenbase Foundation
and released as the LucidDB project.

The web site includes files generated by Jekyll.
Apache HBase - Prefix Tree
Copyright 2007-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.921 string-width 2.1.1
1.921.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.922 popperjs 1.15.0 1.15.0
1.922.1 Available under license:

No license file was found, but licenses were detected in source scan.

Code and documentation copyright 2016 **Federico Zivolo**. Code released under the [MIT license](LICENSE.md). Docs released under Creative Commons.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/README.md

No license file was found, but licenses were detected in source scan.

/**!
 * @fileOverview Kickass library to create and place poppers near their reference elements.
 * @version 1.15.0
 * @license
 * Copyright (c) 2016 Federico Zivolo and contributors
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the Software is
 * furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included in all
 * copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/dist/esm/popper-utils.js
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/dist/umd/popper-utils.js
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/dist/umd/popper.js
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/dist/esm/popper.js

No license file was found, but licenses were detected in source scan.

/*
Copyright (C) Federico Zivolo 2019
Distributed under the MIT License (license terms are at http://opensource.org/licenses/MIT).
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/dist/popper.min.js
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/dist/umd/popper.min.js
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/dist/umd/popper-utils.min.js
* /opt/ws_local/PERMITS_SQL/1025666388_1594212234.43/0/popper-js-1-15-0-tgz/package/dist/umd/popper-utils.min.js

1.923 p-is-promise 2.1.0
1.923.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.924 duplexer2 0.1.4
1.924.1 Available under license:
Copyright (c) 2013, Deoxxa Development

All rights reserved.

Open Source Used In DNAC Platform Antelope 3189
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of Deoxxa Development nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY DEOXXA DEVELOPMENT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL DEOXXA DEVELOPMENT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.925 ua-parser-js 0.7.18 0.7.18

1.925.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.926 jsr305 3.0.1
1.927 caseless 0.12.0

1.927.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such
license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.930 esprima 1.2.2 1.2.2
1.930.1 Available under license:
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.931 ajv 5.5.2
1.931.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.932 yallist 2.1.2
1.932.1 Available under license :
   The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.933 base 0.11.2
1.933.1 Available under license :
   The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.934 lodash.uniq 4.5.0
1.934.1 Available under license :
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.935 jnr-constants 0.9.9
1.935.1 Available under license :
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit)
alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and You must cause any modified files to carry prominent notices stating that You changed the files; and You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: HOW TO APPLY THE APACHE LICENSE TO YOUR WORK

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.936 deep-extend 0.6.0

1.936.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013-2018, Viacheslav Lotsmanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.937 api-spec-converter 2.12.0 2.12.2
1.937.1 Available under license:
core-js@2.5.1
MIT
Copyright (c) 2014-2017 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

webpack@3.6.0
MIT
Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

zone.js@0.8.18
MIT
The MIT License

Copyright (c) 2016 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

@angular/core@4.4.5
MIT
MIT

@angular/router@4.4.5
MIT
MIT

@angular/http@4.4.5
MIT
MIT

@angular/forms@4.4.5
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

webpack@3.11.0
MIT
Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

bootstrap-sass@3.4.1
MIT
The MIT License (MIT)

Copyright (c) 2011-2016 Twitter, Inc
Copyright (c) 2011-2016 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

@angular/core@4.4.7
MIT
MIT

@angular/platform-browser@4.4.7
MIT
MIT

@angular/router@4.4.7
MIT
MIT

@angular/forms@4.4.7
MIT
MIT

@angular/http@4.4.7
MIT
MIT

@angular/common@4.4.7
MIT
MIT

file-saver@1.3.8
MIT
The MIT License

Copyright 2016 [Eli Grey][1].

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[1]: http://eligrey.com

@angular/platform-browser-dynamic@4.4.7
MIT
MIT
(MIT License)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.938 camelcase 5.0.0
1.938.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.939 hbase-asyncfs 2.4.2
1.939.1 Available under license:

Apache HBase - Asynchronous FileSystem
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.940 strip-json-comments 2.0.1

1.940.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.941 swagger-jersey2-jaxrs 1.5.0


1.942.1 Available under license :
MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.943 is-windows 1.0.2

1.943.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.944 @types/node 12.12.68
1.944.1 Available under license :
   MIT License

Copyright (c) Microsoft Corporation.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE

1.945 ansistyles 0.1.3
1.945.1 Available under license:

Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.946 jcodings 1.0.55

1.946.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Permission is hereby granted, free of charge, to any person obtaining a copy of
 * this software and associated documentation files (the "Software"), to deal in
 * the Software without restriction, including without limitation the rights to
 * use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
 * of the Software, and to permit persons to whom the Software is furnished to do
 * so, subject to the following conditions:
 * *
 * The above copyright notice and this permission notice shall be included in all
 * copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
*/
Open Source Used In DNAC Platform Antelope 3214

* SOFTWARE.
* /

Found in path(s):
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/util/ArrayReader.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/ISO8859_5Encoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/Universal_newline_Transcoder.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/To_UTF_16BE_Transcoder.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/constants/PosixBracket.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/util/CaseInsensitiveBytesHash.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/AbstractEncoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/ISO8859_10Encoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/Stateless_iso2022jp_to_eucjp_Transcoder.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/Cp50221_decoder_Transcoder.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/To_UTF_32BE_Transcoder.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/Sjis2eucjp_Transcoder.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/exception/JCodingsException.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/Windows_31JEncoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/Windows_1251Encoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/To_UTF_16LE_Transcoder.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/CP949Encoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/GBKEncoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/UTF32BEEncoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/Big5Encoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/AsciiTables.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/specific/Windows_1254Encoding.java
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/jcodings/transcode/specific/Iso2022jp_kddi_decoder_Transcoder.java
No license file was found, but licenses were detected in source scan.

/*
 * Permission is hereby granted, free of charge, to any person obtaining a copy of
 * this software and associated documentation files (the "Software"), to deal in
 * the Software without restriction, including without limitation the rights to
 * use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
 * of the Software, and to permit persons to whom the Software is furnished to do
 * so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included in all
 * copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 */

Found in path(s):
* /opt/cola/permits/1202447017_1631568351.65/0/jcodings-1-0-55-sources-jar/org/jcodings/unicode/FixedWidthUnicodeEncoding.java

**1.947 toggle-selection 1.0.6**

**1.947.1 Available under license:**
MIT License

Copyright (c) 2017 sudodoki <smd.deluzion@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.948 keygrip 1.0.3
1.948.1 Available under license :
The MIT License (MIT)

Copyright (c) 2011-2014 Jed Schmidt <where@jed.is> (http://jedschmidt.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.949 picocli 4.5.2
1.949.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.950 lodash-map 4.6.0
1.950.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own
licenses; we recommend you read them, as their terms may differ from the
terms above.
1.951 util-deprecate 1.0.2 1.0.2

1.951.1 Available under license:
(The MIT License)

Copyright (c) 2014 Nathan Rajlich <nathan@tootallnate.net>

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

1.952 lodash.once 4.1.1 4.1.1

1.952.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

====
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.953 mongodb-driver-core 4.2.3

1.953.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright 2017 Tom Bentley
 * *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 */
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/authentication/SaslPrep.java

Found in path(s):
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/annotations/NotThreadSafe.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/annotations/Immutable.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/annotations/ThreadSafe.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2005 Brian Goetz and Tim Peierls
* Released under the Creative Commons Attribution License
* (http://creativecommons.org/licenses/by/2.5)
* Official home: http://www.jcip.net
* *
* Any republication or derived work distributed in source code form
* must include this copyright and license notice.
*/

/*
* Copyright 2008-present MongoDB, Inc.
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
* */
* Original Work: MIT License, Copyright (c) [2015-2020] all contributors
* https://github.com/marianobarrios/tls-channel
*/

Found in path(s):
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/impl/BufferHolder.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/BufferAllocator.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/TlsChannelFlowControlException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/SniSslContextFactory.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/ClientTlsChannel.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/HeapBufferAllocator.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/WouldBlockException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/ClientTlsChannel.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/NeedsReadException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/Impl/TlsChannelImpl.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/ServerTlsChannel.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/async/AsyncynchronousTlsChannel.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/async/package-info.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/NeedsTaskException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/package-info.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/TlsChannelBuilder.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/impl/ByteBufferSet.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/util/DirectBufferDeallocator.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/TlsChannelCallbackException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/async/ExtendedAsyncynchronousByteChannel.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/async/package-info.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/impl/ByteBufferUtil.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/tlschannel/DirectBufferAllocator.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/*
 * Written by Doug Lea with assistance from members of JCP JSR-166
 * Expert Group and released to the public domain, as explained at
 * http://creativecommons.org/publicdomain/zero/1.0/
 */

/**
 * Returns an array containing all of the elements in this deque, in
 * proper sequence (from first to last element).
 *
 * <p>The returned array will be "safe" in that no references to it are
 * maintained by this deque. (In other words, this method must allocate
 * a new array). The caller is thus free to modify the returned array.
 *
 * <p>This method acts as bridge between array-based and collection-based
 * APIs.
 *
 * @return an array containing all of the elements in this deque
 */
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright 2010 The Guava Authors
 * Copyright 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/annotations/Beta.java
  No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/operation/QueryHelper.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/event/ServerMonitorListenerMulticaster.java
Open Source Used In DNAC Platform Antelope
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/connection/PowerOfTwoBufferPool.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/connection/FieldTrackingBsonWriter.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/MongoNotPrimaryException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/connection/CompositeByteBuf.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/client/model/geojson/GeometryCollection.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/client/model/ValidationOptions.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/client/model/geojson/MultiPolygon.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/client/model/geojson/GeoJsonObjectType.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/client/model/geojson/CoordinateReferenceSystem.java

Open Source Used In DNAC Platform Antelope 3237
jar/com/mongodb/internal/operation/AsyncWriteOperation.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/ScramShaAuthenticator.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/SocketStreamHelper.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/MongoWriteConcernException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/DefaultConnectionFactory.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/DefaultServerMonitorFactory.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/ReplyHeader.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/DnsSrvRecordMonitor.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/DefaultClusterFactory.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/ServerAddress.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/operation/CommandResultArrayCodec.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/capi/MongoCryptHelper.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/CommandProtocolImpl.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/operation/ListCollectionsOperation.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/async/SingleResultCallback.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/MongoCommandException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/BsonWriterHelper.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/MongoSocketReadException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/ByteBufBsonDocument.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/TagSet.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/WriteProtocol.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/bulk/BulkWriteResult.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/operation/DropDatabaseOperation.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/operation/MapReduceStatistics.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/connection/netty/NettyStreamFactoryFactory.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/selector/CompositeServerSelector.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/connection/ClusterId.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/client/model/geojson/LineString.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/SocketStream.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/client/model/CountOptionsHelper.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/binding/SingleConnectionReadBinding.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/client/model/changestream/UpdateDescription.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/operation/BsonDocumentWrapperHelper.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/operation/CreateCollectionOperation.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/client/model/geojson/PolygonCoordinates.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/binding/ReadBinding.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/async/AsyncBatchCursor.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/client/model/ReplaceOptions.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/client/gridfs/codecs/package-info.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/client/model/PushOptions.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/management/NullMBeanServer.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/KillCursorProtocol.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/binding/SingleServerProtocol.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/event/ConnectionCheckedOutEvent.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/AbstractMultiServerCluster.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/WriteConcernResult.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/client/model/CreateCollectionOptions.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/connection/ClusterConnectionMode.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
Open Source Used In DNAC Platform Antelope

- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/validator/MappedFieldNameValidator.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/connection/QueryMessage.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/client/model/AggregationLevel.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/package-info.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/Sorts.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/ReturnDocument.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/vault/DataKeyOptions.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/UpdateOptions.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/geojson/codecs/MultiPointCodec.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/connection/AsynchronousSocketChannelStreamFactory.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/UpdateManyModel.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/package-info.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/Variable.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/UpdateManyModel.java
- /opt/cola/permits/120246273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/client/model/package-info.java
jar/com/mongodb/BSONTimestampCodec.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/authentication/NativeAuthenticationHelper.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/connection/BulkWriteBatchCombiner.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/MongoIncompatibleDriverException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/connection/Server.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/MongoCursorNotFoundException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/BasicDBObjectFactory.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/binding/AsyncReadBinding.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/ErrorCategory.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/connection/UpdateProtocol.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/operation/ChangeStreamBatchCursor.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/MongoSecurityException.java
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/internal/connection/AuthorizationHeader.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright (c) 2008-2014 Atlassian Pty Ltd
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 *   http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources.jar/com/mongodb/assertions/Assertions.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202464273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/ReadPreferenceHedgeOptions.java
* /opt/cola/permits/1202464273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/lang/NonNullApi.java
* /opt/cola/permits/1202464273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/internal/operation/WriteConcernHelper.java
* /opt/cola/permits/1202464273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/lang/Nullable.java
* /opt/cola/permits/1202464273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/lang/NonNull.java
* /opt/cola/permits/1202464273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/TransactionOptions.java
* /opt/cola/permits/1202464273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-jar/com/mongodb/lang/package-info.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.

Open Source Used In DNAC Platform Antelope 3255
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
* Original Work: MIT License, Copyright (c) [2015-2018] all contributors
* https://github.com/marianobarrios/tls-channel
*/

Found in path(s):
* /opt/cola/permits/1202446273_1631568031.75/0/mongodb-driver-core-4-2-3-sources-
jar/com/mongodb/internal/connection/ExtendedAsynchronousByteChannel.java

1.954 hogan 3.0.2
1.954.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.955 react-json-view 1.19.1 1.19.1

1.955.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Mac Gainor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.956 safe-json-stringify 1.2.0 1.2.0

1.956.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.957 javax-inject 2.5.0-b32
1.957.1 Available under license:
CDDL + GPLv2 with classpath exception
https://glassfish.java.net/nonav/public/CDDL+GPL_1_1.html
A business-friendly OSS license

1.958 is-number 7.0.0
1.958.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.959 isarray 0.0.1
1.959.1 Available under license:
No license file was found, but licenses were detected in source scan.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in use, copy, modify, publish, distribute, sublicense, and/or sell copies so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1110632768_1606842153.35/0/fis-components-isarray-0-0-1-0-g1f8a1d4-tar-gz/fis-components-isarray-84d6fbc/README.md

1.960 dashdash 1.14.1 1.14.1
1.960.1 Available under license:
# This is the MIT license

Copyright (c) 2013 Trent Mick. All rights reserved.
Copyright (c) 2013 Joyent Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.961 alsa 1.1.0-0ubuntu1
1.961.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.
We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.
In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library’s complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.
5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
_KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2.1 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

1.962 codemirror 5.59.2 5.59.2
1.962.1 Available under license :
MIT License

Copyright (C) 2017 by Marijn Haverbeke <marijnh@gmail.com> and others

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.963 global-modules 1.0.0

1.963.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.964 unique-string 2.0.0

1.964.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
1.965 ncp 2.0.0

1.965.1 Available under license:

# MIT License

###Copyright (C) 2011 by Charlie McConnell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.966 kafka-clients 2.8.0

1.966.1 Available under license:

Apache Kafka
Copyright 2021 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

This distribution has a binary dependency on jersey, which is available under the CDDL License. The source code of jersey can be found at https://github.com/jersey/jersey/.

The streams-scala (streams/streams-scala) module was donated by Lightbend and the original code was copyrighted by them:
This project contains the following code copied from Apache Hadoop:
clients/src/main/java/org/apache/kafka/common/utils/PureJavaCrc32C.java
Some portions of this file Copyright (c) 2004-2006 Intel Corporation and licensed under the BSD license.

This project contains the following code copied from Apache Hive:
streams/src/main/java/org/apache/kafka/streams/state/internals/Murmur3.java

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.967 sshpk 1.16.1 1.16.1

1.967.1 Available under license:

Copyright Joyent, Inc. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

1.968 form-data 1.0.1 1.0.1
1.968.1 Available under license:
Copyright (c) 2012 Felix Geisendrfer (felix@debuggable.com) and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.969 string-width 4.2.2 4.2.2
1.969.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.970 hbase-replication 2.4.2
1.970.1 Available under license:

Apache HBase - Replication
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.971 istanbuljs-load-nyc-config 1.1.0

1.971.1 Available under license:

ISC License

Copyright (c) 2019, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
 ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.972 akka 2.5.29
1.972.1 Available under license:

Copyright (c) 2002-2006, Marc Prud'hommeaux <mwp1@cornell.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

Open Source Used In DNAC Platform Antelope 3294
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and
attrition notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This software includes the following components from the Apache Software Foundation, also released under the Apache Software License Version 2.0:

commons-cli-1.1.jar
jakarta-oro-2.0.8.jar
log4j-1.2.15.jar

The following components are covered under different licenses:

jline-0.9.94.jar (LICENSE-JLINE)
One-Jar (LICENSE-ONEJAR)
Java/Tcl v1.3.3 (LICENSE-TCLJAVA,-2,-3,-4)
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---------------
Licenses for dependency projects can be found here:
[http://akka.io/docs/akka/snapshot/project/licenses.html]
---------------

akka-protobuf contains the sources of Google protobuf 2.5.0 runtime support,
moved into the source package `akka.protobuf` so as to avoid version conflicts.
For license information see COPYING.protobuf
The following license terms apply to the TJC compiler source
and test files located in the src/tjc, src/tests/tjc, and
tests/tjc directories.

2005 Advanced Micro Devices, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that existing copyright notices
are retained in all copies, this notice is included verbatim in any
distributions, and the terms and conditions hererin are met.

Use of the this software manifests acceptance of the terms of this
The name of Advanced Micro Devices, Inc. may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY ADVANCED MICRO DEVICES, INC. "AS IS" AND ANY EXPRESS, IMPLIED OR STATUTORY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT, OR THOSE ARISING FROM CUSTOM OF TRADE OR COURSE OF USAGE ARE DISCLAIMED.

IN NO EVENT SHALL ADVANCED MICRO DEVICES, INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED, AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE ITS DOCUMENTATION OR ANY DERIVATIVES THEREOF, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. BY USING THIS SOFTWARE WITHOUT CHARGE, YOU ACCEPT THIS ALLOCATION OF RISK. THIS DISCLAIMER OF LIABILITY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. ADVANCED MICRO DEVICES, INC. HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS OF THIS SOFTWARE.

In the redistribution and use of this software, each party shall at all times comply with all applicable governmental laws, statutes, ordinances, rules, regulations, orders, and other requirements, including without limitation such governmental requirements applicable to environmental protection, health, safety, wages, hours, equal employment opportunity, nondiscrimination, working conditions, import or export control, and transportation. Without limiting the foregoing, each party shall adhere to the U.S. Export Administration Regulations (EAR), currently found at 15 C.F.R. Sections 730 through 744, and, unless properly authorized by the U.S. Government, shall not (1) export, re-export or release restricted technology, software, or source code to a national of a country in Country Groups D:1 or E:1, or (2) export to Country Groups D:1 or E:1 the direct product of such technology or software, if such foreign produced direct product is subject to national security controls as identified on the Commerce Control List (currently found in Supplement 1 to Section 774 of EAR). These export requirements shall survive any expiration or termination of this agreement.

# Licenses

## Akka License

```
This software is licensed under the Apache 2 license, quoted below.
```
## Akka Committer License Agreement

All committers have signed this [CLA](http://www.lightbend.com/contribute/current-cla). It can be [signed online](http://www.lightbend.com/contribute/cla).

## Licenses for Dependency Libraries

Each dependency and its license can be seen in the project build file (the comment on the side of each dependency): 
@extref[AkkaBuild.scala](github:project/AkkaBuild.scala#L1054)

Portions of Jacl and Tcl Blend are 
Copyright (c) 1997-1999 The Regents of the University of California.
All rights reserved.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
The following license terms apply to the Itcl source and test files located in the src/itcl and tests/itcl directories.

This software is copyrighted by Cadence Design Systems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

----------------------------------------------------------------------------------------

Following is the original agreement for the Tcl/Tk software from Sun Microsystems.
----------------------------------------------------------------------------------------
This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Apache Commons CLI
Copyright 2001-2007 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
SUN MICROSYSTEMS, INC. THROUGH ITS SUN MICROSYSTEMS LABORATORIES DIVISION ("SUN") WILL LICENSE THIS SOFTWARE AND THE ACCOMPANYING DOCUMENTATION TO YOU (a "Licensee") ONLY ON YOUR ACCEPTANCE OF ALL THE TERMS SET FORTH BELOW.

Sun grants Licensee a non-exclusive, royalty-free right to download, install, compile, use, copy and distribute the Software, modify or otherwise create derivative works from the Software (each, a "Modification") and distribute any Modification in source code and/or binary code form to its customers with a license agreement containing these terms and noting that the Software has been modified. The Software is copyrighted by Sun and other third parties and Licensee shall retain and reproduce all copyright and other notices presently on the Software. As between Sun and Licensee, Sun is the sole owner of all rights in and to the Software other than the limited rights granted to Licensee herein; Licensee will own its Modifications, expressly subject to Sun's continuing ownership of the Software. Licensee will, at its expense, defend and indemnify Sun and its licensors from and against any third party claims, including costs and reasonable attorneys' fees, and be wholly responsible for any
liabilities arising out of or related to Licensee's development, use
or distribution of the Software or Modifications. Any distribution of
the Software and Modifications must comply with all applicable United
States export control laws.

THE SOFTWARE IS BEING PROVIDED TO LICENSEE "AS IS" AND ALL EXPRESS OR
IMPLIED CONDITIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT,
ARE DISCLAIMED. IN NO EVENT WILL SUN BE LIABLE HEREUNDER FOR ANY
DIRECT DAMAGES OR ANY INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES OF ANY KIND.
The jmxsh library has been obtained from https://code.google.com/p/jmxsh/ under the Apache License 2.0.
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.
Apache log4j
Copyright 2007 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

/* ====================================================================
 * The Apache Software License, Version 1.1
 * Copyright (c) 2000-2002 The Apache Software Foundation. All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 * if any, must include the following acknowledgment:
 * "This product includes software developed by the
 * Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgment may appear in the software itself,
 * if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation", "Jakarta-Oro"
 * must not be used to endorse or promote products derived from this
 * software without prior written permission. For written
 * permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache"
 * or "Jakarta-Oro", nor may "Apache" or "Jakarta-Oro" appear in their
 * name, without prior written permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE. */
This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

One-JAR (http://www.simontuffs.com/one-jar). Copyright (c) 2004-2007, P. Simon Tuffs (simon@simontuffs.com). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of P. Simon Tuffs, nor the names of any contributors, nor the name One-JAR may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Including this file inside the built One-JAR file conforms with these terms.

1.973 commons-crypto 1.0.0
1.974 jsreport-exceljs 3.7.1

1.974.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2019 Guyon Roche

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.975 cors 2.8.5

1.975.1 Available under license:

(The MIT License)

Copyright (c) 2013 Troy Goode <troygoode@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.976 referrer-policy 1.0.0 1.0.0

1.976.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Evan Hahn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.977 lodash-es 4.17.15 4.17.15

1.977.1 Available under license :

Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.978 jline-reader 3.12.1

1.978.1 Available under license:
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (c) 2002-2016, the original author or authors.
* *
* This software is distributable under the BSD license. See the terms of the
* BSD license in the documentation provided with this software.
* *
* https://opensource.org/licenses/BSD-3-Clause
Found in path(s):
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/completer/StringsCompleter.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/completer/package-info.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Binding.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/DefaultHighlighter.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Reference.java
* /opt/ cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/completer/NullCompleter.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/package-info.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/completer/AggregateCompleter.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/ReaderUtils.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/history/package-info.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Highlighter.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Widget.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/DefaultExpander.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/UndoTree.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/EndOfFileException.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/ReaderInterruptException.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/completer/EnumCompleter.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Expander.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Macro.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/keymap/KeyMap.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2002-2019, the original author or authors.
*
* This software is distributable under the BSD license. See the terms of the
* BSD license in the documentation provided with this software.
*
* https://opensource.org/licenses/BSD-3-Clause
*/

Found in path(s):
/*
 * Copyright (c) 2002-2017, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/LineReaderImpl.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/BufferImpl.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/SimpleMaskingCallback.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Buffer.java

No license file was found, but licenses were detected in source scan.

2016, the original author or authors.

This software is distributable under the BSD license. See the terms of the
BSD license in the documentation provided with this software.

https://opensource.org/licenses/BSD

Found in path(s):
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/META-INF/maven/org.jline/jline-reader/pom.xml

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/EOFError.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/SyntaxError.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2002-2018, the original author or authors.
*
* This software is distributable under the BSD license. See the terms of the
* BSD license in the documentation provided with this software.
* 
* https://opensource.org/licenses/BSD-3-Clause
*/

Found in path(s):
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/DefaultParser.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/LineReader.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/MaskingCallback.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/LineReaderBuilder.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/history/DefaultHistory.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Candidate.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/Completer.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/completer/FileNameCompleter.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/ParsedLine.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/History.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/keymap/BindingReader.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/CompletingParsedLine.java
* /opt/cola/permits/1114600224_1607534234.11/0/jline-reader-3-12-1-sources-jar/org/jline/reader/impl/completer/ArgumentCompleter.java

1.979 combined-stream 1.0.8 1.0.8
1.979.1 Available under license:
Copyright (c) 2011 Debuggable Limited <felix@debuggable.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.980 debug 2.2.0

1.980.1 Available under license:
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.981 jsreport-html-to-xlsx 2.8.3

1.981.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.
You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.982 jackson-databind 2.11.4

1.982.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.
## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

### 1.983 snake-yaml 1.26

#### 1.983.1 Available under license:

<table>
<thead>
<tr>
<th>License Type</th>
<th>License URI</th>
<th>License URI Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPL</td>
<td><a href="http://www.eclipse.org/legal">http://www.eclipse.org/legal</a></td>
<td>Eclipse Public License, V1.0 or later</td>
</tr>
<tr>
<td>LGPL</td>
<td><a href="http://www.gnu.org/licenses/lgpl.html">http://www.gnu.org/licenses/lgpl.html</a></td>
<td>GNU Lesser General Public License, V2.1 or later</td>
</tr>
<tr>
<td>GPL</td>
<td><a href="http://www.gnu.org/licenses/gpl.html">http://www.gnu.org/licenses/gpl.html</a></td>
<td>GNU General Public License, V2 or later</td>
</tr>
<tr>
<td>AL</td>
<td><a href="http://www.apache.org/licenses">http://www.apache.org/licenses</a></td>
<td>Apache License, V2.0 or later</td>
</tr>
<tr>
<td>BSD</td>
<td><a href="http://www.opensource.org/licenses/bsd-license.php">http://www.opensource.org/licenses/bsd-license.php</a></td>
<td>BSD License</td>
</tr>
</tbody>
</table>

### 1.984 fresh 0.5.2

#### 1.984.1 Available under license:

(The MIT License)

Copyright (c) 2012 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2016-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be...
1.985 hive-shims 3.1.2

1.985.1 Available under license:

Hive Shims
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation.
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.986 grpc-stub 1.27.1
1.986.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date
such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.987 isbinaryfile 3.0.3
1.987.1 Available under license:

Copyright (c) 2017 Garen J. Torikian

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.988 guice-assistedinject 3.0
1.988.1 Available under license:

Google Guice - Extensions - AssistedInject
Copyright 2006-2011 Google, Inc.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.989 babel-helper-explode-assignable-expression 6.24.1

1.989.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is
being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
   following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
   from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.990 babel-plugin-transform-es2015-object-super 6.24.1
1.990.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.
Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)
Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.991 parseqs 0.0.5
1.991.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.992 is-binary-path 2.1.0

1.992.1 Available under license:
MIT License

Copyright (c) 2019 Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com), Paul Miller (https://paulmillr.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.993 helmet-csp 1.2.2

1.993.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.994 lodash.debounce 4.0.8

1.994.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====
Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

### 1.995 frameguard 2.0.0

**1.995.1 Available under license:**

The MIT License (MIT)

Copyright (c) 2014-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.996 has-binary2 1.0.3

**1.996.1 Available under license:**

The MIT License (MIT)

Copyright (c) 2014 Kevin Roark

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.997 jetty-http 9.4.17.v20190418

1.997.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when
combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties
and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose; ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product
offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced
with your own identifying information. (Don't include the brackets!) The text
should be enclosed in the appropriate comment syntax for the file format. We also recommend
that a file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier identification
within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

=====================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================
The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License  
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License  

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.  
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.  
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.  
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.  
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api  
* javax.annotation:javax.annotation-api  
* javax.transaction:javax.transaction-api  
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl
These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.  
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.998 formidable 1.2.2

1.998.1 Available under license:

The MIT License (MIT)

Copyright (c) 2011-present Felix Geisendrfer, and contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.999 snake-yaml 1.12
1.999.1 Available under license:

Found license 'GNU Lesser General Public License' in '// This module is multi-licensed and may be used under the terms // EPL, Eclipse Public License, V1.0 or later, http://www.eclipse.org/legal // LGPL, GNU Lesser General Public License, V2.1 or later, http://www.gnu.org/licenses/lgpl.html // GPL, GNU General Public License, V2 or later, http://www.gnu.org/licenses/gpl.html // AL, Apache License, V2.0 or later, http://www.apache.org/licenses // BSD, BSD License, http://www.opensource.org/licenses/bsd-license.php Multi-licensed: EPL / LGPL / GPL / AL / BSD.'

Found license 'Eclipse Public License 1.0' in '// This module is multi-licensed and may be used under the terms // EPL, Eclipse Public License, V1.0 or later, http://www.eclipse.org/legal // LGPL, GNU Lesser General Public License, V2.1 or later, http://www.gnu.org/licenses/lgpl.html // GPL, GNU General Public License, V2 or later, http://www.gnu.org/licenses/gpl.html // AL, Apache License, V2.0 or later, http://www.apache.org/licenses // BSD, BSD License, http://www.opensource.org/licenses/bsd-license.php Multi-licensed: EPL / LGPL / GPL / AL / BSD.'

1.1000 redux 3.7.2 3.7.2

1.1000.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1001 @babel/helper-split-export-declaration

7.12.13
1.1001.1 Available under license :

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1002 jetty 9.4.41.v20210516

1.1002.1 Available under license :

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and  
ii) additions to the Program;  

where such changes and/or additions to the Program originate from and are 
distributed by that particular Contributor. A Contribution 'originates' 
from a Contributor if it was added to the Program by such Contributor 
itself or anyone acting on such Contributor's behalf. Contributions do not 
include additions to the Program which: (i) are separate modules of 
software distributed in conjunction with the Program under their own 
license agreement, and (ii) are not derivative works of the Program.  

"Contributor" means any person or entity that distributes the Program.  

"Licensed Patents" mean patent claims licensable by a Contributor which are 
necessarily infringed by the use or sale of its Contribution alone or when 
combined with the Program.  

"Program" means the Contributions distributed in accordance with this 
Agreement.  

"Recipient" means anyone who receives the Program under this Agreement, 
including all Contributors.  

2. GRANT OF RIGHTS  
a) Subject to the terms of this Agreement, each Contributor hereby grants 
Recipient a non-exclusive, worldwide, royalty-free copyright license to 
reproduce, prepare derivative works of, publicly display, publicly 
perform, distribute and sublicense the Contribution of such Contributor, 
if any, and such derivative works, in source code and object code form.  
b) Subject to the terms of this Agreement, each Contributor hereby grants 
Recipient a non-exclusive, worldwide, royalty-free patent license under 
Licensed Patents to make, use, sell, offer to sell, import and otherwise 
transfer the Contribution of such Contributor, if any, in source code and 
object code form. This patent license shall apply to the combination of 
the Contribution and the Program if, at the time the Contribution is 
added by the Contributor, such addition of the Contribution causes such 
combination to be covered by the Licensed Patents. The patent license 
shall not apply to any other combinations which include the Contribution. 
No hardware per se is licensed hereunder.  
c) Recipient understands that although each Contributor grants the licenses 
to its Contributions set forth herein, no assurances are provided by any 
Contributor that the Program does not infringe the patent or other 
intellectual property rights of any other entity. Each Contributor 
disclaims any liability to Recipient for claims brought by any other 
entity based on infringement of intellectual property rights or 
otherwise. As a condition to exercising the rights and licenses granted 
hereunder, each Recipient hereby assumes sole responsibility to secure 
any other intellectual property rights needed, if any. For example, if a
third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS.
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

=====================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions. http://openjdk.java.net/legal/gplv2+ce.html

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

### 1.1003 uri-js 4.2.2 4.2.2

#### 1.1003.1 Available under license:

```xml
<OWNER> = Regents of the University of California
<ORGANIZATION> = University of California, Berkeley
<YEAR> = 1998
```

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1004 asm-commons 7.3.1
1.1004.1 Available under license:
No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
//    this software without specific prior written permission.
/**
 * Constructs a new {@link LocalVariablesSorter}. <i>Subclasses must not use this constructor</i>.
 * Instead, they must use the {@link #LocalVariablesSorter(int, int, String, MethodVisitor)}
 * version.
 * @param access access flags of the adapted method.
 * @param descriptor the method's descriptor (see {@link Type}).
 * @param methodVisitor the method visitor to which this adapter delegates calls.
* @throws IllegalStateException if a subclass calls this constructor.
*/

Found in path(s):
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
jar/org/objectweb/asm/commons/LocalVariablesSorter.java
No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
jar/org/objectweb/asm/commons/package.html
No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
//    this software without specific prior written permission.
/**
 * Constructs a new {@link SerialVersionUIDAdder}. <i>Subclasses must not use this
 * constructor</i>. Instead, they must use the {@link #SerialVersionUIDAdder(int, ClassVisitor)}
 * version.
 *
 * @param classVisitor a {@link ClassVisitor} to which this visitor will delegate calls.
 * @throws IllegalStateException If a subclass calls this constructor.
 */

Found in path(s):
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/SerialVersionUIDAdder.java

No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
//    this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/StaticInitMerger.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/ModuleResolutionAttribute.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/AnalyzerAdapter.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/ClassRemapper.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/SignatureRemapper.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/Method.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/AdviceAdapter.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/Remapper.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-
  jar/org/objectweb/asm/commons/MethodRemapper.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/SimpleRemapper.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/AnnotationRemapper.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/ModuleTargetAttribute.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/FieldRemapper.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/CodeSizeEvaluator.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/TableSwitchGenerator.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/ModuleHashesAttribute.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/InstructionAdapter.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/JSRInlinerAdapter.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/TryCatchBlockSorter.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/ModuleRemapper.java
* /opt/cola/permits/1135907511_1613628615.48/0/asm-commons-7-3-1-sources-1-jar/org/objectweb/asm/commons/RecordComponentRemapper.java

No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
// this software without specific prior written permission.
/**
 * Constructs a new { @link GeneratorAdapter}. <i>Subclasses must not use this constructor</i>.
 * Instead, they must use the { @link #GeneratorAdapter(int, MethodVisitor, int, String, String)}
 * version.
 * @param methodVisitor the method visitor to which this adapter delegates calls.
 * @param access the method's access flags (see { @link Opcodes}).
 * @param name the method's name.
 * @param descriptor the method's descriptor (see { @link Type}).
 * @throws IllegalStateException if a subclass calls this constructor.
 */
1.1005 drill-format-spss 1.19.0

1.1005.1 Available under license:

Drill : Contrib : Format : SPSS
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

Open Source Used In DNAC Platform Antelope 3384
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1006 zlib 1.2.11

1.1006.1 Available under license :
/* zlib.h -- interface of the 'zlib' general purpose compression library
  version 1.2.11, January 15th, 2017

Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
Open Source Used In DNAC Platform Antelope  3388

misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu

*/
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

1.1007 is-glob 2.0.1
1.1007.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1008 is-plain-object 2.0.4 2.0.4
1.1008.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1009 drill-mongo-storage 1.19.0
1.1009.1 Available under license :

Drill : Contrib : Storage : MongoDB
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1010 parquet-format 2.4.0

1.1010.1 Available under license:

Apache Parquet Format
Copyright 2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

--------------------------------------------------------------------------------

This product depends on SLF4J and includes SLF4J in this binary artifact. SLF4J is a simple logging facade for Java.

Copyright: 2004-2013 QOS.ch.
Home page: http://www.slf4j.org/
License: http://slf4j.org/license.html (MIT license)

The following is the SLF4J license (MIT):

Copyright (c) 2004-2013 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---------------------------------------------------------------------------------------------------------------------------------------

This product depends on Apache Thrift and includes it in this binary artifact.

Copyright: 2006-2010 The Apache Software Foundation.
Home page: https://thrift.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

1.1011.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1012 strip-ansi 6.0.0
1.1012.1 Available under license :
   MIT License

   Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
   Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
   INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
   PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
   COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
   AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
   WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1013 websocket-client 9.4.17.v20190418
1.1013.1 Available under license :
   This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at
   http://www.eclipse.org/legal/epl-2.0, or the Apache Software License
   2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

   Eclipse Public License - v 1.0

   THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
   LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
   CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

   1. DEFINITIONS

   "Contribution" means:

   a) in the case of the initial Contributor, the initial code and documentation
distributed under this Agreement, and
   b) in the case of each subsequent Contributor:
      i) changes to the Program, and
ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute
the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:
a) promptly notify the Commercial Contributor in writing of such claim, and 
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL
If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html
* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions. http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1014 bson 1.1.6 1.1.6

1.1014.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1015 mem 4.3.0

1.1015.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1016 eclipse-egit-github-core 2.1.5

1.1016.1 Available under license:
/* *************************************************************************/
/* Copyright (c) 2011 GitHub Inc. */
/* All rights reserved. This program and the accompanying materials */
/* are made available under the terms of the Eclipse Public License v1.0 */
/* which accompanies this distribution, and is available at */
/* */
/* Contributors: */
/* Kevin Sawicki (GitHub Inc.) - initial API and implementation */
/*****************************************************************************/
package org.eclipse.egit.github.core;

import java.io.Serializable;

/** */
/** * Repository contributor model class */
/** */
public class Contributor implements Serializable {

/** serialVersionUID */
private static final long serialVersionUID = -8434028880839230626L;
/**
 * Anonymous contributor type value
 */
public static final String TYPE_ANONYMOUS = "Anonymous"; //SNON-NLS-1S

private int contributions;

private int id;

private String avatarUrl;

private String login;

private String name;

private String type;

private String url;

/**
 * @return contributions
 */
public int getContributions() {
    return contributions;
}

/**
 * @param contributions
 * @return this contributor
 */
public Contributor setContributions(int contributions) {
    this.contributions = contributions;
    return this;
}

/**
 * @return id
 */
public int getId() {
    return id;
}

/**
 * @param id
 * @return this contributor
 */
public Contributor setId(int id) {
```java
this.id = id;
return this;
}

/**
 * @return avatarUrl
 */
public String getAvatarUrl() {
return avatarUrl;
}

/**
 * @param avatarUrl
 * @return this contributor
 */
public Contributor setAvatarUrl(String avatarUrl) {
this.avatarUrl = avatarUrl;
return this;
}

/**
 * @return login
 */
public String getLogin() {
return login;
}

/**
 * @param login
 * @return this contributor
 */
public Contributor setLogin(String login) {
this.login = login;
return this;
}

/**
 * @return name
 */
public String getName() {
return name;
}

/**
 * @param name
 * @return this contributor
 */
public Contributor setName(String name) {
```
this.name = name;
return this;
}

/**
 * @return type
 */
public String getType() {
return type;
}

/**
 * @param type
 * @return this contributor
 */
public Contributor setType(String type) {
this.type = type;
return this;
}

/**
 * @return url
 */
public String getUrl() {
return url;
}

/**
 * @param url
 * @return this contributor
 */
public Contributor setUrl(String url) {
this.url = url;
return this;
}

1.1017 proj4j 0.1.0

1.1017.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
Copyright 2011 Martin Davis

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Open Source Used In DNAC Platform Antelope 3419
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/Eckert6Projection.java
No license file was found, but licenses were detected in source scan.

*/

Copyright 2006 Jerry Huxtable

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/PerspectiveConicProjection.java
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/UrmaevFlatPolarSinusoidalProjection.java
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/TransverseCylindricalEqualArea.java
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/MercatorProjection.java
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/HammerProjection.java
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/BipolarProjection.java
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/ConicProjection.java
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/...
No license file was found, but licenses were detected in source scan.

/*
Copyright 2006 Martin Davis

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/proj/SwissObliqueMercatorProjection.java
No license file was found, but licenses were detected in source scan.

/*
Copyright 2006 Jerry Huxtable

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446574_1631568180.3/0/proj4j-0-1-0-sources-jar/org/osgeo/proj4j/datum/Ellipsoid.java

1.1018 has-cors 1.1.0
1.1018.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1019 jsonpointer 4.0.1 4.0.1

1.1019.1 Available under license :

The MIT License (MIT)

Copyright (c) 2011-2015 Jan Lehnardt <jan@apache.org> & Marc Bachmann <https://github.com/marcbachmann>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1020 strip-ansi 3.0.1 3.0.1

1.1020.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1021 browserslist 3.2.8 3.2.8
1.1021.1 Available under license:
The MIT License (MIT)

Copyright 2014 Andrey Sitnik <andrey@sitnik.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1022 check-error 1.0.2
1.1022.1 Available under license:
Copyright (c) 2013 Jake Luer <jake@alogicalparadox.com> (http://alogicalparadox.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1023 yeast 0.1.2
1.1023.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1024 optionator 0.8.2
1.1024.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

1.1025 is-relative 1.0.0
1.1025.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.1026 mkdirp 0.5.5 0.5.5
1.1026.1 Available under license:
Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1027 babel-plugin-transform-es2015-unicode-regex 6.24.1
1.1027.1 Available under license:
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is
being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1028 hk2-api 2.5.0-b32

1.1028.1 Available under license:

CDDL + GPLv2 with classpath exception
https://glassfish.java.net/nonav/public/CDDL+GPL_1_1.html
A business-friendly OSS license

1.1029 babel-messages 6.23.0 6.23.0
1.1029.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1030 base16 1.0.0 1.0.0

1.1030.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1031 failureaccess 1.0.1

1.1031.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1199742917_1630577017.47/0/failureaccess-1-0-1-sources-3-
  jar/com/google/common/util/concurrent/internal/InternalFutureFailureAccess.java
* /opt/cola/permits/1199742917_1630577017.47/0/failureaccess-1-0-1-sources-3-
  jar/com/google/common/util/concurrent/internal/InternalFutures.java

1.1032 debug 2.3.3 2.3.3

1.1032.1 Available under license:
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
1.1033 jetty-client 9.4.17.v20190418

1.1033.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.
Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

“Program” means the Contributions distributed in accordance with this Agreement.

“Recipient” means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution.

No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:
a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties
      and conditions, express and implied, including warranties or
      conditions of title and non-infringement, and implied warranties or
      conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for
      damages, including direct, indirect, special, incidental and
      consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are
      offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such
      Contributor, and informs licensees how to obtain it in a reasonable
      manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
   Contributors may not remove or alter any copyright notices contained
   within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
   any, in a manner that reasonably allows subsequent Recipients to identify the
   originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore,
if a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and
other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such Commercial
Contributor in connection with its distribution of the Program in a commercial
product offering. The obligations in this section do not apply to any claims
or Losses relating to any actual or alleged intellectual property
infringement. In order to qualify, an Indemnified Contributor must:
   a) promptly notify the Commercial Contributor in writing of such claim, and
   b) allow the Commercial Contributor to control, and cooperate with the
   Commercial Contributor in, the defense and any related settlement
   negotiations. The Indemnified Contributor may participate in any such claim at
its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted
under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient’s rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient’s rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

====================================================================

Jetty Web Container
The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN
module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1034 append-field 0.1.0 0.1.0

1.1034.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Linus Unnebeck

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of
the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the
Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULDN'T ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.
6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. "252.227-7014(a)(1)") and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License
is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new
free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you
also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special
exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this document.
License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

/*
 * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
 *
 * Copyright (c) 2010-2011 Oracle and/or its affiliates. All rights reserved.
 *
 * The contents of this file are subject to the terms of either the GNU
 * General Public License Version 2 only ("GPL") or the Common Development
 * and Distribution License("CDDL") (collectively, the "License"). You
 * may not use this file except in compliance with the License. You can
 * obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to such option by the copyright
* holder.
*/

1.1036 uuid 3.3.2 3.3.2
1.1036.1 Available under license :

The MIT License (MIT)

Copyright (c) 2010-2016 Robert Kieffer and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1037 resourcelocator 1.0.3
1.1037.1 Available under license:

Found license 'Eclipse Public License 1.0' in*: Copyright (c) 2009, 2018 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'
Found license 'General Public License 2.0' in*: Copyright (c) 2009, 2018 Oracle and/or its affiliates. All rights reserved. * This program and the accompanying materials are made available under the * terms of the Eclipse Public License v. 2.0, which is available at * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,'

1.1038 pkg-dir 3.0.0
1.1038.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1039 stdio 0.2.2
**1.1039.1 Available under license:**

(The MIT License)

Copyright (c) 2013- Sergio Garca <sgmonda@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**1.1040 swagger-annotations 2.0.7**

**1.1040.1 Available under license:**

```java
/**
 * Copyright 2017 SmartBear Software
 * <p>
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * <p>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p>
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package io.swagger.v3.oas.annotations.info;

import io.swagger.v3.oas.annotations.extensions.Extension;
```
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
 * The annotation may be used in { @link Info#license()} to define a license for the OpenAPI spec.
 * 
 * @see <a target="_new" href="https://github.com/OAI/OpenAPI-Specification/blob/3.0.1/versions/3.0.1.md#licenseObject">License (OpenAPI specification)</a>
 * @see io.swagger.v3.oas.annotations.OpenAPIDefinition
 * @see Info
 **/
@Target({})
@Retention(RetentionPolicy.RUNTIME)
public @interface License {

/**
 * The license name used for the API.
 */
@Retention(RetentionPolicy.RUNTIME)
public interface License {

/**
 * @return the name of the license
 */
String name() default "";

/**
 * A URL to the license used for the API. MUST be in the format of a URL.
 */
String url() default "";

/**
 * The list of optional extensions
 */
Extension[] extensions() default {};
}

1.1041 zstd 1.4.9
1.1041.1 Available under license:
BSD License

For Zstandard software

Copyright (c) 2016-present, Facebook, Inc. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name Facebook nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of
thus License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANDABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

1.1042 jetty-servlet 9.4.17.v20190418
1.1042.1 Available under license :

This program and the accompanying materials are made available under the
terms of the Eclipse Public License 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0, or the Apache Software License
2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:
a) in the case of the initial Contributor, the initial code and documentation
distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;

where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution ‘originates’
from a Contributor if it was added to the Program by such Contributor
itself or anyone acting on such Contributor's behalf. Contributions do not
include additions to the Program which: (i) are separate modules of
software distributed in conjunction with the Program under their own
license agreement, and (ii) are not derivative works of the Program.

“Contributor” means any person or entity that distributes the Program.

“Licensed Patents” mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when
combined with the Program.

"Program” means the Contributions distributed in accordance with this
Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free copyright license to
reproduce, prepare derivative works of, publicly display, publicly
perform, distribute and sublicense the Contribution of such Contributor,
if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free patent license under
Licensed Patents to make, use, sell, offer to sell, import and otherwise
transfer the Contribution of such Contributor, if any, in source code and
object code form. This patent license shall apply to the combination of
the Contribution and the Program if, at the time the Contribution is
added by the Contributor, such addition of the Contribution causes such
combination to be covered by the Licensed Patents. The patent license
shall not apply to any other combinations which include the Contribution.
No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses
to its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other
intellectual property rights of any other entity. Each Contributor
disclaims any liability to Recipient for claims brought by any other
entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of the
remainder of the terms of this Agreement, and without further action by the
parties hereto, such provision shall be reformed to the minimum extent
necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Program itself
(excluding combinations of the Program with other software or hardware)
infringes such Recipient's patent(s), then such Recipient's rights granted
under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to
comply with any of the material terms or conditions of this Agreement and does
not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate,
Recipient agrees to cease use and distribution of the Program as soon as
reasonably practicable. However, Recipient's obligations under this Agreement
and any licenses granted by Recipient relating to the Program shall continue
and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
order to avoid inconsistency the Agreement is copyrighted and may only be
modified in the following manner. The Agreement Steward reserves the right to
publish new versions (including revisions) of this Agreement from time to
time. No one other than the Agreement Steward has the right to modify this
Agreement. The Eclipse Foundation is the initial Agreement Steward. The
Eclipse Foundation may assign the responsibility to serve as the Agreement
Steward to a suitable separate entity. Each new version of the Agreement will
be given a distinguishing version number. The Program (including
Contributions) may always be distributed subject to the version of the
Agreement under which it was received. In addition, after a new version of the
Agreement is published, Contributor may elect to distribute the Program
(including its Contributions) under the new version. Except as expressly
stated in Sections 2(a) and 2(b) above, Recipient receives no rights or
licenses to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in the
Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==============================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
  * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
  * org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
  * org.eclipse.jetty.orbit:javax.mail.glassfish
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.  
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.  
http://openjdk.java.net/legal/gplv2+ce.html

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay
The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1043 httpcomponents-client 4.5.6
1.1043.1 Available under license :

Apache HttpClient
Copyright 1999-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1044 regex-cache 0.4.4
1.1044.1 Available under license :
    The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1045 bluebird 3.7.2
1.1045.1 Available under license :
    The MIT License (MIT)

Copyright (c) 2013-2018 Petka Antonov

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1046 on-finished 2.3.0 2.3.0

1.1046.1 Available under license :

(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1047 has-yarn 2.1.0
1.1047.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1048 protobuf 2.6.1-1.3

1.1048.1 Available under license:
This file contains a list of people who have made large contributions to the public version of Protocol Buffers.

Original Protocol Buffers design and implementation:
Sanjay Ghemawat <sanjay@google.com>
Jeff Dean <jeff@google.com>
Daniel Dulitz <daniel@google.com>
Craig Silverstein
Paul Haahr <haahr@google.com>
Corey Anderson <corin@google.com>
(and many others)

Proto2 C++ and Java primary author:
Kenton Varda <kenton@google.com>

Proto2 Python primary authors:
Will Robinson <robinson@google.com>
Petar Petrov <petar@google.com>

Large code contributions:
Jason Hsueh <jasonh@google.com>
Joseph Schorr <jschorr@google.com>
Wenbo Zhu <wenboz@google.com>
Large quantity of code reviews:
Scott Bruce <sbruce@google.com>
Frank Yellin
Neal Norwitz <nnorwitz@google.com>
Jeffrey Yasskin <jyasskin@google.com>
Ambrose Feinstein <ambrose@google.com>

Documentation:
Lisa Carey <lcarey@google.com>

Maven packaging:
Gregory Kick <gak@google.com>

Patch contributors:
Kevin Ko <kevin.s.ko@gmail.com>
  * Small patch to handle trailing slashes in --proto_path flag.
Johan Euphrosine <proppy@aminche.com>
  * Small patch to fix Python CallMethod().
Ulrich Kunitz <kune@deine-taler.de>
  * Small optimizations to Python serialization.
Leandro Lucarella <llucax@gmail.com>
  * VI syntax highlighting tweaks.
  * Fix compiler to not make output executable.
Dilip Joseph <dilip.antony.joseph@gmail.com>
  * Heuristic detection of sub-messages when printing unknown fields in
text format.
Brian Atkinson <nairb774@gmail.com>
  * Added @Override annotation to generated Java code where appropriate.
Vincent Choinire <Choiniere.Vincent@hydro.qc.ca>
  * Tru64 support.
Monty Taylor <monty.taylor@gmail.com>
  * Solaris 10 + Sun Studio fixes.
Alek Storm <alek.storm@gmail.com>
  * Slicing support for repeated scalar fields for the Python API.
Oleg Smolsky <oleg.smolsky@gmail.com>
  * MS Visual Studio error format option.
  * Detect unordered_map in stl_hash.m4.
Brian Olson <brianolson@google.com>
  * gzip/zlib I/O support.
Michael Poole <mdpoole@troilus.org>
  * Fixed warnings about generated constructors not explicitly initializing
    all fields (only present with certain compiler settings).
  * Added generation of field number constants.
Wink Saville <wink@google.com>
  * Fixed initialization ordering problem in logging code.
Will Pierce <willp@nuclei.com>
  * Small patch improving performance of in Python serialization.
Alexandre Vassalotti <alexandre@peadrop.com>
* Emacs mode for Protocol Buffers (editors/protobuf-mode.el).
  Scott Stafford <scott.stafford@gmail.com>
* Added Swap(), SwapElements(), and RemoveLast() to Reflection interface.
  Alexander Melnikov <alm@sibmail.ru>
* HPUX support.
  Oliver Jowett <oliver.jowett@gmail.com>
  * Detect whether zlib is new enough in configure script.
  * Fixes for Solaris 10 32/64-bit confusion.
  Evan Jones <evanj@mit.edu>
  * Optimize Java serialization code when writing a small message to a stream.
  * Optimize Java serialization of strings so that UTF-8 encoding happens only once per string per serialization call.
* Clean up some Java warnings.
  * Fix bug with permanent callbacks that delete themselves when run.
  Michael Kucharski <m.kucharski@gmail.com>
  * Added CodedInputStream.getTotalBytesRead().
  Kacper Kowalik <xarthisius.kk@gmail.com>
  * Fixed m4/acx_pthread.m4 problem for some Linux distributions.
  William Orr <will@worrbase.com>
  * Fixed detection of sched_yield on Solaris.
  * Added atomicops for Solaris

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

# This file contains a list of people who've made non-trivial
# contribution to the Google C++ Testing Framework project. People
# who commit code to the project are encouraged to add their names
# here. Please keep the list sorted by first names.

Ajay Joshi <jaj@google.com>
Balzs Dn <balazs.dan@gmail.com>
Bharat Mediratta <bharat@menalto.com>
Chandler Carruth <chandlerc@google.com>
Chris Prince <cprince@gmail.com>
Chris Taylor <taylorc@gmail.com>
Dan Egnor <egnor@gmail.com>
Eric Roman <eroman@chromium.org>
Hady Zalek <hady.zalek@gmail.com>
Open Source Used In DNAC Platform Antelope

Jeffrey Yasskin <jyasskin@google.com>
Ji Sigursson <joi@google.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kenton Varda <kenton@google.com>
Manuel Klimek <klimek@google.com>
Markus Heule <markus.heule@gmail.com>
Mika Raento <mikie@iki.fi>
Mikls Fazekas <mfazekas@szemafor.com>
Pasi Valminen <pasi.valminen@gmail.com>
Patrick Hanna <phanna@gmail.com>
Patrick Riley <pfr@gmail.com>
Peter Kaminski <piotrsk@google.com>
Preston Jackson <preston.a.jackson@gmail.com>
Rainer Klaffenboeck <rainer.klaffenboeck@dynatrace.com>
Russ Cox <rsc@google.com>
Russ Rufer <russ@pentad.com>
Sean McAfee <seefacman@gmail.com>
Sigurur sgeirsson <siggi@google.com>
Tracy Bialik <tracy@pentad.com>
Vadim Berman <vadimb@gmail.com>
Vlad Losev <vladl@gmail.com>
Zhanyong Wan <wan@gmail.com>

1.1049 js-tokens 4.0.0
1.1049.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014, 2015, 2016, 2017, 2018 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1050 resolve-from 2.0.0

1.1050.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1051 jackson-annotations 2.12.1

1.1051.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1052 serializator 1.0.2
1.1052.1 Available under license:
The MIT License (MIT)

Copyright (c) 2018 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1053 duplexer 0.1.1 0.1.1
1.1053.1 Available under license:

Copyright (c) 2012 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1054 cookie-signature 1.0.6 1.0.6

1.1054.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1055 abbrev 1.0.9

1.1055.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
/*
 * Apache License
 * Version 2.0, January 2004
 * http://www.apache.org/licenses/
 *
 * TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
 *
 * 1. Definitions.
 *
 * "License" shall mean the terms and conditions for use, reproduction,
 * and distribution as defined by Sections 1 through 9 of this document.
 *
 * "Licensor" shall mean the copyright owner or entity authorized by
 * the copyright owner that is granting the License.
 *
 * "Legal Entity" shall mean the union of the acting entity and all
 * other entities that control, are controlled by, or are under common
 * control with that entity. For the purposes of this definition,
 * "control" means (i) the power, direct or indirect, to cause the
 * direction or management of such entity, whether by contract or
 * otherwise, or (ii) ownership of fifty percent (50%) or more of the
 * outstanding shares, or (iii) beneficial ownership of such entity.
 *
 * "You" (or "Your") shall mean an individual or Legal Entity
 * exercising permissions granted by this License.
 *
 * "Source" form shall mean the preferred form for making modifications,
 * including but not limited to software source code, documentation
 * source, and configuration files.
 *
 * "Object" form shall mean any form resulting from mechanical
 * transformation or translation of a Source form, including but
 * not limited to compiled object code, generated documentation,
 * and conversions to other media types.
 *
 * "Work" shall mean the work of authorship, whether in Source or
 * Object form, made available under the License, as indicated by a
 * copyright notice that is included in or attached to the work
 * (an example is provided in the Appendix below).
 *
 * "Derivative Works" shall mean any work, whether in Source or Object
 * form, that is based on (or derived from) the Work and for which the
 * editorial revisions, annotations, elaborations, or other modifications
* represent, as a whole, an original work of authorship. For the purposes
* of this License, Derivative Works shall not include works that remain
* separable from, or merely link (or bind by name) to the interfaces of,
* the Work and Derivative Works thereof.
*
* "Contribution" shall mean any work of authorship, including
* the original version of the Work and any modifications or additions
* to that Work or Derivative Works thereof, that is intentionally
* submitted to Licensor for inclusion in the Work by the copyright owner
* or by an individual or Legal Entity authorized to submit on behalf of
* the copyright owner. For the purposes of this definition, "submitted"
* means any form of electronic, verbal, or written communication sent
* to the Licensor or its representatives, including but not limited to
* communication on electronic mailing lists, source code control systems,
* and issue tracking systems that are managed by, or on behalf of, the
* Licensor for the purpose of discussing and improving the Work, but
* excluding communication that is conspicuously marked or otherwise
* designated in writing by the copyright owner as "Not a Contribution."
*
* "Contributor" shall mean Licensor and any individual or Legal Entity
* on behalf of whom a Contribution has been received by Licensor and
* subsequently incorporated within the Work.
*
* 2. Grant of Copyright License. Subject to the terms and conditions of
* this License, each Contributor hereby grants to You a perpetual,
* worldwide, non-exclusive, no-charge, royalty-free, irrevocable
* copyright license to reproduce, prepare Derivative Works of,
* publicly display, publicly perform, sublicense, and distribute the
* Work and such Derivative Works in Source or Object form.
*
* 3. Grant of Patent License. Subject to the terms and conditions of
* this License, each Contributor hereby grants to You a perpetual,
* worldwide, non-exclusive, no-charge, royalty-free, irrevocable
* (except as stated in this section) patent license to make, have made,
* use, offer to sell, sell, import, and otherwise transfer the Work,
* where such license applies only to those patent claims licensable
* by such Contributor that are necessarily infringed by their
* Contribution(s) alone or by combination of their Contribution(s)
* with the Work to which such Contribution(s) was submitted. If You
* institute patent litigation against any entity (including a
* cross-claim or counterclaim in a lawsuit) alleging that the Work
* or a Contribution incorporated within the Work constitutes direct
* or contributory patent infringement, then any patent licenses
* granted to You under this License for that Work shall terminate
* as of the date such litigation is filed.
*
* 4. Redistribution. You may reproduce and distribute copies of the
* Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
Open Source Used In DNAC Platform - Antelope 3504

* identification within third-party archives.
* Copyright [yyyy] [name of copyright owner]
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* https://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
limitations under the License.
*/

W3C SOFTWARE NOTICE AND LICENSE
https://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

____________________________________

This formulation of W3C’s notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered under http://saxproject.org/?selected=pd

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05
Apache Ant
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.
1.1057 urix 0.1.0
1.1057.1 Available under license :
The MIT License (MIT)

Copyright (c) 2013 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1058 babel-register 6.26.0
1.1058.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1059 normalize-url 4.5.0

1.1059.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1060 babel-helper-remap-async-to-generator 6.24.1

1.1060.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

MIT License
Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1061 has-value 1.0.0
1.1061.1 Available under license :
Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE
LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN
ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS
INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES
REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS
PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM
THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED
HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer
exclusive Copyright and Related Rights (defined below) upon the creator
and subsequent owner(s) (each and all, an "owner") of an original work of
authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for
the purpose of contributing to a commons of creative, cultural and
scientific works ("Commons") that the public can reliably and without fear
of later claims of infringement build upon, modify, incorporate in other
works, reuse and redistribute as freely as possible in any form whatsoever
and for any purposes, including without limitation commercial purposes.
These owners may contribute to the Commons to promote the ideal of a free
culture and the further production of creative, cultural and scientific
works, or to gain reputation or greater distribution for their Work in
part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
ii. moral rights retained by the original author(s) and/or performer(s);
iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public.
3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.
1.1062 @babel/plugin-proposal-export-namespace-from 7.12.13
1.1062.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1063 validate-npm-package-license 3.0.4
1.1063.1 Available under license:
validate-npm-package-license

Give me a string and I'll tell you if it's a valid npm package license string.

```javascript
var valid = require('validate-npm-package-license');
```

SPDX license identifiers are valid license strings:

```javascript
var assert = require('assert');
var validSPDXExpression = {
```
validForNewPackages: true,
validForOldPackages: true,
spdx: true
};

assert.deepEqual(valid('MIT'), validSPDXExpression);
assert.deepEqual(valid('BSD-2-Clause'), validSPDXExpression);
assert.deepEqual(valid('Apache-2.0'), validSPDXExpression);
assert.deepEqual(valid('ISC'), validSPDXExpression);
```
The function will return a warning and suggestion for nearly-correct license identifiers:
```
```javascript
assert.deepEqual(
valid('Apache 2.0'),
{
validForOldPackages: false,
validForNewPackages: false,
warnings: [
'license should be ' +
'a valid SPDX license expression (without "LicenseRef"), ' +
'"UNLICENSED", or ' +
""SEE LICENSE IN <filename>"'.
'license is similar to the valid expression "Apache-2.0"'
]
}
)
```

SPDX expressions are valid, too ...

```javascript
// Simple SPDX license expression for dual licensing
assert.deepEqual(
valid('(GPL-3.0-only OR BSD-2-Clause)'),
validSPDXExpression
);
```

... except if they contain `LicenseRef`:

```javascript
var warningAboutLicenseRef = {
validForOldPackages: false,
validForNewPackages: false,
spdx: true,
warnings: [
'license should be ' +
```
'a valid SPDX license expression (without "LicenseRef"), ' +
""UNLICENSED", or ' +
""SEE LICENSE IN <filename>"',
]
);

assert.deepEqual(
 valid('LicenseRef-Made-Up'),
 warningAboutLicenseRef
);

assert.deepEqual(
 valid('(MIT OR LicenseRef-Made-Up)'),
 warningAboutLicenseRef
);
```

If you can't describe your licensing terms with standardized SPDX identifiers, put the terms in a file in the package and point users there:

```
```javascript
assert.deepEqual(
 valid('SEE LICENSE IN LICENSE.txt'),
 {  
  validForNewPackages: true,
  validForOldPackages: true,
  inFile: 'LICENSE.txt'
 }
);

assert.deepEqual(
 valid('SEE LICENSE IN license.md'),
 {  
  validForNewPackages: true,
  validForOldPackages: true,
  inFile: 'license.md'
 }
);
```

If there aren't any licensing terms, use `UNLICENSED`:

```
```javascript
var unlicensed = {  
  validForNewPackages: true,
  validForOldPackages: true,
  unlicensed: true
};
```
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
```
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1064 browserslist 3.2.8

1.1064.1 Available under license :
The MIT License (MIT)

Copyright 2014 Andrey Sitnik <andrey@sitnik.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1065 netty-transport-native-epoll

4.1.45.Final

1.1065.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
 * Set the {@code TCP_MD5SIG} option on the socket. See {@code linux/tcp.h} for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
 */
/**
 * Set the {@code TCP_QUICKACK} option on the socket. See <a href="http://linux.die.net/man/7/tcp">TCP_QUICKACK</a>
 * for more details.
 */

No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE
~
~ 2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
jar/io/netty/channel/epoll/EpollSocketChannelConfig.java
No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE
~
~ 2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
jar/META-INF/maven/io.netty/netty-transport-native-epoll/pom.xml
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

/**
 * Set the [TCP_MD5SIG] option on the socket. See [linux/tcp.h] for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
 */

Found in path(s):
*/
/opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4.1.45-final-sources.jar/io/netty/channel/epoll/EpollServerSocketChannelConfig.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
*/
/opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4.1.45-final-sources.jar/io/netty/channel/epoll/EpollEventArray.java

Open Source Used In DNAC Platform Antelope 3525
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-jar/io/netty/channel/epoll/EpollDatagramChannelConfig.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/AbstractEpollChannel.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollDatagramChannel.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/NativeDatagramPacketArray.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollEventLoop.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollServerSocketChannel.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/package-info.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollTcpInfo.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollEventLoop.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollServerSocketChannel.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollSocketChannel.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/Epoll.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4-1-45-final-sources-
  jar/io/netty/channel/epoll/EpollChannelOption.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4.1.45-final-sources-
  jar/io/netty/channel/epoll/Native.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4.1.45-final-sources-
  jar/netty_epoll_native.c
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4.1.45-final-sources-
  jar/io/netty/channel/epoll/LinuxSocket.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4.1.45-final-sources-
  jar/io/netty/channel/epoll/NativeStaticallyReferencedJniMethods.java
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4.1.45-final-sources-
  jar/netty_epoll_linuxsocket.c
* /opt/ws_local/PERMITS_SQL/1074832144_1595959525.4/0/netty-transport-native-epoll-4.1.45-final-sources-
  jar/netty_epoll_linuxsocket.h

1.1066 diff2html 2.5.0
1.1066.1 Available under license:

Copyright 2014-2016 Rodrigo Fernandes https://rtfpessoa.github.io/

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1067 yaml 1.10.2 1.10.2
1.1067.1 Available under license:
Copyright 2018 Eemeli Aro <eemeli@gmail.com>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1068 multer 1.3.1 1.3.1
1.1068.1 Available under license:
Copyright (c) 2014 Hage Yaapa <http://www.hacksparrow.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1069 ecc-jsbn 0.1.2

1.1069.1 Available under license:

Licensing
--------

This software is covered under the following copyright:

/*
 * Copyright (c) 2003-2005 Tom Wu
 * All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY
 * WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
 *
 * IN NO EVENT SHALL TOM WU BE LIABLE FOR ANY SPECIAL, INCIDENTAL,
 * INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER
 * RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF
 * THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT
 * OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * In addition, the following condition applies:
 *
 * All redistributions must retain an intact copy of this copyright notice
 * and disclaimer.
Address all questions regarding this license to:

Tom Wu
tjw@cs.Stanford.EDU
The MIT License (MIT)

Copyright (c) 2014 Jeremie Miller

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1070 jetty-util 9.4.41.v20210516
1.1070.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:
a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and 
b) in the case of each subsequent Contributor: 
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other
entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of the
remainder of the terms of this Agreement, and without further action by the
parties hereto, such provision shall be reformed to the minimum extent
necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Program itself
(excluding combinations of the Program with other software or hardware)
infringes such Recipient's patent(s), then such Recipient's rights granted
under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to
comply with any of the material terms or conditions of this Agreement and does
not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate,
Recipient agrees to cease use and distribution of the Program as soon as
reasonably practicable. However, Recipient's obligations under this Agreement
and any licenses granted by Recipient relating to the Program shall continue
and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
order to avoid inconsistency the Agreement is copyrighted and may only be
modified in the following manner. The Agreement Steward reserves the right to
publish new versions (including revisions) of this Agreement from time to
time. No one other than the Agreement Steward has the right to modify this
Agreement. The Eclipse Foundation is the initial Agreement Steward. The
Eclipse Foundation may assign the responsibility to serve as the Agreement
Steward to a suitable separate entity. Each new version of the Agreement will
be given a distinguishing version number. The Program (including
Contributions) may always be distributed subject to the version of the
Agreement under which it was received. In addition, after a new version of the
Agreement is published, Contributor may elect to distribute the Program
(including its Contributions) under the new version. Except as expressly
stated in Sections 2(a) and 2(b) above, Recipient receives no rights or
licenses to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in the
Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

=====================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception. https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions. http://openjdk.java.net/legal/gplv2+ce.html

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay
The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1071 slf4j 1.7.28

1.1072 iеноopen 1.0.0 1.0.0
1.1072.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014, 2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1073 guava 28.2-jre
1.1073.1 Available under license:
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark,
and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute
must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices
that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text
file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the
Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices
normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the
License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an
addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license
terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works
as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions
stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for
inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any
additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of
any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product
names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and
reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work
(and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR
PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and
assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or
otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing,
shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or
consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of
such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may
choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1074 babel-plugin-transform-async-to-generator 6.24.1
1.1074.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1075 log4j-slf4j-impl 2.14.0

1.1075.1 Available under license:

Apache Log4j SLF4J Binding
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1076 validator 10.11.0 10.11.0

1.1076.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1077 lodash.isnil 4.0.0

1.1077.1 Available under license:

Copyright 2012-2016 The Dojo Foundation <http://dojofoundation.org/>
Based on Underscore.js, copyright 2009-2016 Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1078 imurmurhash 0.1.4

1.1078.1 Available under license:

No license file was found, but licenses were detected in source scan.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
subject to the following conditions:
The above copyright notice and this permission notice shall be included in all
1.1079 merge2 1.3.0

1.1079.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2019 Teambition

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1080 make-iterator 1.0.1

1.1080.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1081 jnr-posix 3.0.27
1.1081.1 Available under license :

Found license 'GNU Lesser General Public License' in 'License Version 1.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of Software distributed under the License is distributed on an "AS Alternatively, the contents of this file may be used under the terms of either of the GNU General Public License Version 2 or later (the "GPL"), or the GNU Lesser General Public License Version 2.1 or later (the "LGPL"), use your version of this file under the terms of the CPL, indicate your'

1.1082 bootstrap 3.4.1 3.4.1
1.1082.1 Available under license :

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions
a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography;
a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a
modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You
must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor’s copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the
applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICensor BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in
compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.

The MIT License (MIT)

Copyright (c) 2011-2019 Twitter, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1083 property-expr 2.0.4 2.0.4

1.1083.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jason Quense

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1084 unicode-trie 0.3.1
1.1084.1 Available under license :
Copyright 2018

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1085 levn 0.3.0 0.3.0
1.1085.1 Available under license :
Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

1.1086 bytes 3.0.0
1.1086.1 Available under license :
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Jed Watson <jed.watson@me.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1087 responselike 1.0.2
1.1087.1 Available under license :
Copyright (c) 2017 Luke Childs

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1088 lcid 2.0.0

1.1088.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1089 path-exists 4.0.0 4.0.0

1.1089.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1090 avatica-metrics 1.17.0
1.1090.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Apache Calcite -- Avatica
Copyright 2012-2020 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.1091 socket.io-client 1.7.4
1.1091.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Guillermo Rauch

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1092 jkeylockmanager 2.1.0
1.1092.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "["]
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1093 hdrhistogram 2.1.12
1.1093.1 Available under license:

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1575980548657
Build-Jdk: 1.8.0_232
Built-By: gil
Bundle-Description: HdrHistogram supports the recording and analyzing sampled data value counts across a configurable integer value range with configurable value precision within the range. Value precision is expressed as the number of significant digits in the value recording, and provides control over value quantization behavior across the value range and the subsequent value resolution at any given level.
Bundle-License: http://creativecommons.org/publicdomain/zero/1.0/, https://opensource.org/licenses/BSD-2-Clause
Bundle-ManifestVersion: 2
Bundle-Name: HdrHistogram
Bundle-SymbolicName: org.hdrhistogram.HdrHistogram
Bundle-Version: 2.1.12
Created-By: Apache Maven Bundle Plugin
Export-Package: org.HdrHistogram;version="2.1.12",org.HdrHistogram.packagedarray;version="2.1.12"
Implementation-Title: HdrHistogram
Implementation-Vendor-Id: org.hdrhistogram
Implementation-Version: 2.1.12
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.7))"
Specification-Title: HdrHistogram
Specification-Version: 2.1.12
Tool: Bnd-2.3.0.201405100607

Found in path(s):
* /opt/cola/permits/1183539882_1627427044.77/0/hdrhistogram-2-1-12-2-jar/META-INF/MANIFEST.MF
No license file was found, but licenses were detected in source scan.

* public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
  <name>Public Domain, per Creative Commons CC0</name>

Found in path(s):
* /opt/cola/permits/1183539882_1627427044.77/0/hdrhistogram-2-1-12-2-jar/META-INF/maven/org.hdrhistogram/HdrHistogram/pom.xml

1.1094 babel-plugin-proposal-nullish-coalescing-operator 7.13.8
1.1094.1 Available under license :

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1095 npm-run-path 2.0.2
1.1095.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1096 drill-storage-hbase 1.19.0
1.1096.1 Available under license:

Drill : Contrib : Storage : HBase
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

Open Source Used In DNAC Platform Antelope 3579
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1097 vary 1.0.1
1.1097.1 Available under license :
(The MIT License)

Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
1.1098 javaparser-core 3.18.0

1.1098.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.
You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.1099 v8 8.8.8.8
1.1099.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# The 3-Clause BSD License

Copyright 2019 web-platform-tests contributors

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The JsonCpp library's source code, including accompanying documentation, tests and demonstration applications, are licensed under the following conditions...

The author (Baptiste Lepilleur) explicitly disclaims copyright in all jurisdictions which recognize such a disclaimer. In such jurisdictions, this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of 2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur, and is released under the terms of the MIT License (see below).
In jurisdictions which recognize Public Domain property, the user of this software may choose to accept it either as 1) Public Domain, 2) under the conditions of the MIT License (see below), or 3) under the terms of dual Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is described in clear, concise terms at:

http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

========================================================================
Copyright (c) 2007-2010 Baptiste Lepilleur

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
========================================================================
(END LICENSE TEXT)

The MIT license is compatible with both the GPL and commercial software, affording one all of the rights of Public Domain with the minor nuisance of being required to keep the above copyright notice and license text in the source code. Note also that by accepting the Public Domain "license" you can re-license your copy using whatever license you like.

All files in this directory are provided by the following license if not stated otherwise in the individual file:

=================================
This file is provided under a dual BSD/GPLv2 license. When using or redistributing this file, you may do so under either license.
GPL LICENSE SUMMARY

Copyright (c) 2005-2012 Intel Corporation. All rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St - Fifth Floor, Boston, MA 02110-1301 USA. The full GNU General Public License is included in this distribution in the file called LICENSE/GPL.

Contact Information:

BSD LICENSE

Copyright (c) 2005-2012 Intel Corporation. All rights reserved.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This license applies to all parts of V8 that are not externally maintained libraries. The externally maintained libraries used by V8 are:

- PCRE test suite, located in test/mjsunit/third_party/regexp-pcre/regexp-pcre.js. This is based on the test suite from PCRE-7.3, which is copyrighted by the University of Cambridge and Google, Inc. The copyright notice and license are embedded in regexp-pcre.js.

- Layout tests, located in test/mjsunit/third_party/object-keys. These are based on layout tests from webkit.org which are copyrighted by Apple Computer, Inc. and released under a 3-clause BSD license.

- Strongtalk assembler, the basis of the files assembler-arm-inl.h, assembler-arm.cc, assembler-arm.h, assembler-ia32-inl.h, assembler-ia32.cc, assembler-ia32.h, assembler-x64-inl.h, assembler-x64.cc, assembler-x64.h, assembler-mips-inl.h, assembler-mips.cc, assembler-mips.h, assembler.cc and assembler.h. This code is copyrighted by Sun Microsystems Inc. and released under a 3-clause BSD license.

- Valgrind client API header, located at src/third_party/valgrind/valgrind.h. This is released under the BSD license.

- The Wasm C/C++ API headers, located at third_party/wasm-api/wasm.{h,hh}. This is released under the Apache license. The API's upstream prototype implementation also formed the basis of V8's implementation in src/wasm/c-api.cc.

These libraries have their own licenses; we recommend you read them, as their terms may differ from the terms below.

Further license information can be found in LICENSE files located in sub-directories.

Copyright 2014, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
with the distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 1994-2006 Sun Microsystems Inc.
All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

- Redistribution in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

- Neither the name of Sun Microsystems or the names of contributors may
  be used to endorse or promote products derived from this software without
  specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Tests included here are based on build output generated by the six-speed
benchmark suite.

Copyright (c) 2015 Kevin Decker

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

CC0 1.0 Universal

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer
exclusive Copyright and Related Rights (defined below) upon the creator and
subsequent owner(s) (each and all, an "owner") of an original work of
authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the
purpose of contributing to a commons of creative, cultural and scientific
works ("Commons") that the public can reliably and without fear of later
claims of infringement build upon, modify, incorporate in other works, reuse
and redistribute as freely as possible in any form whatsoever and for any
purposes, including without limitation commercial purposes. These owners may
contribute to the Commons to promote the ideal of a free culture and the
further production of creative, cultural and scientific works, or to gain
reputation or greater distribution for their Work in part through the use and
efforts of others.

For these and/or other purposes and motivations, and without any expectation
of additional consideration or compensation, the person associating CC0 with a
Work (the "Affirmer"), to the extent that he or she is an owner of Copyright
and Related Rights in the Work, voluntarily elects to apply CC0 to the Work
and publicly distribute the Work under its terms, with knowledge of his or her
Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;

ii. moral rights retained by the original author(s) and/or performer(s);

iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;

iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;

v. rights protecting the extraction, dissemination, use and reuse of data in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver
shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.

b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.

c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.

d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

For more information, please see <http://creativecommons.org/publicdomain/zero/1.0/>

Notice that the following BSD-style license applies to this one file (valgrind.h) only. The rest of Valgrind is licensed under the terms of the GNU General Public License, version 2, unless
otherwise indicated. See the COPYING file in the source
distribution for details.

This file is part of Valgrind, a dynamic binary instrumentation
framework.

Copyright (C) 2000-2010 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must
   not claim that you wrote the original software. If you use this
   software in a product, an acknowledgment in the product
   documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must
   not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior written
   permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Notice that the above BSD-style license applies to this one file
(valgrind.h) only. The entire rest of Valgrind is licensed under
the terms of the GNU General Public License, version 2. See the
COPYING file in the source distribution for details.
// Copyright 2016 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PCRE LICENCE
------------

PCRE is a library of functions to support regular expressions whose syntax
and semantics are as close as possible to those of the Perl 5 language.

Release 7 of PCRE is distributed under the terms of the "BSD" licence, as
specified below. The documentation for PCRE, supplied in the "doc"
directory, is distributed under the same terms as the software itself.

The basic library functions are written in C and are freestanding. Also
included in the distribution is a set of C++ wrapper functions.

THE BASIC LIBRARY FUNCTIONS
-------------------------------

Written by: Philip Hazel
Email local part: ph10
Email domain: cam.ac.uk
University of Cambridge Computing Service,

Copyright (c) 1997-2007 University of Cambridge
All rights reserved.

THE C++ WRAPPER FUNCTIONS
-------------------------

Contributed by: Google Inc.

Copyright (c) 2007, Google Inc.
All rights reserved.

THE "BSD" LICENCE
---------------

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the name of Google
  Inc. nor the names of their contributors may be used to endorse or
  promote products derived from this software without specific prior
  written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

End
Copyright (c) 2006 Apple Computer, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder(s) nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2010 Jonathan Hartley
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holders, nor those of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property.

Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

<table>
<thead>
<tr>
<th>Release</th>
<th>Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td></td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>
Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.


3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT
INFRINGEMENT ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON
FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any
relationship of agency, partnership, or joint venture between PSF and
Licensee. This License Agreement does not grant permission to use PSF
trademarks or trade name in a trademark sense to endorse or promote
products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0
---------------------------------------------

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an
office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the
Individual or Organization ("Licensee") accessing and otherwise using
this software in source or binary form and its associated
documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License
Agreement, BeOpen hereby grants Licensee a non-exclusive,
royalty-free, world-wide license to reproduce, analyze, test, perform
and/or display publicly, prepare derivative works, distribute, and
otherwise use the Software alone or in any derivative version,
provided, however, that the BeOpen Python License is retained in the
Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS"
basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT
4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1
------------------------------------------

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following..."
3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT
CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Developed at SunSoft, a Sun Microsystems, Inc. business. Permission to use, copy, modify, and distribute this software is freely granted, provided that this notice is preserved.

Copyright 2006-2011, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
1.1100 de.huxhorn.lilith.data.eventsource
0.9.44

1.1100.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the
Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

/*
 * Lilith - a log event viewer.
 * Copyright (C) 2007-2011 Joern Huxhorn
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Lesser General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Lesser General Public License for more details.
 */
* You should have received a copy of the GNU Lesser General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

/*
* Copyright 2007-2011 Joern Huxhorn
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

1.1101 zip-stream 1.2.0

1.1101.1 Available under license :

Copyright (c) 2014 Chris Talkington, contributors.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
1.1102 nock 11.9.1

1.1102.1 Available under license:

MIT License

Copyright (c) 2011-2019 Pedro Teixeira and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Contributors

This list has been generated to thank everyone here for their contribution. Nock wouldn't be what it is without the individuals who've given their time to it. In alphabetical order, here is a list of people who the maintainers think are worthy of being thanked publicly and permanently here.

Aalhad Saraf @saraf https://github.com/saraf
Aaron D Borden @adborden https://github.com/adborden
Aaron Greenwald @aarongreenwald https://github.com/aarongreenwald
Abdul Alkhatib @abdulito https://github.com/abdulito
Abdullah Ali @AbdullahAli https://github.com/AbdullahAli
Abhijeet Singh @abhijeetps https://github.com/abhijeetps
Abimbola Idowu @hisabimbola https://github.com/hisabimbola
Adam @snowmac https://github.com/snowmac
Adam Bretz @arb https://github.com/arb
Adam Cofer @acofer https://github.com/acofer
Adam Ever-Hadani @adamhadani https://github.com/adamhadani
Adam LaFave @lafave https://github.com/lafave
Adam Tyler @AdamTyler https://github.com/AdamTyler
Adam Voss @adamvoss https://github.com/adamvoss
Adara @adarrra https://github.com/adarrra
Aditya Bhardwaj @purezen https://github.com/purezen
Aditya Gupta @aditya1315 https://github.com/aditya1315
Adrian Mejia @amejiarosario https://github.com/amejiarosario
Adrian Perez @blackxored https://github.com/blackxored
Aidan Feldman @afeld https://github.com/afeld
Akshaya Kumar Sharma @lordakshaya https://github.com/lordakshaya
Alan Clarke @alanclarke https://github.com/alanclarke
Alan Lee @mralanlee https://github.com/mralanlee
Alastair Brunton @pyrat https://github.com/pyrat
Alberto Gasparin @albertogasparin https://github.com/albertogasparin
Alberto Pose @pose https://github.com/pose
Aleksander Barszczewski @alekbarszczewski https://github.com/alekbarszczewski
Aleksandr Gidenko @aleksgidenko https://github.com/aleksgidenko
Aleksi Pekkala @epiphone https://github.com/epiphone
Alex Banks @MoonTahoe https://github.com/MoonTahoe
Alex Dunn @dunn https://github.com/dunn
Alex Gilleran @AlexGilleran https://github.com/AlexGilleran
Alex Layton @awlayton https://github.com/awlayton
Alex Olson @alexkolson https://github.com/alexkolson
Alex Panov @alexpanov https://github.com/alexpanov
Alex Pavlov @macscripter https://github.com/macscripter
Alex Voitau @voitau https://github.com/voitau
Alex Zylman @azyylman https://github.com/azyylman
Alexander Christiansson @alexanderchr https://github.com/alexanderchr
Alexander Rusakov @arusakov https://github.com/arusakov
Alexander Simmerl @xla https://github.com/xla
Alexander Tesfamichael @alextes https://github.com/alextes
Alexandru Vlduu @alessioalex https://github.com/alessioalex
Alexey @akaguny https://github.com/akaguny
Alexey @i4got10 https://github.com/i4got10
Alexis Metaireau @almet https://github.com/almet
Ali Abu Ras @aliaburas80 https://github.com/aliaburas80
Ali Chen @ali322 https://github.com/ali322
Alison Kohl @alisonkohl https://github.com/alisonkohl
Alix Bhagooa @alixe https://github.com/alixe
Allan Stewart @allan-stewart https://github.com/allan-stewart
Allen Luce @allenluce https://github.com/allenluce
Am44 @ami44 https://github.com/ami44
Amru E @amrue https://github.com/amrue
Anders D. Johnson @AndersDJohnson https://github.com/AndersDJohnson
Anders S. Ootuka @anderson-otuka-dextra https://github.com/anderson-otuka-dextra
Andras @andrasq https://github.com/andrasq
Andrea Gariboldi @aaaristo https://github.com/aaaristo
Andrea Passaglia @gurghet https://github.com/gurghet
Andrea Reginato @andreareginato https://github.com/andreareginato
Andreas Plisson @apals https://github.com/apals
Andrei Rusu @beatfactor https://github.com/beatfactor
Andres Gariglio @andresgariglio https://github.com/andresgariglio
Andrew Chilton @chilts https://github.com/chilts
Andrew Krespanis @andrewk https://github.com/andrewk
Andrew Luetergs @andrewluetergs https://github.com/andrewluetergs
Andrew Luhring @andrew-luhring https://github.com/andrew-luhring
Andrew Matteson @aromatt https://github.com/aromatt
Andrew T. Poe @andrewtpoe https://github.com/andrewtpoe
Andri MII @moll https://github.com/moll
Andr Cruz @satazor https://github.com/satazor
Andr Gaul @andrenarchy https://github.com/andrenarchy
Andr Rodrigues @andrerod https://github.com/andrerod
Andy Fowler @andyfowler https://github.com/andyfowler
Andy Kramolisch @andykram https://github.com/andykram
Ankur Oberoi @aomkarlo https://github.com/aomkarlo
Ankush Sharma @darxtrix https://github.com/darxtrix
Anny He @annyh https://github.com/annyh
Anshuman Verma @anshumanv https://github.com/anshumanv
Anthony BARRE @abarre https://github.com/abarre
Anthony Hildoor @ahildoor https://github.com/ahildoor
Antoine Le Taxin @ModuloM https://github.com/ModuloM
Antoine R. Dumont @ardumont https://github.com/ardumont
Antoine Rey @antoinerey https://github.com/antoinerey
Anton @z-vr https://github.com/z-vr
Anton Samper Rivaya @antonsampler https://github.com/antonsampler
Aria Stewart @aredridel https://github.com/aredridel
Arjan Frans @arjanfrans https://github.com/arjanfrans
Arnaud Gonzlez @arnaugm https://github.com/arnaugm
Aron Woost @aronwoost https://github.com/aronwoost
Artem Zakarchenko @blackrabbit99 https://github.com/blackrabbit99
Artur Mtkrtyan @arturmkrtyan https://github.com/arturmkrtyan
Asaf Mesika @asafm https://github.com/asafm
Ashik Vetrivelu @ashik94vc https://github.com/ashik94vc
Assaf Arkin @assaf https://github.com/assaf
Atishay Jain @atishay https://github.com/atishay
Atila Incze @atimb https://github.com/atimb
Aurlien Thieriot @athieriot https://github.com/athieriot
Austin Birch @austinbirch https://github.com/austinbirch
Austin Mao @austinmao https://github.com/austinmao
Avi (Ananya) Das @avidas https://github.com/avidas
Avi Deitcher @deitch https://github.com/deitch
Aviad Moreshet @CodeJiang https://github.com/CodeJiang
Balazs Nagy @balnagy https://github.com/balnagy
Bartek Szopka @bartaz https://github.com/bartaz
Bay @yamikuronue https://github.com/yamikuronue
Beilan Wang @beilanwang https://github.com/beilanwang
Ben Buckman @benbuckman https://github.com/benbuckman
Ben Feigin @bfeigin https://github.com/bfeigin
Ben Gilman @BenMGilman https://github.com/BenMGilman
Ben Stokoe @benstokoe https://github.com/benstokoe
Benjamin E. Coe @bcoe https://github.com/bcoe

Open Source Used In DNAC Platform Antelope 3616
Open Source Used In DNAC Platform Antelope 3617

Benjamin Urban @benurb https://github.com/benurb
Benny Neugebauer @bennyn https://github.com/bennyn
Bharath Raja @bigomega https://github.com/bigomega
Bin Chang @BinChang https://github.com/BinChang
Boris Serebrov @serebrov https://github.com/serebrov
Boubker Boulahdid @bboulahdid https://github.com/bboulahdid
Bradley Bridges @bbridges https://github.com/bbridges
Brandon Keepers @bkeepers https://github.com/bkeepers
Branko Zivanovic @zivanovicb1 https://github.com/zivanovicb1
Brent Van Minnen @Bjvanminnen https://github.com/Bjvanminnen
Brett Porter @brettporter https://github.com/brettporter
Brian Beck @exogen https://github.com/exogen
Brian Dentino @bdentino https://github.com/bdentino
Brian Hann @c0bra https://github.com/c0bra
Brian J Brennan @brianloveswords https://github.com/brianloveswords
Brian M. Jemilo II @JemiloII https://github.com/JemiloII
Brian Mann @brian-mann https://github.com/brian-mann
Brian Mortenson @bgmort https://github.com/bgmort
Bruno Quaresma @BrunoQuaresma https://github.com/BrunoQuaresma
Bryan Joseph @bj97301 https://github.com/bj97301
Byron Bai @happybai https://github.com/happybai
C. T. Lin @chentsulin https://github.com/chentsulin
Cache Hamm @CacheControl https://github.com/CacheControl
Cacie Prins @cacieprins https://github.com/cacieprins
Cade Ward @cademward https://github.com/cademward
Cameron Lakenen @lakenen https://github.com/lakenen
Cameron Roe @cameronroe https://github.com/cameronroe
Cameron Sutter @cameronsutter https://github.com/cameronsutter
Carl Ansley @carlansley https://github.com/carlansley
Carl Napier-Cook @carlnx https://github.com/carlnx
Carlos Guerrero @guerrerocarlos https://github.com/guerrerocarlos
Carlos Villuendas Zambrana @carlosvillu https://github.com/carlosvillu
Celestino Ferreira Gomes @tinogomes https://github.com/tinogomes
Charles Lindsay @chazomaticus https://github.com/chazomaticus
Charles Phillips @doublerebel https://github.com/doublerebel
Charlie Briggs @Limess https://github.com/Limess
Charlie Robbins @indexzero https://github.com/indexzero
Charlie Rudolph @charlierudolph https://github.com/charlierudolph
Chase Wilson @jiggliemon https://github.com/jiggliemon
Chevdor @chevdor https://github.com/chevdor
Chris Cuellar @ChrisC https://github.com/ChrisC
Chris Jamieson @jamiesoncj https://github.com/jamiesoncj
Chris Keen @zedd45 https://github.com/zedd45
Chris Salvato @csalvato https://github.com/csalvato
Chris Thompson @cjthompson https://github.com/cjthompson
Chris Ward @cw6365 https://github.com/cw6365
Christoph Neuroth @c089 https://github.com/c089
Christoph Wagner @PandaWhisperer https://github.com/PandaWhisperer
Christopher Stone @ctstone https://github.com/ctstone
Claudio Petrini @claudiopetrini https://github.com/claudiopetrini
Claudius Coenen @ccoenen https://github.com/ccoenen
Cody A. Taylor @CodeMan99 https://github.com/CodeMan99
Cody Greene @cody-greene https://github.com/cody-greene
Cole Turner @coleturner https://github.com/coleturner
Cory Reed @swashcap https://github.com/swashcap
Cosmo Wolfe @cozmo https://github.com/cozmo
Coveralls @coveralls https://github.com/coveralls
Craig Freeman @CFreeAtEbsco https://github.com/CFreeAtEbsco
Craig Morris @morrislaptop https://github.com/morrislaptop
Cristian Cialli @ciccia https://github.com/ciccia
Cristian Szwarc @cristianszwarc https://github.com/cristianszwarc
Csaba Palfi @csabapalfi https://github.com/csatapalfi
Cyril Auburtin @caub https://github.com/caub
DONGKEON KIM @shallwefootball https://github.com/shallwefootball
Dallon Feldner @dallonf https://github.com/dallonf
Damian Janowski @djanowski https://github.com/djanowski
Damian Kaczmarek @Rush https://github.com/Rush
Damian Krzeminski @pirxpilot https://github.com/pirxpilot
Dan @danpantry https://github.com/danpantry
Dan Jenkins @danjenkins https://github.com/danjenkins
Dan Ristea @danrr https://github.com/danrr
Dan Riti @danti https://github.com/danti
Daniel Bretoi @danielb2 https://github.com/danielb2
Daniel Fenton @dmfenton https://github.com/dmfenton
Daniel Pupius @dpup https://github.com/dpup
Daniel Upton @elkelk https://github.com/elkelk
Daniil Zavyalov @daniilzav123 https://github.com/daniilzav123
Danil Ilouz @danilouz https://github.com/danilouz
Dank @dannyid https://github.com/dannyid
Danny Andrews @danny-andrews https://github.com/danny-andrews
Dario Spadoni @dariospadoni https://github.com/dariospadoni
Dave Cadwallader @geekdave https://github.com/geekdave
Dave Jensen @djensen47 https://github.com/djensen47
Dave Meehan @dmeelan1968 https://github.com/dmeelan1968
Dave Schinkel @dscinkel https://github.com/dscinkel
David Bjorklund @kesla https://github.com/kesla
David Braun @NodeGuy https://github.com/NodeGuy
David Chambers @davidchambers https://github.com/davidchambers
David Chase @davidchase https://github.com/davidchase
David Cho-Lerat @david-cho-lerat-HL2 https://github.com/david-cho-lerat-HL2
David Judik @judikdavid https://github.com/judikdavid
David M. Lee @leedm777 https://github.com/leedm777
David Pate @DavidTPate https://github.com/DavidTPate
David Rissato Cruz @davidrissato https://github.com/davidrissato
David Rousselie @dax https://github.com/dax
David Schoen @neerolyte https://github.com/neerolyte
Joaquim Serafim @joaquimserafim https://github.com/joaquimserafim
Joe Becher @drazisil https://github.com/drazisil
Joe Kent @itsjoekent https://github.com/itsjoekent
Joe Norman @joseph-norman https://github.com/joseph-norman
Joe Smith @Yasumoto https://github.com/Yasumoto
Johan Borestad @borestad https://github.com/borestad
Johan Nordberg @jnordberg https://github.com/jnordberg
Johannes Jrg Schmidt @jo https://github.com/jo
Johannes Zellner @nebulade https://github.com/nebulade
John @jhnlsn https://github.com/jhnlsn
John Duhamel @jduhamel https://github.com/jduhamel
John Dzak @jdzak https://github.com/jdzak
John Mathis @JohnDMathis https://github.com/JohnDMathis
John T @digitaltsai https://github.com/digitaltsai
John-David Dalton @jdalton https://github.com/jdalton
Jon B @PizzaAficionado https://github.com/PizzaAficionado
Jon Jaques @jonjaques https://github.com/jonjaques
Jon Skulski @jskulski https://github.com/jskulski
Jonas Lilja @jlilja https://github.com/jlilja
Jonathan Bergknoff @jbergknoff https://github.com/jbergknoff
Jonathan Glock @glockjt https://github.com/glockjt
Jonathan Lau @jonahlau https://github.com/jonahlau
Jonathan Otto @jotto https://github.com/jotto
Jonathan Petitcolas @jpetitcolas https://github.com/jpetitcolas
Jonathan Rainey @Tivoli https://github.com/Tivoli
Jonathon Herbert @jonathonherbert https://github.com/jonathonherbert
Joni Salmi @josalmi https://github.com/josalmi
Jordan Harband @ljharb https://github.com/ljharb
Joseph Callaars @bcallaars https://github.com/bcallaars
Josh Crowther @jshcrowthe https://github.com/jshcrowthe
Josh Marchello @jmarchello https://github.com/jmarchello
Josh McMillan @mcmillan https://github.com/mcmillan
Joshua Acheson @joshacheson https://github.com/joshacheson
Joshua Holbrook @jfhbrook https://github.com/jfhbrook
Josip Delic @delijati https://github.com/delijati
Jos F. Romaniello @jromaniello https://github.com/jromaniello
Joo Ferreira @jmnsf https://github.com/jmnsf
Juan Cruz Viotti @jviotti https://github.com/jviotti
Julian Hille @julianhille https://github.com/julianhille
Juraci de Lima Vieira Neto @Juraci https://github.com/Juraci
Justin @justincy https://github.com/justincy
Justin Helmer @justinhelmer https://github.com/justinhelmer
Justin Marrington @HowlingEverett https://github.com/HowlingEverett
Jrme Avoustin @rephia https://github.com/rephia
Jrg Henning @joerx https://github.com/joerx
Kafo @hyalkaf https://github.com/hyalkaf
Kai Schwarz @papakai https://github.com/papakai
Matteo Mazzarolo @mmazzarolo https://github.com/mmazzarolo
Matthew Mirande @busticated https://github.com/busticated
Matthew Oaxaca @moaxaca https://github.com/moaxaca
Matthew Robben @mfrobben https://github.com/mfrobben
Matthew Turney @pho3nixf1re https://github.com/pho3nixf1re
Mattia Asti @mtt87 https://github.com/mtt87
Max Brosnahan @gingermusketeer https://github.com/gingermusketeer
Max Greenblatt @maxGreenblatt https://github.com/maxGreenblatt
Mayank Badola @mbad0la https://github.com/mbad0la
Mayur Awaghade @mayurva https://github.com/mayurva
Michael Diep @MADiep https://github.com/MADiep
Michael Ford @mtford90 https://github.com/mtford90
Michael Graf @MaerF0x0 https://github.com/MaerF0x0
Michael J. Abraham @mjabraham47 https://github.com/mjabraham47
Michael K @0815fox https://github.com/0815fox
Michael Lippens @mlippens https://github.com/mlippens
Michael Nisi @michaelnisi https://github.com/michaelnisi
Michael Pardue @mpardue https://github.com/mpardue
Michael Pratt @Urthen https://github.com/Urthen
Michael Seidel @mjsiedel https://github.com/mjsiedel
Michael Vogt @neophob https://github.com/neophob
Michael Walker @cmswalker https://github.com/cmswalker
Michael Weibel @mweibel https://github.com/mweibel
Micha Wadas @Ginden https://github.com/Ginden
Michele Spina @mikspi https://github.com/mikspi
Michiel Helvensiveijn @mhelvens https://github.com/mhelvens
Mickal Tricot @mickaeltr https://github.com/mickaeltr
Miguel Espinoza @purefan https://github.com/purefan
Miguel Isidoro @misidoro https://github.com/misidoro
Miha Rebernik @mihar https://github.com/mihar
Mikael Arneborn @marneborn https://github.com/marneborn
Mike Chen @mhchen https://github.com/mhchen
Mike Emery @MikeEmery https://github.com/MikeEmery
Mike James @export-mike https://github.com/export-mike
Mike Reinstein @mreinstein https://github.com/mreinstein
Mikela @glassresistor https://github.com/glassresistor
Mikey Powers @mpowers https://github.com/mpowers
Mikko Koponen @mtkopone https://github.com/mtkopone
Minh Son Nguyen @nguymin4 https://github.com/nguymin4
Mohammad Salim @MSalim https://github.com/MSalim
MooYeol Prescott Lee @mooyoul https://github.com/mooyoul
Naga Pavan Kumar Kunisetty @nagapavan https://github.com/nagapavan
Nate Wang @supnate https://github.com/supnate
Nathan @naw https://github.com/naw
Nathan Friedly @nfriedly https://github.com/nfriedly
Nathan Houle @ndhoule https://github.com/ndhoule
Nathan Richards @LAD500 https://github.com/LAD500
Neeraj @smurfpandey https://github.com/smurfpandey
Nic Haynes @nicinabox https://github.com/nicinabox
Nicholas Calugar @SocalNick https://github.com/SocalNick
Nick Howes @halfninja https://github.com/halfninja
Nick Stefan @NickStefan https://github.com/NickStefan
Nicky Jay @niksajanjic https://github.com/niksajanjic
Nico Schlmer @nschloe https://github.com/nschloe
Nicolas Del Valle @ndelvalle https://github.com/ndelvalle
Nicolas Froidure @nfroidure https://github.com/nfroidure
Nicolas Lacasse @nlacasse https://github.com/nlacasse
Nicols Bevacqua @bevacqua https://github.com/bevacqua
Niels Krijger @nielskrijger https://github.com/nielskrijger
Nikita Vasilevskiy @Nigrimmist https://github.com/Nigrimmist
Nikolaus Piccolotto @prayerslayer https://github.com/prayerslayer
Nils Buenger @nilsbunger https://github.com/nilsbunger
Nils Knappmeier @nknap https://github.com/nknap
Nock Bot @nockbot https://github.com/nockbot
Nordes Mnard-Lamarre @Nordes https://github.com/Nordes
Nuno Job @dscape https://github.com/dscape
Nuno Sousa @nunofgs https://github.com/nunofgs
Oleg @omakoleg https://github.com/omakoleg
Oliver Wong @owiber https://github.com/owiber
Olivier Louvignes @mgcrea https://github.com/mgcrea
Olivier Tassinari @oliviertassinari https://github.com/oliviertassinari
Omar Khan @omarkhan https://github.com/omarkhan
Ondej dych @zdychacek https://github.com/zdychacek
Operations Research Engineering Software+ @ORESoftware https://github.com/ORESoftware
Owen Evans @buildmaster https://github.com/buildmaster
Pablo Lacerda de Miranda @pablolmiranda https://github.com/pablolmiranda
Pascal Lalancette @okcompute https://github.com/okcompute
Patrick Williams @pwmckenna https://github.com/pwmckenna
Paul Dechov @dechov https://github.com/dechov
Paul Fox @gibbitz https://github.com/gibbitz
Paul Jaworski @paulwithap https://github.com/paulwithap
Paul Lunow @lunow https://github.com/lunow
Paul Melnikow @paulmelnikow https://github.com/paulmelnikow
Pauli Price @marfarma https://github.com/marfarma
Pavel @PavelPolyakov https://github.com/PavelPolyakov
Pavel Teshchin @satispunk https://github.com/satispunk
Pedro Costa Neves @persocon https://github.com/persocon
Pedro Pinho @donbitto https://github.com/donbitto
Pedro Teixeira @pgte https://github.com/pgte
Petar Dudev @dodev https://github.com/dodev
Peter A. Bigot @pabigot https://github.com/pabigot
Peter Berg @peteratticusberg https://github.com/peteratticusberg
Peter Chanthamnyavong @peterkc https://github.com/peterkc
Peter Conerly @pconerly https://github.com/pconerly
Peter Czibik @peteyycz https://github.com/peteyycz
Peter Lyons @focusaurus https://github.com/focusaurus

Open Source Used In DNAC Platform Antelope 3627
Peter Striman @stroiman https://github.com/stroiman
Petr Joachim @petrjoachim https://github.com/petrjoachim
Phaninder Pasupula @pasupulaphani https://github.com/pasupulaphani
Phil Wells @thephilwells https://github.com/thephilwells
Philip Kobernik @philipkobernik https://github.com/philipkobernik
Philipp Kretzschmar @k0pernikus https://github.com/k0pernikus
Philipp Kyeck @pyeck https://github.com/pyeck
Pierre Voisin @pvoisin https://github.com/pvoisin
Prabir Shrestha @prabirshrestha https://github.com/prabirshrestha
Prodigy @surya12badrinath https://github.com/surya12badrinath
Prokop Simek @prokopsimek https://github.com/prokopsimek
Puneet Gupta @puneetguptanitj https://github.com/puneetguptanitj
Rachel H @rainyday https://github.com/rainyday
Rafael @rgarcia https://github.com/rgarcia
Rahul Chaudhary @rash805115 https://github.com/rash805115
Raine Virta @raine https://github.com/raine
Raphael Luciano de Pontes @raphael-luciano https://github.com/raphael-luciano
Raquel Vlez @rockbot https://github.com/rockbot
Rashid Omar @rashtedude https://github.com/rashtedude
Ray Tung @raytung https://github.com/raytung
Reda @redben https://github.com/redben
Rhys Evans @wheresrhys https://github.com/wheresrhys
Ricard Fredin @Cordazar https://github.com/Cordazar
Ricardo Bordon @rjbordon https://github.com/rjbordon
Ricardo Gama @ricardogama https://github.com/ricardogama
Richard Harrington @FuzzySockets https://github.com/FuzzySockets
Richard Lay @richardlay https://github.com/richardlay
Richard Littauer @RichardLitt https://github.com/RichardLitt
Richard Meijer @richmeij https://github.com/richmeij
Richard Scarrott @richardscarrott https://github.com/richardscarrott
Richard Watkins @RichardWatkins1 https://github.com/RichardWatkins1
Rico Sta. Cruz @rstacruz https://github.com/rstacruz
Rob Ballou @robballou https://github.com/robballou
Rob Calcroft @robcalcroft https://github.com/robcalcroft
Rob Forsythe @robforsythe https://github.com/robforsythe
Rob Stevenson @thisdotrob https://github.com/thisdotrob
Rob Wu @Rob--W https://github.com/Rob--W
Robert Hurst @RobertWHurst https://github.com/RobertWHurst
Robert Simpson @rsimp https://github.com/rsimp
Robin Bobbitt @robinbobbitt https://github.com/robinbobbitt
Roch Devost @rochdev https://github.com/rochdev
Rocky Assad @CodisRedding https://github.com/CodisRedding
Rod Howard @rodhoward https://github.com/rodhoward
Rodrigo Gomes da Silva @rodrigogs https://github.com/rodrigogs
Rohit Kalkur @rovolution https://github.com/rovolution
Rolf Strijdhorst @rolfst https://github.com/rolfst
Romain @rprieto https://github.com/rprieto
Roman @moltar https://github.com/moltar
@bitgal https://github.com/bitgal
@blumus https://github.com/blumus
@brian-poncho https://github.com/brian-poncho
@chitsuhein https://github.com/chitsuhein
@clydehunt76 https://github.com/clydehunt76
@congdcit https://github.com/congdcit
@craigcosmo https://github.com/craigcosmo
@ddragosd https://github.com/ddragosd
@deepaktiwari-1987 https://github.com/deepaktiwari-1987
@devang-kredx https://github.com/devang-kredx
@divo-ie https://github.com/divo-ie
@dragoplut https://github.com/dragoplut
@ dunse https://github.com/dunse
@dynamitesushi https://github.com/dynamitesushi
@esatterwhite https://github.com/esatterwhite
@ewoutmeyns https://github.com/ewoutmeyns
@fent https://github.com/fent
@flushentitypacket https://github.com/flushentitypacket
@galenus https://github.com/galenus
@getlittletech https://github.com/getlittletech
@gmatroskin https://github.com/gmatroskin
@greenkeeper null
@handane123 https://github.com/handane123
@hellboy81 https://github.com/hellboy81
@hermano360 https://github.com/hermano360
@ianmacl https://github.com/ianmacl
@ikokostya https://github.com/ikokostya
@initialize https://github.com/initialize
@j-brown https://github.com/j-brown
@jeffsmale90 https://github.com/jeffsmale90
@jimkang https://github.com/jimkang
@karlismelderis https://github.com/karlismelderis
@keenaudio https://github.com/keenaudio
@kgividen https://github.com/kgividen
@kharandziuk https://github.com/kharandziuk
@kjona https://github.com/kjona
@krzysztofslonka https://github.com/krzysztofslonka
@lblazecki https://github.com/lblazecki
@leslc https://github.com/leslc
@liorur https://github.com/liorur
@loahou04 https://github.com/loahou04
@lock null
@louib https://github.com/louib
@lukas-gitl https://github.com/lukas-gitl
@madhavan020985 https://github.com/madhavan020985
@massimocode https://github.com/massimocode
@miazoom https://github.com/miazoom
@michaelmosher https://github.com/michaelmosher
1.1103 @babel/helper-get-function-arity

1.1103.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1104 source-map 0.5.7

1.1104.1 Available under license:

Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the names of the Mozilla Foundation nor the names of project contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1105 htrace-core4 4.2.0-incubating

1.1105.1 Available under license:

htrace-core4
Copyright 2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1106 source-map-support 0.4.18

1.1106.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Evan Wallace

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1107 fill-range 2.2.4

1.1107.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1108 esutils 2.0.3 2.0.3

1.1108.1 Available under license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1109 fast-levenshtein 2.0.6 2.0.6

1.1109.1 Available under license:
   (MIT License)

Copyright (c) 2013 [Ramesh Nair](http://www.hiddentao.com/)

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
1.1110 csv-parse 4.8.2 4.8.2

1.1110.1 Available under license :

The MIT License (MIT)

Copyright (c) 2010 Adaltas

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1111 bluebird 3.5.4

1.1111.1 Available under license :

MIT

1.1112 connect 3.4.1 3.4.1

1.1112.1 Available under license :

(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011-2014 TJ Holowaychuk
Copyright (c) 2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1113 asm 3.2

1.1113.1 Available under license:

No license file was found, but licenses were detected in source scan.

/****
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2007 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/commons/JSRInlinerAdapter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/JumpInsnNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/analysis/Value.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/VarInsnNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/analysis/SourceValue.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/analysis/SmallSet.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/commons/StaticInitMerger.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/util/TraceClassVisitor.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/AnnotationWriter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/InsnList.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/util/CheckSignatureAdapter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/util/TraceAnnotationVisitor.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/FrameNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/AnnotationVisitor.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/util/ASMifierMethodVisitor.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/util/ASMifierAbstractVisitor.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/MethodInsnNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/analysis/Subroutine.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/InnerClassNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/ClassAdapter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/commons/AdviceAdapter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/util/ASMifierFieldVisitor.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/util/TraceAbstractVisitor.java
Open Source Used In DNAC Platform Antelope 3646

* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/LdcInsnNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/tree/LineNumberNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/signature/SignatureReader.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/asm-all-3-2-sources-2-jar.org/objectweb/asm/tree/FieldNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/asm-all-3-2-sources-2-jar.org/objectweb/asm/commons/Remapper.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/asm-all-3-2-sources-2-jar.org/objectweb/asm/commons/TableSwitchGenerator.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/asm-all-3-2-sources-2-jar.org/objectweb/asm/tree/LocalVariableNode.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/asm-all-3-2-sources-2-jar.org/objectweb/asm/tree/TryCatchBlockNode.java
No license file was found, but licenses were detected in source scan.

/**
 * ASM XML Adapter
 * Copyright (c) 2004, Eugene Kuleshov
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-
jar/org/objectweb/asm/xml/SAXCodeAdapter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-
jar/org/objectweb/asm/xml/SAXClassAdapter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-
jar/org/objectweb/asm/xml/ASMContentHandler.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-
jar/org/objectweb/asm/xml/Processor.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-
jar/org/objectweb/asm/xml/SAXAnnotationAdapter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-
jar/org/objectweb/asm/xml/SAXFieldAdapter.java
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-
jar/org/objectweb/asm/xml/SAXAdapter.java

No license file was found, but licenses were detected in source scan.

* ASM XML Adapter
* Copyright (c) 2004, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/xml/package.html

No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2007 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */

Found in path(s):
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-
ASM XML Adapter
Copyright (c) 2004, Eugene Kuleshov
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1136493993_1613774733.35/0/asm-all-3-2-sources-2-jar/org/objectweb/asm/xml/asm-xml.dtd

No license file was found, but licenses were detected in source scan.

2005 INRIA, France Telecom
* All rights reserved.

* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1114 jooq 3.13.1

1.1114.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following
conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may
choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by
brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and
description of purpose be included on the same "printed page" as the copyright notice for easier identification within
third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1115 send 0.13.1 0.13.1

1.1115.1 Available under license:
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1116 socket.io 1.7.4 1.7.4

1.1116.1 Available under license:
(The MIT License)

Copyright (c) 2014-2016 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1117 compression 1.7.4

1.1117.1 Available under license:
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1118 datanucleus-core 4.1.17
1.1118.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS.
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2008-2013 DataNucleus

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ObjectWeb ASM 4.1 is repackaged in this datanucleus-core jar under org.datanucleus.asm. The files in that package are licensed under the following terms

=====================================================================  
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

====================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the DataNucleus distribution.       ==
====================================================================

This product includes software developed by many individuals, including the following:

Erik Bengtson
Andy Jefferson

This product has included contributions from some individuals, including the following:

Joerg von Frantzius
Thomas Marti
Barry Haddow
Marco Schulze
Ralph Ullrich
David Ezzio
Brendan de Beer
David Eaves
Martin Taal
Tony Lai
Roland Szabo
Marcus Mennemeier
Xuan Baldauf
Eric Sultan

This product also includes software developed by the TJDO project (http://tjdo.sourceforge.net/).
1.1119 de.huxhorn.lilith.logback.converter-classic 0.9.44

1.1119.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.
You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

/*
 * Lilith - a log event viewer.
 * Copyright (C) 2007-2011 Joern Huxhorn
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Lesser General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

/*
 * Copyright 2007-2011 Joern Huxhorn
 */
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

1.1120 drill-jdbc-all 1.19.0

1.1120.1 Available under license:

Drill : Exec : JDBC JAR with all dependencies
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1121 pascalcase 0.1.1

1.1121.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1122 batik-anim 1.13

1.1122.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMFEOffsetElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMAnimatedBoolean.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/AbstractAnimation.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMSVGEElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/XBLOMTemplateElement.java
Open Source Used In DNAC Platform Antelope 3673
Open Source Used in DNAC Platform Antelope

- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMForeignObjectElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMRadialGradientElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMFESpotLightElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/values/AnimatableMotionPointValue.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMFETurbulenceElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMFEMorphologyElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMGlyphElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMPolygonElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMFEFuncBElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMTextPositioningElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/values/AnimatableValue.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMLength.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMStopElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/AbstractElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMAElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVG12DOMImplementation.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/AbstractSVGAnimatedLength.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMRectElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/timing/TimegraphAdapter.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/XBLOMXBLElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMFEComponentTransferElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/SVGOMMURReferenceElement.java
- /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
  jar/org/apache/batik/anim/dom/AnimationTarget.java
Open Source Used In DNAC Platform Antelope 3676

* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGLocatableSupport.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/timing/OffsetTimingSpecifier.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMFEFuncRElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMDescElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMFlowParaElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMPolylineElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMSubImageRefElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMMetadataElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMTitleElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMMarkerElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMAnimatedLength.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGURIReferenceGraphicsElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMTSpanElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGOMToBeImplementedElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/timing/EventLikeTimingSpecifier.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/timing/Trace.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/SVGMVKernElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1.jar/org/apache/batik/anim/dom/TraitInformation.java
Open Source Used In DNAC Platform Antelope 3678

* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGPointShapeElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/values/AnimatableStringValue.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/values/AnimatableNumberOrPercentageValue.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/timing/IndefiniteTimingSpecifier.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/timing/AccesskeyTimingSpecifier.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMFlowRegionBreakElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMTextElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMImageElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/values/AnimatableNumberValue.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/values/AnimatablePointListValue.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/AnimationTargetListener.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/TransformAnimation.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/values/AnimatableBooleanValue.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/values/AnimatableLengthValue.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/XBLEventSupport.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMColorProfileElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMUseElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMAltGlyphItemElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/XBLOMElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMPatternElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMViewElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/dom/SVGOMGElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/timing/TimeContainer.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/timing/WallclockTimingSpecifier.java
Open Source Used In DNAC Platform Antelope 3680
No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/dom/XBLOMContentElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/dom/SVGMOMAnimateMotionElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/values/AnimatableAngleOrIdentValue.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/dom/SVGMOMAnimateColorElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/values/AnimatableLengthListValue.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/timing/TimedElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/dom/SVGMOMStyleElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/dom/SVGMOMElement.java
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-
jar/org/apache/batik/anim/dom/SVGMOMAnimatedTransformList.java

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/org/apache/batik/anim/AnimationException.java
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
collector license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1146878560_1616835256.03/0/batik-anim-1-13-sources-1-jar/META-INF/maven/org.apache.xmlgraphics/batik-anim/pom.xml

1.1123 request 2.87.0 2.87.0

1.1123.1 Available under license:
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date
such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1124 keypress 0.1.0
1.1124.1 Available under license:
No license file was found, but licenses were detected in source scan.

keypress
========

### Make any Node ReadableStream emit "keypress" events

Previous to Node `v0.8.x`, there was an undocumented `"keypress"` event that `process.stdin` would emit when it was a TTY. Some people discovered this hidden gem, and started using it in their own code.

Now in Node `v0.8.x`, this `"keypress"` event does not get emitted by default, but rather only when it is being used in conjunction with the `readline` (or by extension, the `repl`) module.

This module is the exact logic from the node `v0.8.x` releases ripped out into its own module.

__Bonus:__ Now with mouse support!

Installation
------------

Install with `npm`:

```bash
$ npm install keypress
```

Or add it to the `"dependencies"` section of your `_package.json_` file.
Example
-------

#### Listening for "keypress" events

``` js
var keypress = require('keypress');

// make `process.stdin` begin emitting "keypress" events
keypress(process.stdin);

// listen for the "keypress" event
process.stdin.on('keypress', function (ch, key) {
    console.log('got "keypress"', key);
    if (key && key.ctrl && key.name == 'c') {
        process.stdin.pause();
    }
});

process.stdin.setRawMode(true);
process.stdin.resume();
```

#### Listening for "mousepress" events

``` js
var keypress = require('keypress');

// make `process.stdin` begin emitting "mousepress" (and "keypress") events
keypress(process.stdin);

// you must enable the mouse events before they will begin firing
keypress.enableMouse(process.stdout);

process.stdin.on('mousepress', function (info) {
    console.log('got "mousepress" event at %d x %d', info.x, info.y);
});

process.on('exit', function () {
    // disable mouse on exit, so that the state
    // is back to normal for the terminal
    keypress.disableMouse(process.stdout);
});
```

License
-------
(The MIT License)

Copyright (c) 2012 Nathan Rajlich &lt;nathan@tootallnate.net&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):
* /opt/cola/permits/1127121008_1611790907.33/0/tootallnate-keypress-v0-1-0-0-ga7457cc-1-tar-gz/TooTallNate-keypress-a7457cc/README.md

**1.1125 redux-saga 0.16.0 0.16.0**

**1.1125.1 Available under license:**

The MIT License (MIT)

Copyright (c) 2015 Yassine Elouafi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1126 tslib 1.14.1
1.1126.1 Available under license:

/*! *****************************************************************************/
/* Copyright (c) Microsoft Corporation. */

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

/*! *****************************************************************************/
/* Copyright (c) Microsoft Corporation. */

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1127 entities 1.1.1 1.1.1
1.1127.1 Available under license:

Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following

disclaimer.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1128 drill-rpc 1.19.0

1.1128.1 Available under license:

Drill : Exec : RPC
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1129 json5 0.5.1 0.5.1

1.1129.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1130 ecdsa-sig-formatter 1.0.11
1.1130.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2015 D2L Corporation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1131 query-string 7.0.0 7.0.0

1.1131.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (http://sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1132 react-highlight-words 0.16.0 0.16.0

1.1132.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Treasure Data

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1133 hbase-hadoop-compat 2.0.0-alpha4

1.1133.1 Available under license:

Apache HBase - Hadoop Two Compatibility
Copyright 2007-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1134 resolve 1.9.0
1.1134.1 Available under license:
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1135 kerb-common 1.0.0
1.1135.1 Available under license:
Kerby-kerb Common
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
This software is covered under the following copyright:

/*
 * Copyright (c) 2003-2005 Tom Wu
 * All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY
 * WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
 *
 * IN NO EVENT SHALL TOM WU BE LIABLE FOR ANY SPECIAL, INCIDENTAL,
 * INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER
 * RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF
 * THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT
 * OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * In addition, the following condition applies:
 *
 * All redistributions must retain an intact copy of this copyright notice
 * and disclaimer.
 */

Address all questions regarding this license to:

Tom Wu
tjw@cs.Stanford.EDU
1.1137 fs 0.0.2 0.0.2
1.1137.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1138 disruptor 3.4.2

1.1139 compress-lzf 1.1
1.1139.1 Available under license:

# Compress LZF

This library contains efficient implementation of LZF compression format, as well as additional helper classes that build on JDK-provided gzip (deflat) codec.

## Licensing

Library is licensed under Apache License 2.0, as per accompanying LICENSE file.

## Credit
Library has been written by Tatu Saloranta (tatu.saloranta@iki.fi).
It was started at Ning, Inc., as an official Open Source process used by
platform backend, but after initial versions has been developed outside of
Ning by supporting community.

Other contributors include:

* Jon Hartlaub (first versions of streaming reader/writer; unit tests)
* Cedrik Lime: parallel LZF implementation

Various community members have contributed bug reports, and suggested minor
fixes; these can be found from file "VERSION.txt" in SCM.

This copy of Compress-LZF library is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.1140 wtf-8 1.0.0 1.0.0

1.1140.1 Available under license:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1141 babel-types 6.26.0

1.1141.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>

Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
1.1142 jsreport-data 2.4.0

1.1142.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:
a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:
a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.1143 xtend 4.0.2

1.1143.1 Available under license:

The MIT License (MIT)
Copyright (c) 2012-2014 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1144 @babel/plugin-transform-object-super
7.12.13
1.1144.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1145 ajv 6.10.2 6.10.2
1.1145.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1146 zlib 1.2.1
1.1146.1 Available under license :
/* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.1, November 17th, 2003

Copyright (C) 1995-2003 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
 misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly       Mark Adler
jlooup@gzip.org        madler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for
Comments) 1950 to 1952 in the files http://www.ietf.org/rfc/rfc1950.txt
(zlib format), rfc1951.txt (deflate format) and rfc1952.txt (gzip format).
1.1147 kotlin 1.3.61
1.1147.1 Available under license:

No license file was found, but licenses were detected in source scan.
convenient ways of constructing Longs.

The internal representation of a long is the two given signed, 32-bit values.

We use 32-bit pieces because these are the size of integers on which Javascript performs bit-operations. For operations like addition and multiplication, we split each number into 16-bit pieces, which can easily be multiplied within Javascript's floating-point representation without overflow.

Not handling this case would often result in infinite recursion.

The high (signed) 32 bits of the long.

We use 32-bit pieces because these are the size of integers on which Javascript performs bit-operations. For operations like addition and multiplication, we split each number into 16-bit pieces, which can easily be multiplied within Javascript's floating-point representation without overflow or change in sign.

In the algorithms below, we frequently reduce the negative case to the positive case by negating the input(s) and then post-processing the result.

Note that we must ALWAYS check specially whether those values are MIN_VALUE or -MIN_VALUE (since 2^63 cannot be represented as a positive number, it overflows back into a negative). Not handling this case would often result in infinite recursion.

The low (signed) 32 bits of the long.

The low (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.

The high (signed) 32 bits of the long.
This must appear before the first call to any function below:

```kotlin
Kotlin.Long.TWO_PWR_16_DBL_ = 1 << 16;
Kotlin.Long.TWO_PWR_24_DBL_ = 1 << 24;
Kotlin.Long.TWO_PWR_32_DBL_ = 1 << 32;
Kotlin.Long.TWO_PWR_64_DBL_ = 1 << 64;
Kotlin.Long.TWO_PWR_48_DBL_ = Kotlin.Long.TWO_PWR_16_DBL_ *
  Kotlin.Long.TWO_PWR_32_DBL_;
Kotlin.Long.TWO_PWR_63_DBL_ = Kotlin.Long.TWO_PWR_64_DBL_ / 2;
Kotlin.Long.TWO_PWR_64_DBL_ = Kotlin.Long.TWO_PWR_32_DBL_ * 2;
Kotlin.Long.TWO_PWR_31_DBL_ = Kotlin.Long.TWO_PWR_32_DBL_;
Kotlin.Long.TWO_PWR_30_DBL_ = Kotlin.Long.TWO_PWR_31_DBL_ / 2;
Kotlin.Long.ZERO = Kotlin.Long.fromInt(0);
Kotlin.Long.ONE = Kotlin.Long.fromInt(1);
Kotlin.Long.MAX_VALUE = Kotlin.Long.fromInt(0x7FFFFFFF | 0x7FFFFFFF | 0);
Kotlin.Long.MIN_VALUE = Kotlin.Long.fromInt(0x80000000 | 0);
Kotlin.Long.TWO_PWR_24_ = Kotlin.Long.fromInt(1 << 24);
Kotlin.Long.prototype.toInt = function() {
  return this.low_;
};
Kotlin.Long.prototype.toNumber = function() {
  return this.high_ * Kotlin.Long.TWO_PWR_32_DBL_ +
  this.getLowBitsUnsigned();
};
Kotlin.Long.prototype.hashCode = function() {
  return this.high_ ^ this.low_;
};
Kotlin.Long.prototype.toString = function(opt_radix) {
  var radix = opt_radix || 10;
  if (radix < 2 || 36 < radix) {
    throw Error('radix out of range: ' + radix);
  }

  if (this.isZero()) {
    return '0';
  }

  if (this.isNegative()) {
    if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
      return '-0';
    } else {
      return '-' + this.negate().toString(radix);
    }
  }

  // Do several (6) digits each time through the loop,
  // so as to minimize the calls to the very expensive 
  // emulated div.
  var radixLong = Kotlin.Long.fromInt(10);
  var var rem = this;
  var result = '';
  while (true) {
    var intval = rem.subtract(radixLong.multiply(var div));
    var digits = intval.toString(radix);

    rem = intval;
    if (rem.isZero()) {
      return digits;
    } else {
      result = '' + digits + result;
    }
  }
};
```
```javascript
/**
 * @param {Kotlin.Long} other Long to compare against.
 * @return {boolean} Whether this Long does not equal the other.
 */
Kotlin.Long.prototype.notEqualsLong = function(other) {
  return this.compare(other) != 0;
};


/**
 * Compares this Long with the given one.
 * @param {number} other number to compare against.
 * @return {number} 0 if they are the same, 1 if the this is greater, and -1 if
 *     the given one is greater.
 */
Kotlin.Long.prototype.compare = function(other) {
  if (this.equalsLong(other)) {
    return 0;
  }

  var thisNeg = this.isNegative();
  var otherNeg = other.isNegative();
  if (thisNeg && !otherNeg) {
    return -1;
  }
  if (!thisNeg && otherNeg) {
    return 1;
  }

  // at this point, the signs are the same, so subtraction will not overflow
  if (this.subtract(other).isNegative()) {
    return -1;
  } else {
    return 1;
  }
};


/** @return {!Kotlin.Long} The negation of this value. */
Kotlin.Long.prototype.negate = function() {
  if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return Kotlin.Long.MIN_VALUE;
  } else {
    return this.not().add(Kotlin.Long.ONE);
  }
};


/**
 * Returns the sum of this and the given Long.
 * @param {Kotlin.Long} other Long to add to this one.
 * @return {!Kotlin.Long} The sum of this and the given Long.
 */
Kotlin.Long.prototype.add = function(other) {
  // Divide each number into 4 chunks of 16 bits, and then sum the chunks.
  var a48 = this.high_ >>> 16;
  var a32 = this.high_ & 0xFFFF;
  var a16 = this.low_ >>> 16;
  var a00 = this.low_ & 0xFFFF;

  var b48 = other.high_ >>> 16;
  var b32 = other.high_ & 0xFFFF;
  var b16 = other.low_ >>> 16;
  var b00 = other.low_ & 0xFFFF;

  var c48 = 0, c32 = 0, c16 = 0, c00 = 0;

  c00 += a00 + b00;
  c16 += c00 >>> 16;
  c00 &= 0xFFFF;
  c16 += a16 + b16;
  c32 += c16 >>> 16;
  c16 &= 0xFFFF;
  c32 += a32 + b32;
  c48 += c32 >>> 16;
  c32 &= 0xFFFF;
  c48 += a48 + b48;
  c48 &= 0xFFFF;
  return Kotlin.Long.fromBits(c48, c32, c16, c00, c48 << 16);
};


/**
 * Returns the product of this and the given long.
 * @param {Kotlin.Long} other Long to multiply with this.
 * @return {!Kotlin.Long} The product of this and the other.
 */
Kotlin.Long.prototype.multiply = function(other) {
  if (this.isZero()) {
    return Kotlin.Long.ZERO;
  } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return this.isOdd() ? Kotlin.Long.MIN_VALUE :
  }
```
0xFFFF:
  var b16 = other.low_ >>> 16;
  var b00 = other.low_ & 0xFFFF;

  var c48 = 0, c32 = 0, c16 = 0, c00 = 0;
  var c00 = a00 * b00; var c16 = a16 * b00;
  var c32 = a32 * b00;
  var c48 = a48 * b00;

  return Kotlin.Long.fromBits(c16 << 16) | c00, (c48 << 16) | c32);
};

/**
 * Returns this Long divided by the given one.
 * @param {Kotlin.Long} other Long by which to divide.
 * @return {!Kotlin.Long} Returns this Long divided by the given one.
 */
Kotlin.Long.prototype.div = function(other) {
  if (other.isZero()) {
    throw Error('division by zero');
  } else if (this.equalsLong(Kotlin.Long.ZERO)) {
    return Kotlin.Long.ZERO;
  }

  if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    if (other.equalsLong(Kotlin.Long.ONE) ||
        other.equalsLong(Kotlin.Long.NEG_ONE)) {
      return Kotlin.Long.MIN_VALUE;  // recall that -MIN_VALUE == MIN_VALUE
    } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
      return Kotlin.Long.ONE;
    } else {
      // At this point, we have |other| >= 2, so |this/other| < |MIN_VALUE|.
      var halfThis = this.shiftRight(1);
      var approx = halfThis.div(other).shiftLeft(1);
      if (approx.equalsLong(Kotlin.Long.ZERO)) {
        return other.isNegative() ? Kotlin.Long.ONE : Kotlin.Long.NEG_ONE;
      } else {
        var rem = this.subtract(other.multiply(approx));
        var result = approx.add(rem.div(other));
        return result;
      }
    }
  } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return Kotlin.Long.ZERO;
  }

  if (this.isNegative()) {
    if (other.isNegative()) {
      return this.negate().div(other.negate());
    } else {
      return this.negate().div(other).negate();
    }
  } else if (other.isNegative()) {
    return this.div(other.negate()).negate();
  }

  // Repeat the following until the remainder is less than other:  find an //
  // floating-point that approximates remainder / other *from below*, add this\n  // into the result, and subtract it from
  // the remainder.  It is critical that\n  // the approximate value is less than or equal to the real value so that the\n  // remainder never becomes negative.\n  //\n  // We will tweak the approximate result by changing it in the 48-th digit of\n  // the smallest non-fractional digit,\n  // whichever is larger.\n  //\n  // We know the answer can't be zero... and actually, zero would cause\n  // infinite recursion since we would make no progress.\n  if (approxRes.isZero()) {
    approxRes = Kotlin.Long.ONE;
  }

  var approx = Math.max(1, Math.floor(dividend / divisor));
  var result = approx.add(rem.div(divisor));
  return result;
};

/**
 * Returns the bitwise-OR of this Long and the given one.
 * @param {Kotlin.Long} other The Long with which to OR.
 * @return {!Kotlin.Long} The bitwise-OR of this and the other.
 */
Kotlin.Long.prototype.or = function(other) {
  return Kotlin.Long.fromBits(this.low_ | other.low_,
                               this.high_ | other.high_);
Kotlin.Long.prototype.shiftLeft = function(numBits) {
  numBits &= 63;
  if (numBits == 0) {
    return this;
  } else {
    var low = this.low_;
    if (numBits < 32) {
      var high = this.high_;
      return Kotlin.Long.fromBits(
          low << numBits,
          (high << numBits) | (low >>> (32 - numBits)));
    } else {
      return Kotlin.Long.fromBits(0, low << (numBits - 32));
    }
  }
}

/**
 * Returns this Long with bits shifted to the right by the given amount.
 * @param {number} numBits The number of bits by which to shift.
 * @return {!Kotlin.Long} This shifted to the right by the given amount.
 */
Kotlin.Long.prototype.shiftRight = function(numBits) {
  numBits &= 63;
  if (numBits == 0) {
    return this;
  } else {
    var high = this.high_;
    if (numBits < 32) {
      var low = this.low_;
      return Kotlin.Long.fromBits(
          (low >>> numBits) | (high << (32 - numBits)),
          high >> numBits);
    } else {
      return Kotlin.Long.fromBits(high >> (numBits - 32),
          high >= 0 ? 0 : -1);
    }
  }
}

/**
 * Returns this Long with bits shifted to the right by the given amount, with zeros
 * placed into the new leading bits.
 * @param {number} numBits The number of bits by which to shift.
 * @return {!Kotlin.Long} This shifted to the right by the given amount, with zeros
 *     placed into the new leading bits.
 */
Kotlin.Long.prototype.shiftRightUnsigned = function(numBits) {
  numBits &= 63;
  if (numBits == 0) {
    return this;
  } else {
    var high = this.high_;
    if (numBits < 32) {
      var low = this.low_;
      return Kotlin.Long.fromBits(
          (low >>> numBits) | (high << (32 - numBits)),
          high >>> numBits);
    } else if (numBits == 32) {
      return Kotlin.Long.fromBits(high, 0);
    } else {
      return Kotlin.Long.fromBits(high >>> (numBits - 32), 0);
    }
  }
}

// Support for Kotlin
Kotlin.Long.prototype.equals = function (other) {
    return other instanceof Kotlin.Long &&
    this.equalsLong(other);
}

Kotlin.defineModule = function (id, declaration) {
};

Kotlin.defineInlineFunction = function(tag, fun) {
    return fun;
};

Kotlin.wrapFunction = function(fun) {
    var f = function() {
        f = fun();
        return f.apply(this, arguments);
    };
    return function() {
        return f.apply(this, arguments);
    }
};

Kotlin.isTypeOf = function(type) {
    return function (object) {
        return typeof object === type;
    }
};

Kotlin.isInstanceOf = function (klass) {
    return function (object) {
        return Kotlin.isType(object,
            klass);
    }
};

Kotlin.orNull = function (fn) {
    return function (object) {
        return object == null ||
        fn(object);
    }
};

Kotlin.andPredicate = function (a, b) {
    return function (object) {
        return a(object) &&
        b(object);
    }
};

Kotlin.kotlinModuleMetadata = function (abiVersion, moduleName, data) {
};

Kotlin.suspendCall = function(value) {
    return value;
};

Kotlin.coroutineResult = function(qualifier)
{ throwMarkerError();
};

Kotlin.coroutineController = function(qualifier) {
  throwMarkerError();
};

Kotlin.coroutineReceiver = function(qualifier) {
  throwMarkerError();
};

Kotlin.setCoroutineResult = function(value, qualifier) {
  throwMarkerError();
};

function throwMarkerError() {
    throw new Error("This marker function should never been called."
        + "Looks like compiler did not eliminate it properly.
        Please, report an issue if you caught this exception.");
}

Kotlin.getFunctionById = function(id) {
    return function() {
        return defaultValue;
    }
};
found in the license/LICENSE.txt file.

```kotlin
Kotlin.compareTo = function (a, b) {
    var typeA = typeof a;
    if (typeA === "number") {
        if (typeof b === "number") {
            return Kotlin.doubleCompareTo(a, b);
        }
        return Kotlin.primitiveCompareTo(a, b);
    }
    if (typeA === "string" || typeA === "boolean") {
        return a.compareTo_11rb$(b);
    }
    return a.compareTo_11rb$(b);
};

Kotlin.primitiveCompareTo = function (a, b) {
    return a < b ? -1 : a > b ? 1 : 0;
};

Kotlin.doubleCompareTo = function (a, b) {
    if (a < b) return -1;
    if (a > b) return 1;
    return a !== b ? 0 : 1;
};

Kotlin.charInc = function (value) {
    return Kotlin.toChar(value + 1);
};

Kotlin.charDec = function (value) {
    return Kotlin.toChar(value - 1);
};

if (typeof String.prototype.startsWith === "undefined") {
    String.prototype.startsWith = function(searchString, position) {
        position = position || 0;
        return this.lastIndexOf(searchString, position) === position;
    };
}

if (typeof Math.sign === "undefined") {
    Math.sign = function(x) {
        x = +x; // convert to a number
        if (x === 0 || isNaN(x)) {
            return Number(x);
        }
        return x > 0 ? 1 : -1;
    };
}

(function() {
    var epsilon = 2.220446049250313E-16;
    var taylor_2_bound = Math.sqrt(epsilon);
    var taylor_n_bound = Math.sqrt(taylor_2_bound);
    var upper_taylor_2_bound = 1 / taylor_2_bound;
    var upper_taylor_n_bound = 1 / taylor_n_bound;

    if (typeof Math.sinh === "undefined") {
        Math.sinh = function(x) {
            if (Math.abs(x) < taylor_n_bound) {
                var result = x;
                if (Math.abs(x) > taylor_2_bound) {
                    result += (x * x * x) / 6;
                }
                return result;
            } else {
                var y = Math.exp(x);
                var y1 = 1 / y;
                var result = (y - y1) / 2;
            }
        };
    }
```

Open Source Used In DNAC Platform Antelope 3729
if (typeof Math.tanh === "undefined") {
    Math.tanh = function(x) {
        return (Math.exp(x) - Math.exp(-x)) / (Math.exp(x) + Math.exp(-x));
    };
}

// approximation by laurent series in x at 0 up to order 2
var result = x;
if (Math.abs(x) >= taylor_n_bound) {
    var x3 = x * x * x;
    // approximation by taylor series in x at 0 up to order 4
    return (-x4 / 4 + x3 / 6 - x2 / 2 + x);
}

else if (x - 1 >= taylor_n_bound) {
    var result = x;
    if (Math.abs(x) >= taylor_2_bound) {
        var x3 = x * x * x;
        // approximation by taylor series in x at 0 up to order 4
        return -asinh(-x);
    }
    else
    {
        // approximation by taylor series in x at 0 up to order 2
        var result = x;
        if (y >= taylor_2_bound) {
            var y3 = y * y * y;
            // approximation by taylor series in y at 0 up to order 4
            return -y / 2 + y2 / 6 - y3 / 24 + y4 / 720;
        }
        else
        {
            var y = Math.sqrt(x - 1);
            // approximation by taylor series in y at 0
            return Math.log(y + Math.sqrt(y * y + 1));
        }
    }
}
else if (x <= -taylor_n_bound) {
    var result = x;
    if (Math.abs(x) > taylor_2_bound) {
        var x3 = x * x * x;
        // approximation by taylor series in x at 0 up to order 4
        return -asinh(x);
    }
    else
    {
        // approximation by laurent series in 1/x at 0+ order from -1 to 1
        var y = Math.sqrt(x - 1);
        // approximation by taylor series in y at 0
        return Math.log((1 + y) / (1 - y)) / 2;
    }
}
else
{
    var y = Math.sqrt(x + Math.sqrt(x * x + 1));
    // approximation by taylor series in y at 0
    return Math.log(y + Math.sqrt(y * y + 1));
}

if (typeof Math.asinh === "undefined") {
    Math.asinh = function(x) {
        if (x >= +taylor_n_bound)
        {
            if (x > upper_taylor_n_bound)
            {
                return Math.asinh(+x);
            }
            else
            {
                if (x >= 1)
                {
                    return Math.log(x + Math.sqrt(x * x + 1));
                }
                else
                {
                    return Math.log(x * 2 + (1 / (x * 2)));
                }
            }
        }
        else
        {
            if (x < -taylor_n_bound)
            {
                return Math.asinh(-x);
            }
            else
            {
                var a = Math.exp(x), b = Math.exp(-x);
                return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
            }
        }
        };
    }

    // Inverse hyperbolic function implementations derived from boost special math functions,

    if (typeof Math.acosh === "undefined") {
        Math.acosh = function(x) {
            if (x < 1)
            {
                return NaN;
            }
            else
            {
                return Math.log(x + Math.sqrt(x * x - 1));
            }
        };
    }
    if (typeof Math.atanh === "undefined") {
        Math.atanh = function(x) {
            if (Math.abs(x) < taylor_n_bound) {
                var result = x;
                if (y >= taylor_2_bound) {
                    var y3 = y * y;
                    // approximation by taylor series in y at 0 up to order 4
                    return -y / 2 + y2 / 6 - y3 / 24 + y4 / 720;
                }
                else
                {
                    var y = Math.sqrt(x - 1);
                    // approximation by taylor series in y at 0
                    return Math.log((1 + y) / (1 - y)) / 2;
                }
            }
            else
            {
                // approximation by laurent series in 1/x at 0+ order from -1 to 1
                var y = Math.sqrt(x - 1);
                // approximation by taylor series in y at 0
                return Math.log((1 + y) / (1 - y)) / 2;
            }
        };
    }
    if (typeof Math.log1p === "undefined") {
        Math.log1p = function(x) {
            if (Math.abs(x) < taylor_n_bound) {
                var result = x;
                if (Math.abs(x) > taylor_2_bound) {
                    var x3 = x * x * x;
                    // approximation by taylor series in x at 0 up to order 4
                    return (-x4 / 4 + x3 / 6 - x2 / 2 + x);
                }
                else
                {
                    var a = Math.exp(x), b = Math.exp(-x);
                    return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
                }
            }
            else
            {
                var a = Math.exp(x), b = Math.exp(-x);
                return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
            }
        };
    }
    if (typeof Math.log10 === "undefined") {
        Math.log10 = function(x) {
            if (x >= +taylor_n_bound)
            {
                if (x > upper_taylor_n_bound)
                {
                    return Math.log10(+x);
                }
                else
                {
                    return (Math.log(x) * Math.LN10);
                }
            }
            else
            {
                return (Math.log(x) * Math.LN10);
            }
        };
    }
    if (typeof Math.cbrt === "undefined") {
        Math.cbrt = function(x) {
            if (Math.abs(x) < taylor_n_bound) {
                var result = x;
                if (Math.abs(x) > taylor_2_bound) {
                    var x3 = x * x * x;
                    // approximation by taylor series in x at 0 up to order 4
                    return x3 / 3 - (x2 / 2 + x) / 6;
                }
                else
                {
                    var a = Math.exp(x), b = Math.exp(-x);
                    return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
                }
            }
            else
            {
                var a = Math.exp(x), b = Math.exp(-x);
                return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
            }
        };
    }
    if (typeof Math.hypot === "undefined") {
        Math.hypot = function() {
            var y = 0;
            var length = arguments.length;
            for (var i = 0; i < length; i++)
            {
                var y += arguments[i] ** 2;
            }
            return Math.sqrt(y);
        };
    }
    if (typeof Math.clz32 === "undefined") {
        Math.clz32 = (function(log, LN2) {
            return function(x) {
                var asUint = x >>> 0;
                if (asUint === 0) {
                    return 32;
                }
                return 31 - (log(asUint) / Math.LN2);
            };};
        })((Math.log, Math.LN2), Math.NA);
    }
    if (typeof Math.log2 === "undefined") {
        Math.log2 = function(x) {
            return Math.log(x) * Math.LOG2E;
        };
    }
    if (typeof Math.log10 === "undefined") {
        Math.log10 = function(x) {
            return Math.log(x) * Math.LN10;
        };
    }
    if (typeof Math.expm1 === "undefined") {
        Math.expm1 = function(x) {
            if (Math.abs(x) < taylor_n_bound) {
                var result = x;
                if (x > upper_taylor_n_bound)
                {
                    result -= x4 / 4 + x3 / 6 - x2 / 2 + x;
                }
                else
                {
                    result -= (x * x * x) / 3;
                }
            }
            else
            {
                var a = Math.exp(x), b = Math.exp(-x);
                return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
            }
        };
    }
    if (typeof Math.acosh === "undefined") {
        Math.acosh = function(x) {
            if (x < 1)
            {
                return NaN;
            }
            else
            {
                return Math.log((1 + x) / (1 - x)) / 2;
            }
        };
    }
    if (typeof Math.atanh === "undefined") {
        Math.atanh = function(x) {
            if (Math.abs(x) < taylor_n_bound) {
                var result = x;
                if (y >= taylor_2_bound) {
                    var y3 = y * y;
                    // approximation by taylor series in y at 0 up to order 4
                    return -y / 2 + y2 / 6 - y3 / 24 + y4 / 720;
                }
                else
                {
                    var y = Math.sqrt(x - 1);
                    // approximation by taylor series in y at 0
                    return Math.log((1 + y) / (1 - y)) / 2;
                }
            }
            else
            {
                var a = Math.exp(x), b = Math.exp(-x);
                return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
            }
        };
    }
}
var relativeStart = start >> 0;

if (offset < 0) return Math.max(0, offset + length);

var relativeEnd = end >> 0;

if (type end === "undefined") { throw new RangeError("Array.prototype.slice operator attempted to retrieve value at index out of range 
0 <= index < length."); }

var len = this.length;

if (begin >= len) begin = 0;

if (end >= len) end = len;

if (relativeStart < 0) {
    relativeStart = Math.max(0, len + relativeStart);
}

if (relativeEnd < 0) {
    relativeEnd = Math.max(0, len + relativeEnd);
}

relativeStart = Math.max(0, Math.min(relativeStart, len));
relativeEnd = Math.max(0, Math.min(relativeEnd, len));

var start = relativeStart;
var end = relativeEnd;

var k = start < 0 ? Math.max(len + start, 0) : Math.min(start, len);
var end = this.length < 0 ? Math.max(len + end, 0) : Math.min(end, len);

var O = [
    function slice(callback, self) {
        return 
            // Patch to work with TypedArrays if needed.
            if (typeof callback === "function") {
                return callback.call(self, 
                    this)
            } else {
                return 
                    // Patch to work with TypedArrays if needed.
                    this.map(callback, self);
            }
    }
];

try {
    // Patch to work with TypedArrays if needed.
    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)
    totalOrderComparator = function (a, b) {
        if (a < b) return -1;
        if (a > b) return 1;

        if (a === b) {
            if (a !== 0) return 0;

            var ia = 1 / a;
            return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);
        }

        return a !== a ? (b !== b ? 0 : 1) : -1
    };
}

for (var i = 0; i < arrays.length; ++i) {
    var TypedArray = arrays[i];
    if (typeof TypedArray.prototype.sort === "undefined") {
        Object.defineProperty(TypedArray.prototype, 'sort', {
            value: function(compareFunction) {
                return Array.prototype.sort.call(this, compareFunction || totalOrderComparator);
            }
        });
    }

    // Patch sort to work with TypedArrays if needed.
    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)

    if (typeof TypedArray.prototype.map === "undefined") {
        Object.defineProperty(TypedArray.prototype, 'map', {
            value: function(callback, self) {
                return 
                    // Patch to work with TypedArrays if needed.
                    this.map(callback, self);
            }
        });
    }

    // Patch sort to work with TypedArrays if needed.
    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)

    // Patch map to work with TypedArrays if needed.
    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)

    // Patch map to work with TypedArrays if needed.
    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)

    // Patch map to work with TypedArrays if needed.
    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)

    // Patch map to work with TypedArrays if needed.
    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)

    // Patch map to work with TypedArrays if needed.
    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)

    // Patch map to work with TypedArrays if needed.
isInheritanceFromInterface(superConstructor, iface);

switch (typeof object) {
    case 'string'
        case 'number'
        case 'boolean'
            return true;
    default
        return object instanceof Object;
}

if (object === null || klass === null || (typeof object === 'object' && typeof object !== 'function')) {
    return false;
}

if (typeof klass === 'function' && object instanceof klass) {
    return true;
}

var proto = Object.getPrototypeOf(klass);
var constructor = proto.constructor;
if (constructor !== null) {
    constructor = constructor.$metadata$;
    if (constructor === null) {
        return object instanceof klass;
    } else if (constructor.kind === Kotlin.Kind.OBJECT) {
        return object === klass;
    }
}

var Metaclass = klass.$metadata$; // In WebKit (JavaScriptCore) for some interfaces from DOM typeof returns `Object`, nevertheless they can be used in RHS of instanceof if (klassMetadata === null) {
    return object instanceof klass;
}

if (klassMetadata.kind === Kotlin.Kind.INTERFACE && object.constructor !== null) {
    return isInheritanceFromInterface(object.constructor, klass);
}

return false;

Kotlin.isNumber = function (a) {
    return typeof a == 'number' || a instanceof Kotlin.Long;
};

Kotlin.isChar = function (value) {
    return value instanceof Kotlin.BoxedChar
};

Kotlin.isComparable = function (value) {
    var type = typeof value;

    return type === 'string' ||
           type === 'boolean' ||
           Kotlin.isNumber(value) ||
           Kotlin.isType(value, Kotlin.kotlin.Comparable);
};

Kotlin.isCharSequence = function (value) {
    return typeof value === 'string'
};

override fun toString() = name

companion object
}
operator fun ByteArray.component2(): Byte {\n    return get(1)\n}

operator fun ByteArray.component3(): Byte {\n    return get(2)\n}

operator fun ByteArray.component4(): Byte {\n    return get(3)\n}

operator fun ByteArray.component5(): Byte {\n    return get(4)\n}

operator fun ShortArray.component2(): Short {\n    return get(1)\n}

operator fun ShortArray.component3(): Short {\n    return get(2)\n}

operator fun ShortArray.component4(): Short {\n    return get(3)\n}

operator fun ShortArray.component5(): Short {\n    return get(4)\n}

operator fun IntArray.component2(): Int {\n    return get(1)\n}

operator fun IntArray.component3(): Int {\n    return get(2)\n}

operator fun IntArray.component4(): Int {\n    return get(3)\n}

operator fun IntArray.component5(): Int {\n    return get(4)\n}

operator fun LongArray.component2(): Long {\n    return get(1)\n}

operator fun LongArray.component3(): Long {\n    return get(2)\n}

operator fun LongArray.component4(): Long {\n    return get(3)\n}

operator fun LongArray.component5(): Long {\n    return get(4)\n}

operator fun FloatArray.component2(): Float {\n    return get(1)\n}

operator fun FloatArray.component3(): Float {\n    return get(2)\n}

operator fun FloatArray.component4(): Float {\n    return get(3)\n}

operator fun FloatArray.component5(): Float {\n    return get(4)\n}

operator fun DoubleArray.component2(): Double {\n    return get(1)\n}

operator fun DoubleArray.component3(): Double {\n    return get(2)\n}

operator fun DoubleArray.component4(): Double {\n    return get(3)\n}

operator fun DoubleArray.component5(): Double {\n    return get(4)\n}

operator fun BooleanArray.component2(): Boolean {\n    return get(1)\n}

operator fun BooleanArray.component3(): Boolean {\n    return get(2)\n}

operator fun BooleanArray.component4(): Boolean {\n    return get(3)\n}

operator fun BooleanArray.component5(): Boolean {\n    return get(4)\n}
inline operator fun FloatArray.component5(): Float
    return get(4)

/**
 * Returns 5th *element* from the collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component5(): Double
    return get(4)

/**
 * Returns 5th *element* from the collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component5(): Boolean
    return get(4)

/**
 * Returns 5th *element* from the collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component5(): Char
    return get(4)

inline operator fun FloatArray.component5(): Float {
    return get(4)
}

/**
 * Returns 5th *element* from the collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component5(): Double {
    return get(4)
}

/**
 * Returns 5th *element* from the collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component5(): Boolean {
    return get(4)
}

/**
 * Returns 5th *element* from the collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component5(): Char {
    return get(4)
}

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun <T> Array<out T>.elementAt(index: Int): T

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun ByteArray.elementAt(index: Int): Byte

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun ShortArray.elementAt(index: Int): Short

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun IntArray.elementAt(index: Int): Int

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun LongArray.elementAt(index: Int): Long

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun FloatArray.elementAt(index: Int): Float

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun DoubleArray.elementAt(index: Int): Double

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun BooleanArray.elementAt(index: Int): Boolean

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun CharArray.elementAt(index: Int): Char

/**
 * Returns an element at the given [index] or throws an
 * [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public expect fun Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T

@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
samples.collections.Collections.Elements.elementAtOrElse

ByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of
 * calling the [defaultValue] function if the [index] is out of bounds of this array.
 */
@kotlin.internal.InlineOnly
public inline fun

ShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Short): Short {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

IntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Int): Int {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

LongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Long): Long {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

CharArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

BooleanArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

ByteArray.elementAtOrNull(index: Int): Byte? {
    return getOrNull(index)
}

ShortArray.elementAtOrNull(index: Int): Short? {
    return getOrNull(index)
}

IntArray.elementAtOrNull(index: Int): Int? {
    return getOrNull(index)
}

LongArray.elementAtOrNull(index: Int): Long? {
    return getOrNull(index)
}

CharArray.elementAtOrNull(index: Int): Char? {
    return getOrNull(index)
}

BooleanArray.elementAtOrNull(index: Int): Boolean? {
    return getOrNull(index)
}

< T> Array<out T>.elementAtOrNull(index: Int): T? {
    return getOrNull(index)
}

ByteArray.elementAtOrNull(index: Int): Byte? {
    return getOrNull(index)
}

ShortArray.elementAtOrNull(index: Int): Short? {
    return getOrNull(index)
}

IntArray.elementAtOrNull(index: Int): Int? {
    return getOrNull(index)
}

LongArray.elementAtOrNull(index: Int): Long? {
    return getOrNull(index)
}

CharArray.elementAtOrNull(index: Int): Char? {
    return getOrNull(index)
}

BooleanArray.elementAtOrNull(index: Int): Boolean? {
    return getOrNull(index)
}
FloatArray.elementAtOrNull(index: Int): Float? { return this.getOrElse(index) } // Returns an element at the given [index] or 'null' if the [index] is out of bounds of this array.

/*
 * Returns an element at the given [index] or 'null' if the [index] is out of bounds of this array.
 */

DoubleArray.elementAtOrNull(index: Int): Double? { return this.getOrElse(index) }

/**
 * Returns an element at the given [index] or 'null' if the [index] is out of bounds of this array.
 */

BooleanArray.elementAtOrNull(index: Int): Boolean? { return this.getOrElse(index) }

/**
 * Returns an element at the given [index] or 'null' if the [index] is out of bounds of this array.
 */

CharArray.elementAtOrNull(index: Int): Char? { return this.getOrElse(index) }

/**
 * Returns the first element matching the given [predicate], or 'null' if no such element was found.
 */

[*@kotlin.internal.InlineOnly*/public inline fun <T> Array<out T>.find(predicate: (T) -> Boolean): T? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun ByteArray.find(predicate: (Byte) -> Boolean): Byte? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun ShortArray.find(predicate: (Short) -> Boolean): Short? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun IntArray.find(predicate: (Int) -> Boolean): Int? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun LongArray.find(predicate: (Long) -> Boolean): Long? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun FloatArray.find(predicate: (Float) -> Boolean): Float? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun DoubleArray.find(predicate: (Double) -> Boolean): Double? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun BooleanArray.find(predicate: (Boolean) -> Boolean): Boolean? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun CharArray.find(predicate: (Char) -> Boolean): Char? { return firstOrNull(predicate) } /* Returns the first element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun Array<out T>.findLast(predicate: (T) -> Boolean): T? { return lastOrNull(predicate) } /* Returns the last element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? { return lastOrNull(predicate) } /* Returns the last element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun ShortArray.findLast(predicate: (Short) -> Boolean): Short? { return lastOrNull(predicate) } /* Returns the last element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun IntArray.findLast(predicate: (Int) -> Boolean): Int? { return lastOrNull(predicate) } /* Returns the last element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun LongArray.findLast(predicate: (Long) -> Boolean): Long? { return lastOrNull(predicate) } /* Returns the last element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun FloatArray.findLast(predicate: (Float) -> Boolean): Float? { return lastOrNull(predicate) } /* Returns the last element matching the given [predicate], or 'null' if no such element was found. */

[*@kotlin.internal.InlineOnly*/public inline fun DoubleArray.findLast(predicate: (Double) -> Boolean): Double? { return lastOrNull(predicate) } /* Returns the last element matching the given [predicate], or 'null' if no such element was found. */
no such element was found.*/

@throws [NoSuchElementException]

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.
 */
public fun BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean?
{
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.
 */
public fun CharArray.findLast(predicate: (Char) -> Boolean): Char?
{
    return lastOrNull(predicate)
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun <T> Array<out T>.first(): T
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun ByteArray.first(): Byte
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun ShortArray.first(): Short
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun IntArray.first(): Int
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun LongArray.first(): Long
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun FloatArray.first(): Float
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun DoubleArray.first(): Double
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun BooleanArray.first(): Boolean
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun CharArray.first(): Char
{
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
element matching the given [predicate].

* @throws [NoSuchElementException] if no such element is found.

* @public inline fun BooleanArray.first(predicate: Boolean) -> Boolean {
    return if (element in this){!
        return returnNotNull(element)
    } else {
        throw NoSuchElementException("Array contains no element matching the

* predicate").
    }
}

/**
 * Returns the first element matching the given [predicate].
 * @throws 

* NoSuchElementException if no such element is found.
 */
public inline fun CharArray.first(predicate: (Char) -> Boolean): Char {
    return if (element in this){!
        return returnNotNull(element)
    } else {
        throw NoSuchElementException("Array contains no element matching the

* predicate").
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 * @throws 

* NoSuchElementException if no such element is found.
 */
public fun <T> Array<out T>.firstOrNull(): T? {
    return if (isEmpty()){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 */
public fun ByteArray.firstOrNull(): Byte? {
    return if (isEmpty()){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 */
public fun ShortArray.firstOrNull(): Short? {
    return if (isEmpty()){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 */
public fun IntArray.firstOrNull(): Int? {
    return if (isEmpty(){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 */
public fun LongArray.firstOrNull(): Long? {
    return if (isEmpty(){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 */
public fun FloatArray.firstOrNull(): Float? {
    return if (isEmpty(){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 */
public fun DoubleArray.firstOrNull(): Double? {
    return if (isEmpty(){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 */
public fun BooleanArray.firstOrNull(): Boolean? {
    return if (isNotEmpty()){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns the first
 * element, or `null` if the array is empty.
 */
public fun CharArray.firstOrNull(): Char? {
    return if (isNotEmpty()){!
        return null
    } else {
        return returnNotNull(element)
    }
}

/**
 * Returns an element at the given 

* [index] or the result of calling the 

* [defaultValue] function if the 

* [index] is out of 

* bounds of this array.

* @param index

* @return

* `null` if element was not found. 

* @throws 

* NoSuchElementException if an exception occurs when calling the 

* [defaultValue] function.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.getOrElse(index: Int, defaultValue: (Int) -> T): T {
    return if (index >= 0 && index <= lastIndex){!
        return get(index)
    } else {
        defaultValue(index)
    }
}

/**
 * Returns an element at the given 

* [index] or the result of calling the 

* [defaultValue] function if the 

* [index] is out of 

* bounds of this array.

* @param index

* @return

* `null` if element was not found. 

* @throws 

* NoSuchElementException if an exception occurs when calling the 

* [defaultValue] function.
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.getOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {
    return if (index >= 0 && index <=
this[index]) { \n    return index
  } \n  return -1\n}

* Returns first index of [element], or -1 if the array does not contain element.*

```kotlin
public fun ShortArray.indexOf(element: Short): Int { \n  for (index in indices) { \n    if (element == this[index]) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
LongArray.indexOf(element: Long): Int { \n  for (index in indices) { \n    if (element == this[index]) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
ByteArray.indexOf(element: Byte): Int { \n  for (index in indices) { \n    if (element == this[index]) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
DoubleArray.indexOf(element: Double): Int { \n  for (index in indices) { \n    if (element == this[index]) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
FloatArray.indexOf(element: Float): Int { \n  for (index in indices) { \n    if (element == this[index]) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
IntArray.indexOf(element: Int): Int { \n  for (index in indices) { \n    if (element == this[index]) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
CharArray.indexOf(element: Char): Int { \n  for (index in indices) { \n    if (element == this[index]) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
BooleanArray.indexOf(element: Boolean): Int { \n  for (index in indices) { \n    if (element == this[index]) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
Int {
    return index
  }
```

```kotlin
Array<out T>.indexOfFirst(predicate: (T) -> Boolean): Int { \n  for (index in indices) { \n    if (predicate(this[index])) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
Array<out T>.indexOfLast(predicate: (T) -> Boolean): Int { \n  for (index in indices.reversed()) { \n    if (predicate(this[index])) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
IntArray.indexOfFirst(predicate: (Int) -> Boolean): Int { \n  for (index in indices) { \n    if (predicate(this[index])) { \n      return index
    } \n  } \n  return -1\n}
```

```kotlin
IntArray.indexOfLast(predicate: (Int) -> Boolean): Int { \n  for (index in indices.reversed()) { \n    if (predicate(this[index])) { \n      return index
    } \n  } \n  return -1\n}
```
ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
ShortArray.indexOfLast(predicate: (Short) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
IntArray.indexOfLast(predicate: (Int) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
LongArray.indexOfLast(predicate: (Long) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
CharArray.indexOfLast(predicate: (Char) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
ArrayOfLast last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ByteArray.last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ShortArray.last: Short { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
IntArray.last: Int { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
LongArray.last: Long { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
FloatArray.last: Float { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
DoubleArray.last: Double { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
CharArray.last: Char { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
BooleanArray.last: Boolean { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
ShortArray.indexOfLast(predicate: (Short) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
IntArray.indexOfLast(predicate: (Int) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
LongArray.indexOfLast(predicate: (Long) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
CharArray.indexOfLast(predicate: (Char) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
ArrayOfLast last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ByteArray.last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ShortArray.last: Short { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
IntArray.last: Int { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
LongArray.last: Long { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
FloatArray.last: Float { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
DoubleArray.last: Double { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
CharArray.last: Char { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
ShortArray.indexOfLast(predicate: (Short) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
IntArray.indexOfLast(predicate: (Int) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
LongArray.indexOfLast(predicate: (Long) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
CharArray.indexOfLast(predicate: (Char) -> Boolean): Int { for (index in indices.reversed()) { if (predicate(this[index])) { return index } } return -1
} *\*\n
ArrayOfLast last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ByteArray.last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ShortArray.last: Short { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
IntArray.last: Int { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
LongArray.last: Long { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
FloatArray.last: Float { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
DoubleArray.last: Double { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
CharArray.last: Char { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ArrayOfLast last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ByteArray.last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ShortArray.last: Short { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
IntArray.last: Int { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
LongArray.last: Long { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
FloatArray.last: Float { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
DoubleArray.last: Double { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
CharArray.last: Char { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[lastIndex]
} *\*\n
ArrayOfLast last: Byte { if (isEmpty()) { throw NoSuchElementException("Array is empty.") } return this[}
[NoSuchElementException] if no such element is found.\n
```kotlin
public inline fun ByteArray.last(predicate: (Byte) -> Boolean): Byte {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
```

```kotlin
public inline fun ShortArray.last(predicate: (Short) -> Boolean): Short {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
```

```kotlin
public inline fun IntArray.last(predicate: (Int) -> Boolean): Int {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
```

```kotlin
public inline fun LongArray.last(predicate: (Long) -> Boolean): Long {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
```

```kotlin
public inline fun FloatArray.last(predicate: (Float) -> Boolean): Float {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
```

```kotlin
public inline fun DoubleArray.last(predicate: (Double) -> Boolean): Double {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
```

```kotlin
public inline fun BooleanArray.last(predicate: (Boolean) -> Boolean): Boolean {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
```

```kotlin
public inline fun CharArray.last(predicate: (Char) -> Boolean): Char {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
```

```kotlin
public fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.lastIndexOf(element: T): Int {
    if (element == null) {
        for (index in indices.reversed()) {
            if (this[index] == null) {
                return index
            }
        }
    } else {
        for (index in indices.reversed()) {
            if (element == this[index]) {
                return index
            }
        }
    }
    return -1
}
```

```kotlin
public fun ByteArray.lastIndexOf(element: Byte): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun ShortArray.lastIndexOf(element: Short): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun IntArray.lastIndexOf(element: Int): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun LongArray.lastIndexOf(element: Long): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun FloatArray.lastIndexOf(element: Float): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun DoubleArray.lastIndexOf(element: Double): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun BooleanArray.lastIndexOf(element: Boolean): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun CharArray.lastIndexOf(element: Char): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```
* Returns last index of [element], or -1 if the array does not contain element.

```kotlin
public fun DoubleArray.lastIndexOf(element: Double): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun BooleanArray.lastIndexOf(element: Boolean): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun DoubleArray.lastOrNull(predicate: (Double) -> Boolean): Double? {
    for (index in indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}
```

```kotlin
public fun BooleanArray.lastOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    for (index in indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}
```

```kotlin
public fun IntArray.lastOrNull(): Int? {
    return if (isEmpty()) null else this[size - 1]
}
```

```kotlin
public fun LongArray.lastOrNull(): Long? {
    return if (isEmpty()) null else this[size - 1]
}
```

```kotlin
public fun ByteArray.lastOrNull(): Byte? {
    return if (isEmpty()) null else this[size - 1]
}
```

```kotlin
public fun <T> Array<out T>.lastOrNull(): T? {
    return if (isEmpty()) null else this[size - 1]
}
```

```kotlin
public fun CharArray.lastIndexOf(element: Char): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun ByteArray.lastIndexOf(element: Byte): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun CharArray.lastIndexOf(element: Char): Int {
    for (index in indices.reversed()) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

```kotlin
public fun IntArray.lastIndexOf(): Int? {
    return if (isEmpty()) null else this[size - 1]
}
```

```kotlin
public fun LongArray.lastIndexOf(): Long? {
    return if (isEmpty()) null else this[size - 1]
}
```

```kotlin
public fun ByteArray.lastIndexOf(): Byte? {
    return if (isEmpty()) null else this[size - 1]
}
```

```kotlin
public fun <T> Array<out T>.lastOrNull(): T? {
    return if (isEmpty()) null else this[size - 1]
}
```
random element from this array.

```kotlin
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random)
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun ByteArray.random(): Byte {
    return random(Random)
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun ShortArray.random(): Short {
    return random(Random)
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun IntArray.random(): Int {
    return random(Random)
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun LongArray.random(): Long {
    return random(Random)
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun FloatArray.random(): Float {
    return random(Random)
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.random(): Double {
    return random(Random)
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun BooleanArray.random(): Boolean {
    return random(Random)
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun CharArray.random(): Char {
    return random(Random)
}
```

```kotlin
public fun <T> Array<out T>.random(random: Random): T {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun ByteArray.random(random: Random): Byte {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun ShortArray.random(random: Random): Short {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun IntArray.random(random: Random): Int {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun LongArray.random(random: Random): Long {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun FloatArray.random(random: Random): Float {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun DoubleArray.random(random: Random): Double {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
public fun CharArray.random(): Char {
    return random(Random)
}
```

```kotlin
public fun ShortArray.random(random: Random): Short {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
public fun IntArray.random(random: Random): Int {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
public fun LongArray.random(random: Random): Long {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
public fun FloatArray.random(random: Random): Float {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
public fun DoubleArray.random(random: Random): Double {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
public fun CharArray.random(): Char {
    return random(Random)
}
```
this array is empty.

```kotlin
public fun BooleanArray.random(random: Random): Boolean {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
@SinceKotlin("1.3")
public fun BooleanArray.random(random: Random): Boolean {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
@SinceKotlin("1.3")
public fun CharArray.random(random: Random): Char {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
@SinceKotlin("1.3")
public fun IntArray.random(random: Random): Int {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
@SinceKotlin("1.3")
public fun LongArray.random(random: Random): Long {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
@SinceKotlin("1.3")
public fun FloatArray.random(random: Random): Float {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
@SinceKotlin("1.3")
public fun DoubleArray.random(random: Random): Double {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
```

```kotlin
@SinceKotlin("1.3")
public fun BooleanArray.single(): Boolean {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun CharArray.single(): Char {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public inline fun <T> Array<out T>.single(predicate: (T) -> Boolean): T {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found)
                throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found)
        throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as T
}
```

```kotlin
@SinceKotlin("1.3")
public fun ByteArray.single(): Byte {
    var single: Byte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found)
                throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found)
        throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Byte
}
```

```kotlin
@SinceKotlin("1.3")
public fun IntArray.single(): Int {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun LongArray.single(): Long {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun FloatArray.single(): Float {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun DoubleArray.single(): Double {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun BooleanArray.single(): Boolean {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun CharArray.single(): Char {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun IntArray.single(): Int {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun LongArray.single(): Long {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun FloatArray.single(): Float {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun DoubleArray.single(): Double {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun BooleanArray.single(): Boolean {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun CharArray.single(): Char {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun IntArray.single(): Int {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun LongArray.single(): Long {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun FloatArray.single(): Float {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun DoubleArray.single(): Double {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun BooleanArray.single(): Boolean {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun CharArray.single(): Char {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun IntArray.single(): Int {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun LongArray.single(): Long {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun FloatArray.single(): Float {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```

```kotlin
@SinceKotlin("1.3")
public fun DoubleArray.single(): Double {
    if (size == 0)
        throw NoSuchElementException("Array is empty.")
    return this[0]
}
```
public inline fun ShortArray.single(predicate: (Short) -> Boolean): Short {
    var single: Short? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.
" + single + = element
    found = true
    } else if (!found) throw NoSuchElementException("Array contains no element matching the predicate.
" + @Suppress("UNCHECKED_CAST")
    return single as Short
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */

public inline fun IntArray.single(predicate: (Int) -> Boolean): Int {
    var single: Int? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.
" + single + = element
    found = true
    } else if (!found) throw NoSuchElementException("Array contains no element matching the predicate.
" + @Suppress("UNCHECKED_CAST")
    return single as Int
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */

public inline fun LongArray.single(predicate: (Long) -> Boolean): Long {
    var single: Long? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.
" + single + = element
    found = true
    } else if (!found) throw NoSuchElementException("Array contains no element matching the predicate.
" + @Suppress("UNCHECKED_CAST")
    return single as Long
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */

public inline fun FloatArray.single(predicate: (Float) -> Boolean): Float {
    var single: Float? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.
" + single + = element
    found = true
    } else if (!found) throw NoSuchElementException("Array contains no element matching the predicate.
" + @Suppress("UNCHECKED_CAST")
    return single as Float
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */

public inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {
    var single: Double? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.
" + single + = element
    found = true
    } else if (!found) throw NoSuchElementException("Array contains no element matching the predicate.
" + @Suppress("UNCHECKED_CAST")
    return single as Double
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */

public inline fun BooleanArray.single(predicate: (Boolean) -> Boolean): Boolean {
    var single: Boolean? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.
" + single + = element
    found = true
    } else if (!found) throw NoSuchElementException("Array contains no element matching the predicate.
" + @Suppress("UNCHECKED_CAST")
    return single as Boolean
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */

public fun <T> Array<out T>.singleOrNull(): T? {
    return if (size == 1) this[0] else null
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */

public fun ByteArray.singleOrNull(): Byte? {
    return if (size == 1) this[0] else null
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}
'null' if the array is empty or has more than one element.

public fun LongArray.singleOrNull(): Long? {
    return if (size == 1) this[0] else null
}

public fun FloatArray.singleOrNull(): Float? {
    return if (size == 1) this[0] else null
}

public fun DoubleArray.singleOrNull(): Double? {
    return if (size == 1) this[0] else null
}

public fun BooleanArray.singleOrNull(): Boolean? {
    return if (size == 1) this[0] else null
}

public fun CharArray.singleOrNull(): Char? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun ByteArray.singleOrNull(): Byte? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun IntArray.singleOrNull(): Int? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
samples.collections.Collections.Transformations.drop

*/
public fun <T> Array<out T>.drop(n: Int): List<T> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ByteArray.drop(n: Int): List<Byte> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ShortArray.drop(n: Int): List<Short> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun IntArray.drop(n: Int): List<Int> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun LongArray.drop(n: Int): List<Long> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun FloatArray.drop(n: Int): List<Float> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun DoubleArray.drop(n: Int): List<Double> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun BooleanArray.drop(n: Int): List<Boolean> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun CharArray.drop(n: Int): List<Char> {
    return takeLast((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun <T> Array<out T>.dropLast(n: Int): List<T> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ByteArray.dropLast(n: Int): List<Byte> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ShortArray.dropLast(n: Int): List<Short> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun IntArray.dropLast(n: Int): List<Int> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun LongArray.dropLast(n: Int): List<Long> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun FloatArray.dropLast(n: Int): List<Float> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun DoubleArray.dropLast(n: Int): List<Double> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun BooleanArray.dropLast(n: Int): List<Boolean> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun CharArray.dropLast(n: Int): List<Char> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun <T> Array<out T>.dropLast(n: Int): List<T> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ByteArray.dropLast(n: Int): List<Byte> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ShortArray.dropLast(n: Int): List<Short> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun IntArray.dropLast(n: Int): List<Int> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun LongArray.dropLast(n: Int): List<Long> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun FloatArray.dropLast(n: Int): List<Float> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun DoubleArray.dropLast(n: Int): List<Double> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun BooleanArray.dropLast(n: Int): List<Boolean> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun CharArray.dropLast(n: Int): List<Char> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun <T> Array<out T>.dropLast(n: Int): List<T> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ByteArray.dropLast(n: Int): List<Byte> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ShortArray.dropLast(n: Int): List<Short> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun IntArray.dropLast(n: Int): List<Int> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun LongArray.dropLast(n: Int): List<Long> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun FloatArray.dropLast(n: Int): List<Float> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun DoubleArray.dropLast(n: Int): List<Double> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun BooleanArray.dropLast(n: Int): List<Boolean> {
    return take((size - n).coerceAtLeast(n))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun CharArray.dropLast(n: Int): List<Char> {
    return take((size - n).coerceAtLeast(n))
}
samples.collections.Collections.Transformations.dropIn
*/
public fun IntArray.dropLast(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0) to n) //n/***/n * Returns a list containing all elements except last [n] elements.
} * @throws IllegalArgumentException if [n] is negative.

samples.collections.Collections.Transformations.dropIn
*/
public fun LongArray.dropLast(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0) to n) //n/***/n * Returns a list containing all elements except last [n] elements.
} * @throws IllegalArgumentException if [n] is negative.

samples.collections.Collections.Transformations.dropIn
*/
public fun FloatArray.dropLast(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0) to n) //n/***/n * Returns a list containing all elements except last [n] elements.
} * @throws IllegalArgumentException if [n] is negative.

samples.collections.Collections.Transformations.dropIn
*/
public fun DoubleArray.dropLast(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0) to n) //n/***/n * Returns a list containing all elements except last [n] elements.
} * @throws IllegalArgumentException if [n] is negative.

samples.collections.Collections.Transformations.dropIn
*/
public fun BooleanArray.dropLast(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0) to n) //n/***/n * Returns a list containing all elements except last [n] elements.
} * @throws IllegalArgumentException if [n] is negative.

samples.collections.Collections.Transformations.dropIn
*/
public fun CharArray.dropLast(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0) to n) //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
} * @throws IllegalArgumentException if [n] is negative.

samples.collections.Collections.Transformations.dropIn
*/
public fun ByteArray.dropLastWhile(predicate: (Byte) -> Boolean): List<Byte> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1) //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
        }
    }
    return emptyList() //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
} * @throws IllegalArgumentException if [predicate] is null.

samples.collections.Collections.Transformations.dropIn
*/
public fun ShortArray.dropLastWhile(predicate: (Short) -> Boolean): List<Short> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1) //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
        }
    }
    return emptyList() //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
} * @throws IllegalArgumentException if [predicate] is null.

samples.collections.Collections.Transformations.dropIn
*/
public fun IntArray.dropLastWhile(predicate: (Int) -> Boolean): List<Int> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1) //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
        }
    }
    return emptyList() //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
} * @throws IllegalArgumentException if [predicate] is null.

samples.collections.Collections.Transformations.dropIn
*/
public fun LongArray.dropLastWhile(predicate: (Long) -> Boolean): List<Long> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1) //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
        }
    }
    return emptyList() //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
} * @throws IllegalArgumentException if [predicate] is null.

samples.collections.Collections.Transformations.dropIn
*/
public fun FloatArray.dropLastWhile(predicate: (Float) -> Boolean): List<Float> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1) //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
        }
    }
    return emptyList() //n/***/n * Returns a list containing all elements except last elements that satisfy the given [predicate].
} * @throws IllegalArgumentException if [predicate] is null.
(Double) -> Boolean): List<Double> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun BooleanArray.dropLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun CharArray.dropLastWhile(predicate: (Char) -> Boolean): List<Char> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except first elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun <T> Array<out T>.dropWhile(predicate: (T) -> Boolean): List<T> {
    var yielding = false
    val list = ArrayList<T>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun ByteArray.dropWhile(predicate: (Byte) -> Boolean): List<Byte> {
    var yielding = false
    val list = ArrayList<Byte>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun ShortArray.dropWhile(predicate: (Short) -> Boolean): List<Short> {
    var yielding = false
    val list = ArrayList<Short>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {
    var yielding = false
    val list = ArrayList<Int>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun LongArray.dropWhile(predicate: (Long) -> Boolean): List<Long> {
    var yielding = false
    val list = ArrayList<Long>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {
    var yielding = false
    val list = ArrayList<Float>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given
 * predicate.
 * @sample samples.collections.Collections.Transformations.drop
 */
public inline fun DoubleArray.dropWhile(predicate: (Double) -> Boolean): List<Double> {
    var yielding = false
    val list = ArrayList<Double>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}
(Char) -> Boolean): List<Char> {\n    var yielding = false\n    val list = ArrayList<Char>()\n    for (item in this)\n    if (yielding)\n        list.add(item)\n    else if (!predicate(item)) {\n        list.add(item)\n        yielding = true\n    }\n    return list\n}
BooleanArray.filterIndexed(predicate: (index: Int, Boolean) -> Boolean): List<Boolean>

filterIndexedTo(ArrayList<Boolean>(), predicate)

* @param [predicate] function that takes the index of an element and the element itself

* and returns the result of predicate evaluation on the element.


*/

public inline fun

CharArray.filterIndexed(predicate: (index: Int, Char) -> Boolean): List<Char>

filterIndexedTo(ArrayList<Char>(), predicate)

*/

public inline fun <T, C : MutableCollection<T>> Array<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

public inline fun <C : MutableCollection<in Byte>>

ByteArray.filterIndexedTo(destination: C, predicate: (index: Int, Byte) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

public inline fun <C : MutableCollection<in Short>>

ShortArray.filterIndexedTo(destination: C, predicate: (index: Int, Short) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

public inline fun <C : MutableCollection<in Int>>

IntArray.filterIndexedTo(destination: C, predicate: (index: Int, Int) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

public inline fun <C : MutableCollection<in Long>>

LongArray.filterIndexedTo(destination: C, predicate: (index: Int, Long) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

public inline fun <C : MutableCollection<in Float>>

FloatArray.filterIndexedTo(destination: C, predicate: (index: Int, Float) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

public inline fun <C : MutableCollection<in Double>>

DoubleArray.filterIndexedTo(destination: C, predicate: (index: Int, Double) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

public inline fun 

CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/

public inline fun 

ByteArray.filterIndexedTo(destination: C, predicate: (index: Int, Byte) -> Boolean): C

forEachIndexed {

(index, element) ->

if (predicate(index, element)) destination.add(element)

} return destination

*/
/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun <T> Array<out T>.filterNotTo(destination: Collection<in T>, predicate: (T) -> Boolean): Collection<in T> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun ByteArray.filterNotTo(destination: Collection<in Byte>, predicate: (Byte) -> Boolean): Collection<in Byte> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun ShortArray.filterNotTo(destination: Collection<in Short>, predicate: (Short) -> Boolean): Collection<in Short> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun IntArray.filterNotTo(destination: Collection<in Int>, predicate: (Int) -> Boolean): Collection<in Int> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun LongArray.filterNotTo(destination: Collection<in Long>, predicate: (Long) -> Boolean): Collection<in Long> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun FloatArray.filterNotTo(destination: Collection<in Float>, predicate: (Float) -> Boolean): Collection<in Float> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun DoubleArray.filterNotTo(destination: Collection<in Double>, predicate: (Double) -> Boolean): Collection<in Double> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun BooleanArray.filterNotTo(destination: Collection<in Boolean>, predicate: (Boolean) -> Boolean): Collection<in Boolean> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
public inline fun CharArray.filterNotTo(destination: Collection<in Char>, predicate: (Char) -> Boolean): Collection<in Char> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements that are not null.
 */
public fun <T : Any> Array<out T?>.filterNotNull(): Collection<T> {
    return filterNotNullTo(ArrayList<T>())
}

/**
 * Appends all elements that are not `null` to the given [destination].
 */
public fun <C : MutableCollection<in T>, T : Any> Array<out T?>.filterNotNullTo(destination: C): C {
    for (element in this) if (element != null) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate] to the given [destination].
 */
public fun <T> Array<out T>.filterNotTo(destination: Collection<in T>, predicate: (T) -> Boolean): Collection<in T> {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in T>, T : Any> Array<out T?>.filterNotNullTo(destination: C): C {
    for (element in this) if (element != null) destination.add(element)
    return destination
}
Double>> DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public inline fun <C : MutableCollection<in Double>> BooleanArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Char> CharArray.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Byte> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Short> ShortArray.filterTo(destination: C, predicate: (Short) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Int> IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Long> LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Float> FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Double> DoubleArray.filterTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Boolean> BooleanArray.filterTo(destination: C, predicate: (Boolean) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Char> CharArray.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Byte> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Short> ShortArray.filterTo(destination: C, predicate: (Short) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Int> IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Long> LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Float> FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Double> DoubleArray.filterTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Boolean> BooleanArray.filterTo(destination: C, predicate: (Boolean) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Char> CharArray.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Byte> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Short> ShortArray.filterTo(destination: C, predicate: (Short) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Int> IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Long> LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <Float> FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun <T> Array<T>.slice(indices: IntRange): List<T> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun ByteArray.slice(indices: IntRange): List<Byte> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun ShortArray.slice(indices: IntRange): List<Short> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun IntArray.slice(indices: IntRange): List<Int> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun LongArray.slice(indices: IntRange): List<Long> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun FloatArray.slice(indices: IntRange): List<Float> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun DoubleArray.slice(indices: IntRange): List<Double> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun BooleanArray.slice(indices: IntRange): List<Boolean> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun CharArray.slice(indices: IntRange): List<Char> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun ByteArray.slice(indices: IntRange): List<Byte> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun CharArray.slice(indices: IntRange): List<Char> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}
```java
IntArray.sliceArray(indices: Collection<Int>): IntArray {
    val result = IntArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

/**
 * Returns an array containing elements at specified [indices].
 */
public fun IntArray.slice(indices: Collection<Int>): IntArray {
    val result = IntArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

ByteArray.sliceArray(indices: Collection<Int>): ByteArray {
    val result = ByteArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

Array<T>.sliceArray(indices: Collection<Int>): Array<T> {
    val result = arrayOfNulls(this, indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

ArrayList<Char>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Char> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Char>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Boolean> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Boolean>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Double> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Double>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Float> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Float>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Long> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Long>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Short> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Short>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Byte> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Byte>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<T> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<T>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

Array<out T>.slice(indices: Iterable<Int>): List<T> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

CharArray.slice(indices: Iterable<Int>): List<Char> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

IntArray.sliceArray(indices: Collection<Int>): IntArray {
    val result = IntArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

ByteArray.sliceArray(indices: Collection<Int>): ByteArray {
    val result = ByteArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

Array<T>.sliceArray(indices: Collection<Int>): List<T> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

IntArray.sliceArray(indices: Collection<Int>): IntArray {
    val result = IntArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

ByteArray.sliceArray(indices: Collection<Int>): ByteArray {
    val result = ByteArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

Array<T>.sliceArray(indices: Collection<Int>): List<T> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

IntArray.sliceArray(indices: Collection<Int>): IntArray {
    val result = IntArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}
```

Open Source Used In DNAC Platform Antelope 3755
a list containing first [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun ShortArray.take(n: Int): List<Short> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Short>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing first [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun IntArray.take(n: Int): List<Int> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Int>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing first [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun FloatArray.take(n: Int): List<Float> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Float>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing first [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun LongArray.take(n: Int): List<Long> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Long>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing first [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun BooleanArray.take(n: Int): List<Boolean> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Boolean>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun CharArray.take(n: Int): List<Character> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Char>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun <T> Array<out T>.take(n: Int): List<T> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<T>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun CharArray.take(n: Int): List<Character> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Char>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun ByteArray.take(n: Int): List<Byte> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Byte>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun FloatArray.take(n: Int): List<Float> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Float>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun LongArray.take(n: Int): List<Long> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Long>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun BooleanArray.take(n: Int): List<Boolean> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Boolean>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun CharArray.take(n: Int): List<Character> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Char>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.

* @throws IllegalArgumentException if [n] is negative.

* @sample

samples.collections.Collections.Transformations.take

*<public fun ByteArray.take(n: Int): List<Byte> {

\n require(n >= 0) { "Requested element count \$n is less than zero." }\n if (n == 0) return emptyList()\n if (n > size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n var list = ArrayList<Byte>(n)\n for (item in this) {\n     list.add(item)\n     if (++count == n) break\n }\n return list\n}\n
* Returns a list containing last [n] elements.
Returns a list containing last [n] elements.
 * @throws IllegalArgumentExpection if [n] is negative.

```kotlin
public fun <T> Array<out T>.takeLast(n: Int): List<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<T>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

*public fun IntArray.takeLast(n: Int): List<Int>*

```kotlin
public fun IntArray.takeLast(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<Int>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

*public fun ShortArray.takeLast(n: Int): List<Short>*

```kotlin
public fun ShortArray.takeLast(n: Int): List<Short> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<Short>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

*public fun ByteArray.takeLast(n: Int): List<Byte>*

```kotlin
public fun ByteArray.takeLast(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<Byte>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

*public fun CharArray.takeLast(n: Int): List<Char>*

```kotlin
public fun CharArray.takeLast(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<Char>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

*public fun BooleanArray.takeLast(n: Int): List<Boolean>*

```kotlin
public fun BooleanArray.takeLast(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<Boolean>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

*public fun DoubleArray.takeLast(n: Int): List<Double>*

```kotlin
public fun DoubleArray.takeLast(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<Double>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

*public fun FloatArray.takeLast(n: Int): List<Float>*

```kotlin
public fun FloatArray.takeLast(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<Float>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

*public fun LongArray.takeLast(n: Int): List<Long>*

```kotlin
public fun LongArray.takeLast(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<Long>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```
given [predicate].

* Returns a list containing last elements satisfying the given [predicate].

* @sample samples.collections.Collections.Transformations.take

```kotlin
public inline fun LongArray.takeLastWhile(predicate: (Long) -> Boolean): List<Long> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun FloatArray.takeLastWhile(predicate: (Float) -> Boolean): List<Float> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun DoubleArray.takeLastWhile(predicate: (Double) -> Boolean): List<Double> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun BooleanArray.takeLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun CharArray.takeLastWhile(predicate: (Char) -> Boolean): List<Character> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}
```

```kotlin
public inline fun ByteArray.takeWhile(predicate: (Byte) -> Boolean): List<Byte> {
    val list = ArrayList<Byte>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun ShortArray.takeWhile(predicate: (Short) -> Boolean): List<Short> {
    val list = ArrayList<Short>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun IntArray.takeWhile(predicate: (Int) -> Boolean): List<Int> {
    val list = ArrayList<Int>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun LongArray.takeWhile(predicate: (Long) -> Boolean): List<Long> {
    val list = ArrayList<Long>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}
```

```kotlin
public inline fun FloatArray.takeWhile(predicate: (Float) -> Boolean): List<Float> {
    val list = ArrayList<Float>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun DoubleArray.takeWhile(predicate: (Double) -> Boolean): List<Double> {
    val list = ArrayList<Double>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}
```

```kotlin
public inline fun BooleanArray.takeWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    val list = ArrayList<Boolean>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun CharArray.takeWhile(predicate: (Char) -> Boolean): List<Character> {
    val list = ArrayList<Character>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}
```
fun <T> Array<T>.reversedArray(): Array<T> {
    if (isEmpty()) return this
    val result = arrayOfNulls<T>(this.size)
    for (i in 0..<size) {
        result[i] = this[size - 1 - i]
    }
    return result
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun CharArray.reversed(): List<Char> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun DoubleArray.reversed(): List<Double> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun FloatArray.reversed(): List<Float> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun LongArray.reversed(): List<Long> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun IntArray.reversed(): List<Int> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun ByteArray.reversed(): List<Byte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun <T> Array<out T>.reversed(): List<T> {
    if (isEmpty()) return emptyList()
    val result = arrayOfNulls<T>(this.size)
    val midPoint = size / 2
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
    return result
}

/**
 * Reverses elements in the array in-place.
 */
public fun BooleanArray.reverse(): Unit {
    val midPoint = size / 2
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Reverses elements in the array in-place.
 */
public fun CharArray.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Reverses elements in the array in-place.
 */
public fun DoubleArray.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Reverses elements in the array in-place.
 */
public fun FloatArray.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Reverses elements in the array in-place.
 */
public fun LongArray.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Reverses elements in the array in-place.
 */
public fun IntArray.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Reverses elements in the array in-place.
 */
public fun ShortArray.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Reverses elements in the array in-place.
 */
public fun ByteArray.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Returns a list with elements in reversed order.
 */
public fun <T> List<T>.reversed(): List<T> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun ArrayList<T>().for (item in this) {
    if (!predicate(item))
        break
    list.add(item)
}

/**
 * Returns a list with elements in reversed order.
 */
public fun <T, T> Array<T>.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}

/**
 * Returns a list with elements in reversed order.
 */
public fun <T, T> Array<T>.reversed(): List<T> {
    if (isEmpty()) return emptyList()
    var reverseIndex = lastIndex
    for (index in 0..<midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex - index]
        reverseIndex--
    }
}
size
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun
ByteArray.reversedArray(): ByteArray
{ if (isEmpty()) return this
val result = ByteArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun ShortArray.reversedArray(): ShortArray
{ if (isEmpty()) return this
val result = ShortArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun IntArray.reversedArray(): IntArray
{ if (isEmpty()) return this
val result = IntArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun LongArray.reversedArray(): LongArray
{ if (isEmpty()) return this
val result = LongArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun FloatArray.reversedArray(): FloatArray
{ if (isEmpty()) return this
val result = FloatArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun BooleanArray.reversedArray(): BooleanArray
{ if (isEmpty()) return this
val result = BooleanArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun CharArray.reversedArray(): CharArray
{ if (isEmpty()) return this
val result = CharArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Stable. It means that equal elements preserve their order relative to each other after sorting.

/*
public inline fun <T, R : Comparable<R>> Array<out T>.sortByDescending(crossinline selector: (T) -> R?): Unit
{ if (size > 1) sortWith(compareBy(selector))

* Returns elements in the array in-place according to natural sort order of the value returned by specified [selector] function.

* 
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

/*
public inline fun <T, R : Comparable<R>> Array<out T>.sortBy(crossinline selector: (T) -> R?): Unit
{ if (size > 1) sortWith(compareBy(selector))

* Returns elements in the array in-place according to natural sort order of the value returned by specified [selector] function.

* 
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

/*
public fun ShortArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun FloatArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun DoubleArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun Array<out T>.sortByDescending(crossinline selector: (T) -> R?): Unit
{ if (size > 1) sortWith(compareBy(selector))

* Returns elements in the array in-place according to natural sort order of the value returned by specified [selector] function.

* 
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

/*
public fun Array<out T>.sortBy(crossinline selector: (T) -> R?): Unit
{ if (size > 1) sortWith(compareBy(selector))

* Returns elements in the array in-place according to natural sort order of the value returned by specified [selector] function.

* 
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

/*
public fun IntArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun LongArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun CharArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun BooleanArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun ByteArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun DoubleArray.sortDescending(): Unit
{ if (size > 1) { sort() reverse() }

* Returns elements in the array in-place according to their natural sort order.

/*
public fun Array<out T>.sortDescending(): Unit
{ if (size > 1) sort() lastIndex

for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns elements in the array in-place according to their natural sort order.

/*
public fun IntArray.reversedArray(): IntArray
{ if (isEmpty()) return this
val result = IntArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun ByteArray.reversedArray(): ByteArray
{ if (isEmpty()) return this
val result = ByteArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun LongArray.reversedArray(): LongArray
{ if (isEmpty()) return this
val result = LongArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun ShortArray.reversedArray(): ShortArray
{ if (isEmpty()) return this
val result = ShortArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun CharArray.reversedArray(): CharArray
{ if (isEmpty()) return this
val result = CharArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun DoubleArray.reversedArray(): DoubleArray
{ if (isEmpty()) return this
val result = DoubleArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun IntArray.reversedArray(): IntArray
{ if (isEmpty()) return this
val result = IntArray(size)
val lastIndex = lastIndex
for (i in 0..lastIndex)
result[lastIndex - i] = this[i]
return result

* Returns an array with elements of this array in reversed order.

/*
public fun Array<out T>.sortByDescending(crossinline selector: (T) -> R?): Unit
{ if (size > 1) sortWith(compareByDescending(selector))

* Returns elements in the array in-place according to natural sort order of the value returned by specified [selector] function.

* 
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

/*
public fun Array<out T>.sortByDescending(crossinline selector: (T) -> R?): Unit
{ if (size > 1) sortWith(compareByDescending(selector))

* Returns elements in the array in-place according to natural sort order of the value returned by specified [selector] function.

* 
* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
Array<out T>.sorted(): List<T> { return sortedArray().asList().toList() } Returns a list of all elements sorted according to their natural sort order.

*public fun ByteArray.sorted(): List<Byte> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun DoubleArray.sorted(): List<Double> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun FloatArray.sorted(): List<Float> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun IntArray.sorted(): List<Int> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun LongArray.sorted(): List<Long> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ShortArray.sorted(): List<Short> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ByteArray.sorted(): List<Byte> { return toTypedArray().apply { sortWith(reverseOrder()) }.asList() } Adds all elements sorted according to their natural sort order.

*public fun DoubleArray.sorted(): List<Double> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun FloatArray.sorted(): List<Float> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun IntArray.sorted(): List<Int> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun LongArray.sorted(): List<Long> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ShortArray.sorted(): List<Short> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ByteArray.sorted(): List<Byte> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun DoubleArray.sorted(): List<Double> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun FloatArray.sorted(): List<Float> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun IntArray.sorted(): List<Int> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun LongArray.sorted(): List<Long> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ShortArray.sorted(): List<Short> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ByteArray.sorted(): List<Byte> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun DoubleArray.sorted(): List<Double> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun FloatArray.sorted(): List<Float> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun IntArray.sorted(): List<Int> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun LongArray.sorted(): List<Long> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ShortArray.sorted(): List<Short> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ByteArray.sorted(): List<Byte> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun DoubleArray.sorted(): List<Double> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun FloatArray.sorted(): List<Float> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun IntArray.sorted(): List<Int> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun LongArray.sorted(): List<Long> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ShortArray.sorted(): List<Short> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ByteArray.sorted(): List<Byte> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun DoubleArray.sorted(): List<Double> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun FloatArray.sorted(): List<Float> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun IntArray.sorted(): List<Int> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun LongArray.sorted(): List<Long> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ShortArray.sorted(): List<Short> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun ByteArray.sorted(): List<Byte> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun DoubleArray.sorted(): List<Double> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun FloatArray.sorted(): List<Float> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.

*public fun IntArray.sorted(): List<Int> { return toTypedArray().apply { sort() }.asList() } Adds all elements sorted according to their natural sort order.
this array sorted descending according to their natural sort order.

```kotlin
CharArray.sortedArrayDescending(): CharArray

    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }

* Returns an array with all elements of this array sorted according to the specified [comparator].

    The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

    Examples:
        *<T> Array<out T>.sortedArrayWith(comparator: Comparator<in T>): Array<out T>
          { if (isEmpty()) return this
          return this.copyOf().apply { sortWith(comparator) }

* Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.

    The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

*<T> Array<out T>.sortedBy(crossinline selector: (T) -> R?): List<T>

*<R> Array<out R>.sortedByDescending(crossinline selector: (R) -> T?): List<R>

*<T> Array<out T>.sortedWith(compareBy(selector: (T) -> R?): Comparator<in T>)

*<R> Array<out R>.sortedWith(compareByDescending(selector: (R) -> T?): Comparator<in R>)
```

---

**Open Source Used in DNAC Platform Antelope 3763**

---
For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. The elements of other types are compared for equality with the

```kotlin
@SinceKotlin("1.1")
public expect infix fun BooleanArray.contentDeepEquals(other: BooleanArray): Boolean
```

```kotlin
@SinceKotlin("1.1")
public expect infix fun ByteArray.contentDeepEquals(other: ByteArray): Boolean
```

```kotlin
@SinceKotlin("1.1")
public expect infix fun <T> Array<out T>.contentDeepEquals(other: Array<out T>): Boolean
```

```kotlin
@SinceKotlin("1.1")
public expect fun <T> Array<out T>.contentDeepHashCode(): Int
```

```kotlin
@SinceKotlin("1.1")
public expect fun <T> Array<out T>.contentDeepToString(): String
```

```kotlin
@SinceKotlin("1.1")
public expect fun CharArray.asList(): List<Char>
```

```kotlin
@SinceKotlin("1.1")
public expect fun BooleanArray.asList(): List<Boolean>
```

```kotlin
@SinceKotlin("1.1")
public expect fun DoubleArray.asList(): List<Double>
```

```kotlin
@SinceKotlin("1.1")
public expect fun FloatArray.asList(): List<Float>
```

```kotlin
@SinceKotlin("1.1")
public expect fun IntArray.asList(): List<Int>
```

```kotlin
@SinceKotlin("1.1")
public expect fun LongArray.asList(): List<Long>
```

```kotlin
@SinceKotlin("1.1")
public expect fun ShortArray.asList(): List<Short>
```

```kotlin
@SinceKotlin("1.1")
public expect fun ByteArray.asList(): List<Byte>
```

```kotlin
@SinceKotlin("1.1")
public expect fun <T> Array<out T>.asList(): List<T>
```

```kotlin
Array<out T>: Boolean
```

```kotlin
Array<out T>: Int
```

```kotlin
Array<out T>: Long
```

```kotlin
Array<out T>: Short
```

```kotlin
Array<out T>: Byte
```

```kotlin
Array<out T>: Char
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```

```kotlin
Comparator<in Boolean>: List<Char>
```

```kotlin
Comparator<in Boolean>: List<Byte>
```

```kotlin
Comparator<in Boolean>: List<Short>
```

```kotlin
Comparator<in Boolean>: List<Int>
```

```kotlin
Comparator<in Boolean>: List<Long>
```

```kotlin
Comparator<in Boolean>: List<Double>
```

```kotlin
Comparator<in Boolean>: List<Float>
```
equal to `0.0'.

public expect infix fun FloatArray.contentEquals(other: FloatArray):
Boolean

* Returns `true' if the two specified arrays are *structurally* equal to one another,
* i.e. contain the same number of the same elements in the same order.
* The elements are compared for equality with the
[equals] function.
* For floating point numbers it means that `NaN' is equal to itself and `-0.0' is not equal to `0.0'.

public expect infix fun DoubleArray.contentEquals(other: DoubleArray):
Boolean

* Returns `true' if the two specified arrays are *structurally* equal to one another,
* i.e. contain the same number of the same elements in the same order.
* The elements are compared for equality with the
[equals] function.
* For floating point numbers it means that `NaN' is equal to itself and `-0.0' is not equal to `0.0'.

public expect infix fun CharArray.contentEquals(other: CharArray): Boolean

/**
 * Returns `true' if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 * The elements are compared for equality with the
[equals] function.
 * For floating point numbers it means that `NaN' is equal to itself and `-0.0' is not equal to `0.0'.

public expect infix fun BooleanArray.contentEquals(other: BooleanArray): Boolean

/**
 * Returns `true' if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 * The elements are compared for equality with the
[equals] function.
 * For floating point numbers it means that `NaN' is equal to itself and `-0.0' is not equal to `0.0'.

public expect infix fun ByteArray.contentEquals(other: ByteArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun ShortArray.contentEquals(other: ShortArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun IntArray.contentEquals(other: IntArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun LongArray.contentEquals(other: LongArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun FloatArray.contentEquals(other: FloatArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun DoubleArray.contentEquals(other: DoubleArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun CharArray.contentEquals(other: CharArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun BooleanArray.contentEquals(other: BooleanArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun ByteArray.contentEquals(other: ByteArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun ShortArray.contentEquals(other: ShortArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun IntArray.contentEquals(other: IntArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun LongArray.contentEquals(other: LongArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun FloatArray.contentEquals(other: FloatArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun DoubleArray.contentEquals(other: DoubleArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun CharArray.contentEquals(other: CharArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun BooleanArray.contentEquals(other: BooleanArray): Boolean

/**
 * Returns a hash code based on the contents of this array as if it is 
[List].

public expect infix fun ByteArray.contentEquals(other: ByteArray): Boolean
array as if it is [List].

```kotlin
@SinceKotlin("1.1")
public expect fun CharArray.contentToString(): String

@SinceKotlin("1.1")
public expect fun CharArray.contentToString(): String

@SinceKotlin("1.3")
public expect fun ByteArray.copyInto(destination: ByteArray,
    destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray

@SinceKotlin("1.3")
public expect fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray

@SinceKotlin("1.3")
public expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray
```

It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range. It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

```kotlin
@SinceKotlin("1.3")
public expect fun <T> Array<out T>.copyInto(destination: Array<T>,
    destinationOffset: Int = 0, endIndex: Int = size): Array<T>
```

It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

```kotlin
@SinceKotlin("1.3")
public expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray
```

It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

```kotlin
@SinceKotlin("1.3")
public expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray
```

It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

```kotlin
@SinceKotlin("1.3")
public expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray
```

It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * \n * @throws IndexOutOfBoundsException when the subrange doesn’t fit into the [destination] array starting at the specified [destinationOffset].\n * \n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin("1.3")\npublic expect fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * \n * @param destinationOffset the position in the [destination] array to copy to.\n * \n * @param startIndex the beginning (inclusive) of the subrange to copy,\n * \n * 0 by default.\n * \n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * \n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin("1.3")\npublic expect fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * \n * @param destinationOffset the position in the [destination] array to copy to.\n * \n * @param startIndex the beginning (inclusive) of the subrange to copy,\n * \n * 0 by default.\n * \n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * \n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin("1.3")\npublic expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * \n * @param destinationOffset the position in the [destination] array to copy to.\n * \n * @param startIndex the beginning (inclusive) of the subrange to copy,\n * \n * 0 by default.\n * \n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * \n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin("1.3")\npublic expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.\n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * \n * @param destinationOffset the position in the [destination] array to copy to.\n * \n * @param startIndex the beginning (inclusive) of the subrange to copy,\n * \n * 0 by default.\n * \n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * \n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin("1.3")\npublic expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):
CharArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("NO_ACTUAL_FOR_EXPECT")
public expect fun <T> Array<T>.copyOf():

ByteArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
public expect fun ByteArray.copyOf():

ShortArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
public expect fun ShortArray.copyOf():

IntArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
public expect fun IntArray.copyOf():

LongArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
public expect fun LongArray.copyOf():

FloatArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
public expect fun FloatArray.copyOf():

DoubleArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
public expect fun DoubleArray.copyOf():

CharArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
public expect fun CharArray.copyOf():

ByteArray.copyOf(newSize: Int): ByteArray

ShortArray.copyOf(newSize: Int): ShortArray

IntArray.copyOf(newSize: Int): IntArray

LongArray.copyOf(newSize: Int): LongArray

FloatArray.copyOf(newSize: Int): FloatArray

CharArray.copyOf(newSize: Int): CharArray

ByteArrayList.copyOf(newSize: Int): ByteArrayList

ShortArrayList.copyOf(newSize: Int): ShortArrayList

IntArrayList.copyOf(newSize: Int): IntArrayList

LongArrayList.copyOf(newSize: Int): LongArrayList

FloatArrayList.copyOf(newSize: Int): FloatArrayList

CharArrayList.copyOf(newSize: Int): CharArrayList

ByteArray.copyOf(newSize: Int): ByteArray

ShortArray.copyOf(newSize: Int): ShortArray

IntArray.copyOf(newSize: Int): IntArray

LongArray.copyOf(newSize: Int): LongArray

FloatArray.copyOf(newSize: Int): FloatArray

CharArray.copyOf(newSize: Int): CharArray
### filled with zero values.

* An public expect fun DoubleArray.copyOf(newSize: Int): DoubleArray? Returns new array which is a copy of the original array, resized to the given [newSize].

* @sample samples.collections.Arrays.copyOf

### BooleanArray.copyOf(newSize: Int): BooleanArray? Returns new array which is a copy of the original array, resized to the given [newSize].

* The copy is either truncated or padded at the end with `false` values if necessary.

* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.

* @sample samples.collections.Arrays.copyOf

### CharArray.copyOf(newSize: Int): CharArray? Returns new array which is a copy of the original array, resized to the given [newSize].

* The copy is either truncated or padded at the end with `null` values if necessary.

* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.

* @sample samples.collections.Arrays.copyOf

### ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray Returns a new array which is a copy of the specified range of the original array.

* @param fromIndex the start of the range (inclusive), must be in `0..array.size`

* @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

* @sample samples.collections.Arrays.copyOfRange

### ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray Returns a new array which is a copy of the specified range of the original array.

* @param fromIndex the start of the range (inclusive), must be in `0..array.size`

* @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

* @sample samples.collections.Arrays.copyOfRange

### IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray Returns a new array which is a copy of the specified range of the original array.

* @param fromIndex the start of the range (inclusive), must be in `0..array.size`

* @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

* @sample samples.collections.Arrays.copyOfRange

### LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray Returns a new array which is a copy of the specified range of the original array.

* @param fromIndex the start of the range (inclusive), must be in `0..array.size`

* @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

* @sample samples.collections.Arrays.copyOfRange

### FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray Returns a new array which is a copy of the specified range of the original array.

* @param fromIndex the start of the range (inclusive), must be in `0..array.size`

* @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

* @sample samples.collections.Arrays.copyOfRange

### DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray Returns a new array which is a copy of the specified range of the original array.

* @param fromIndex the start of the range (inclusive), must be in `0..array.size`

* @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

* @sample samples.collections.Arrays.copyOfRange

---

**Open Source Used In DNAC Platform Antelope 3770**
/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 */
@SinceKotlin("1.3")
public expect fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 */
@SinceKotlin("1.3")
public expect fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 */
@SinceKotlin("1.3")
public expect fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 */
@SinceKotlin("1.3")
public expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 */
@SinceKotlin("1.3")
public expect fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 */
@SinceKotlin("1.3")
public expect fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 */
@SinceKotlin("1.3")
public expect fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 */
@SinceKotlin("1.3")
public expect fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
public expect fun <T> Array<T>.indices: IntRange
    get() = IntRange(0, lastIndex)
*\public val ByteArray.indices: IntRange get() = IntRange(0, lastIndex)\n*\public val ShortArray.indices: IntRange get() = IntRange(0, lastIndex)\n*\public val IntArray.indices: IntRange get() = IntRange(0, lastIndex)\n*\public val LongArray.indices: IntRange get() = IntRange(0, lastIndex)\n*\public val FloatArray.indices: IntRange get() = IntRange(0, lastIndex)\n*\public val DoubleArray.indices: IntRange get() = IntRange(0, lastIndex)\n*\public val BooleanArray.indices: IntRange get() = IntRange(0, lastIndex)\n*\public val CharArray.indices: IntRange get() = IntRange(0, lastIndex)\n*\public val Array<out T>.isEmpty(): Boolean { return size == 0\n*\public val ByteArray.isEmpty(): Boolean { return size == 0\n*\public val ShortArray.isEmpty(): Boolean { return size == 0\n*\public val IntArray.isEmpty(): Boolean { return size == 0\n*\public val LongArray.isEmpty(): Boolean { return size == 0\n*\public val FloatArray.isEmpty(): Boolean { return size == 0\n*\public val DoubleArray.isEmpty(): Boolean { return size == 0\n*\public val BooleanArray.isEmpty(): Boolean { return size == 0\n*\public val CharArray.isEmpty(): Boolean { return size == 0\n*\public val <T> Array<out T>.isNotEmpty(): Boolean { return !isEmpty()\n*\public val ByteArray.isNotEmpty(): Boolean { return !isEmpty()\n*\public val ShortArray.isNotEmpty(): Boolean { return !isEmpty()\n*\public val IntArray.isNotEmpty(): Boolean { return !isEmpty()\n*\public val LongArray.isNotEmpty(): Boolean { return !isEmpty()\n*\public val FloatArray.isNotEmpty(): Boolean { return !isEmpty()\n*\public val DoubleArray.isNotEmpty(): Boolean { return !isEmpty()\n*\public val BooleanArray.isNotEmpty(): Boolean { return !isEmpty()\n*\public val CharArray.isNotEmpty(): Boolean { return !isEmpty()\n*\public val Array<out T>.lastIndex: Int get() = size - 1\n*\public val ByteArray.lastIndex: Int get() = size - 1\n*\public val ShortArray.lastIndex: Int get() = size - 1\n*\public val IntArray.lastIndex: Int get() = size - 1\n*\public val LongArray.lastIndex: Int get() = size - 1\n*\public val FloatArray.lastIndex: Int get() = size - 1\n*\public val DoubleArray.lastIndex: Int get() = size - 1\n//\n*\public val IntArray<out T>.operator fun plus(element: T):
Array<T>

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public expect operator fun ByteArray.plus(element: Byte): ByteArray

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public expect operator fun ShortArray.plus(element: Short): ShortArray

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public expect operator fun IntArray.plus(element: Int): IntArray

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public expect operator fun LongArray.plus(element: Long): LongArray

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public expect operator fun FloatArray.plus(element: Float): FloatArray

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public expect operator fun DoubleArray.plus(element: Double): DoubleArray

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public expect operator fun BooleanArray.plus(element: Boolean): BooleanArray

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public expect operator fun CharArray.plus(element: Char): CharArray

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] collection.
 */
@Suppress("NO_ACTUAL_FOR_EXPECT")
public expect operator fun <T> Array<T>.plus(elements: Collection<T>): Array<T>

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] collection.
 */
public expect operator fun ByteArray.plus(elements: ByteArray): ByteArray

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] collection.
 */
public expect operator fun ShortArray.plus(elements: ShortArray): ShortArray

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] collection.
 */
public expect operator fun IntArray.plus(elements: IntArray): IntArray

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] collection.
 */
public expect operator fun LongArray.plus(elements: LongArray): LongArray

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] collection.
 */
public expect operator fun FloatArray.plus(elements: FloatArray): FloatArray

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] collection.
 */
public expect operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] collection.
 */
public expect operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray
the original array and then all elements of the given [elements] array.

*/

public expect operator fun CharArray.plus(elements: CharArray): CharArray

/**
 * Returns an array containing all of the elements of the original array and then the given [element].
 */

@sample samples.collections.Arrays.Sorting.sortArray

public expect fun IntArray.sort(): Unit

public expect fun LongArray.sort(): Unit

public expect fun ShortArray.sort(): Unit

public expect fun ByteArray.sort(): Unit

public expect fun DoubleArray.sort(): Unit

public expect fun FloatArray.sort(): Unit

public expect fun CharArray.sort(): Unit

/**
 * Sorts the array in-place according to the natural order of its elements.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable
 */

public expect fun <T : Comparable<T>> Array<out T>.sort(): Unit

/**
 * Sorts the array in-place according to the order specified by the given [comparator].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * @sample samples.collections.Arrays.Sorting.sortWith
 */

public expect fun ByteArray.toTypedArray(): Array<Byte>

public expect fun ShortArray.toTypedArray(): Array<Short>

public expect fun IntArray.toTypedArray(): Array<Int>

public expect fun LongArray.toTypedArray(): Array<Long>

public expect fun FloatArray.toTypedArray(): Array<Float>

public expect fun DoubleArray.toTypedArray(): Array<Double>

public expect fun BooleanArray.toTypedArray(): Array<Boolean>

public expect fun CharArray.toTypedArray(): Array<Char>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @sample samples.collections.Arrays.TypedArrays
typical
 */

returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given array.

If any of two pairs would have the same key the last one gets added to the map.
map preserves the entry iteration order of the original array.

*public inline fun <T, K, V> Array<out T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    val capacity =
    mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform, |u| u */ */ |v| Returns a [Map] containing key-value pairs provided by [transform] function

* applied to elements of the given array.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original array.

*public inline fun <K, V> ByteArray.associate(transform: (Byte) -> Pair<K, V>): Map<K, V> {
    val capacity =
    mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform, |u| u */ */ |v| Returns a [Map] containing key-value pairs provided by [transform] function

* applied to elements of the given array.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original array.

*public inline fun <K, V> IntArray.associate(transform: (Int) -> Pair<K, V>): Map<K, V> {
    val capacity =
    mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform, |u| u */ */ |v| Returns a [Map] containing key-value pairs provided by [transform] function

* applied to elements of the given array.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original array.

*public inline fun <K, V> LongArray.associate(transform: (Long) -> Pair<K, V>): Map<K, V> {
    val capacity =
    mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform, |u| u */ */ |v| Returns a [Map] containing key-value pairs provided by [transform] function

* applied to elements of the given array.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original array.

*public inline fun <K, V> ShortArray.associate(transform: (Short) -> Pair<K, V>): Map<K, V> {
    val capacity =
    mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform, |u| u */ */ |v| Returns a [Map] containing key-value pairs provided by [transform] function

* applied to elements of the given array.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original array.

*public inline fun <K, V> FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V> {
    val capacity =
    mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform, |u| u */ */ |v| Returns a [Map] containing key-value pairs provided by [transform] function

* applied to elements of the given array.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original array.

*public inline fun <K, V> BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V> {
    val capacity =
    mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform, |u| u */ */ |v| Returns a [Map] containing key-value pairs provided by [transform] function

* applied to elements of the given array.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original array.

*public inline fun <K, V> CharArray.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {
    val capacity =
    mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform, |u| u */ */ |v| Returns a [Map] containing the elements from the given array indexed by the key

* returned from [keySelector] function applied to each element.

* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

* The returned map preserves the entry iteration order of the original array.

*public inline fun <T, K> Array<out T>.associateBy(keySelector: (T) -> K):
    Map<K, T> {
        val capacity = mapCapacity(size).coerceAtLeast(16)
        return
associateByTo(LinkedHashMap<K, T>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> \n\nval capacity = mapCapacity(size).coerceAtLeast(16)\nreturn associateByTo(LinkedHashMap<K, Byte>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> ShortArray.associateBy(keySelector: (Short) -> K): Map<K, Short> \n\nval capacity = mapCapacity(size).coerceAtLeast(16)\nreturn associateByTo(LinkedHashMap<K, Short>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> \n\nval capacity = mapCapacity(size).coerceAtLeast(16)\nreturn associateByTo(LinkedHashMap<K, Int>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> \n\nval capacity = mapCapacity(size).coerceAtLeast(16)\nreturn associateByTo(LinkedHashMap<K, Long>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> FloatArray.associateBy(keySelector: (Float) -> K): Map<K, Float> \n\nval capacity = mapCapacity(size).coerceAtLeast(16)\nreturn associateByTo(LinkedHashMap<K, Float>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> \n\nval capacity = mapCapacity(size).coerceAtLeast(16)\nreturn associateByTo(LinkedHashMap<K, Double>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> BooleanArray.associateBy(keySelector: (Boolean) -> K): Map<K, Boolean> \n\nval capacity = mapCapacity(size).coerceAtLeast(16)\nreturn associateByTo(LinkedHashMap<K, Boolean>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> \n\nval capacity = mapCapacity(size).coerceAtLeast(16)\nreturn associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)\n\n** Returns a [Map] containing the elements from the given array indexed by the key\n* returned from [keySelector] function applied to each element.\n* \n* \n* If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* \n* \n* The returned map preserves the entry iteration order of the original array.\n* \n* \n* In/\n* public inline fun <K> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> \n\nval capacity =
mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform))

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */

public inline fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */

public inline fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */

public inline fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */

public inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */

public inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */

public inline fun <K, V> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */

public inline fun <K, V> CharArray.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,
 * where key is provided by the [keySelector] function
 */
applied to each element of the given array\n and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
</K, M : MutableMap<in K, in T>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
public inline fun <K, M : MutableMap<in K, in Byte>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
public inline fun <K, M : MutableMap<in K, in Short>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
public inline fun <K, M : MutableMap<in K, in Int>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
public inline fun <K, M : MutableMap<in K, in Long>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
public inline fun <K, M : MutableMap<in K, in Float>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
public inline fun <K, M : MutableMap<in K, in Double>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
public inline fun <K, M : MutableMap<in K, in Boolean>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */
public inline fun <K, M : MutableList<in K, in T>> MutableList<out T>.associateByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)\n    }
    return destination\n}
one gets added to the map.

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the
 * [keySelector] function and value is provided by the [valueTransform] function applied to elements of the
 * given array.
 */
public inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        // Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the
        // [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array.
        // If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector]
 * function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector]
 * function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector]
 * function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector]
 * function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector]
 * function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector]
 * function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector]
 * function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```
MutableMap<in K, in V>> CharArray.associateByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        return destination.put(keySelector(element),
            valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> Array<out T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {
    for (element in this) {
        return destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform: (Byte) -> Pair<K, V>): M {
    for (element in this) {
        return destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) -> Pair<K, V>): M {
    for (element in this) {
        return destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {
    for (element in this) {
        return destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 */
    for (element in this) {
        return destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) -> Pair<K, V>): M {
    for (element in this) {
        return destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform: (Double) -> Pair<K, V>): M {
    for (element in this) {
        return destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateTo(destination: M, transform: (Boolean) -> Pair<K, V>): M {
    for (element in this) {
        return destination += transform(element)
    }
    return destination
}
(item in this) \n  destination.add(item)\n return destination

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in T>> Array<T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Return a [HashSet] of all elements.
 */
public fun <T> Array<T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>(mapCapacity(size)))
}

/**
 * Returns a [List] containing all elements.
 */
public fun <T> Array<T>.toList(): List<T> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}

Open Source Used In DNAC Platform Antelope 3781
** Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

```kotlin
public inline fun <R> ByteArray.flatMap(transform: (Byte) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

```
invoked on each element of original array, to the given \[destination].

```kotlin
public inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> CharArray.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
```

```kotlin
public inline fun <T, K> Array<out T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)
}
```

```kotlin
public inline fun <K> ByteArray.groupBy(keySelector: (Byte) -> K): Map<K, List<Byte>> {
    return groupByTo(LinkedHashMap<K, MutableList<Byte>>(), keySelector)
}
```

```kotlin
public inline fun <K> ShortArray.groupBy(keySelector: (Short) -> K): Map<K, List<Short>> {
    return groupByTo(LinkedHashMap<K, MutableList<Short>>(), keySelector)
}
```

```kotlin
public inline fun <K> IntArray.groupBy(keySelector: (Int) -> K): Map<K, List<Int>> {
    return groupByTo(LinkedHashMap<K, MutableList<Int>>(), keySelector)
}
```

```kotlin
public inline fun <K> LongArray.groupBy(keySelector: (Long) -> K): Map<K, List<Long>> {
    return groupByTo(LinkedHashMap<K, MutableList<Long>>(), keySelector)
}
```

```kotlin
public inline fun <K> FloatArray.groupBy(keySelector: (Float) -> K): Map<K, List<Float>> {
    return groupByTo(LinkedHashMap<K, MutableList<Float>>(), keySelector)
}
```

```kotlin
public inline fun <K> DoubleArray.groupBy(keySelector: (Double) -> K): Map<K, List<Double>> {
    return groupByTo(LinkedHashMap<K, MutableList<Double>>(), keySelector)
}
```
entry iteration order of the keys produced from the original array.

```kotlin
BooleanArray.groupBy(keySelector: (Boolean) -> K): Map<K, List<Boolean>>
```

Groups elements of the original array by the key returned by the given `keySelector` function
applied to each element and returns a map where each group key is associated with a list of corresponding elements.

```kotlin
CharArray.groupBy(keySelector: (Char) -> K): Map<K, List<Char>>
```

Groups values returned by the `valueTransform` function applied to each element of the original array
by the key returned by the given `keySelector` function
applied to the element and returns a map where each group key is associated with a list of corresponding values.

```kotlin
Array<out T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>>
```

Groups values returned by the `valueTransform` function applied to each element of the original array
by the key returned by the given `keySelector` function
applied to the element and returns a map where each group key is associated with a list of corresponding values.

```kotlin
ByteArray.groupBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, List<V>>
```

Groups values returned by the `valueTransform` function applied to each element of the original array
by the key returned by the given `keySelector` function
applied to the element and returns a map where each group key is associated with a list of corresponding values.

```kotlin
ShortArray.groupBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, List<V>>
```

Groups values returned by the `valueTransform` function applied to each element of the original array
by the key returned by the given `keySelector` function
applied to the element and returns a map where each group key is associated with a list of corresponding values.

```kotlin
IntArray.groupBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>>
```

Groups values returned by the `valueTransform` function applied to each element of the original array
by the key returned by the given `keySelector` function
applied to the element and returns a map where each group key is associated with a list of corresponding values.

```kotlin
LongArray.groupBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, List<V>>
```

Groups values returned by the `valueTransform` function applied to each element of the original array
by the key returned by the given `keySelector` function
applied to the element and returns a map where each group key is associated with a list of corresponding values.
```kotlin
public inline fun <K, V>
FloatArray.groupBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element, and returns a map where each group key is associated
 * with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K, V>
DoubleArray.groupBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element, and returns a map where each group key is associated
 * with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K, V>
BooleanArray.groupBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element, and returns a map where each group key is associated
 * with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K, V>
CharArray.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * @return The [destination] map.
 */
public inline fun <T, K, M : MutableMap<in K, MutableList<T>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<T>() }
        list.add(element)
    }
    return destination
}
```
Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements:

```kotlin
@public inline fun <K, M : MutableMap<K, MutableList<Long>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Long>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@public inline fun <K, M : MutableMap<K, MutableList<Float>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Float>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@public inline fun <K, M : MutableMap<K, MutableList<Double>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Double>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@public inline fun <K, M : MutableMap<K, MutableList<Boolean>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Boolean>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@public inline fun <K, M : MutableMap<K, MutableList<Char>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Char>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@public inline fun <T, K, V, M : MutableMap<K, MutableList<V>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}
```

```kotlin
@public inline fun <T, K, V, M : MutableMap<K, MutableList<V>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}
```
[destination].

* @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
* @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
* @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
* @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
* @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
* @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
* @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.

*/
public inline fun <R, C : MutableCollection<R>> IntArray.mapIndexedTo(destination: C, transform: (index: Int, Int) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and its index in the original array and appends the results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.

*/
public inline fun <R, C : MutableCollection<R>> LongArray.mapIndexedTo(destination: C, transform: (index: Int, Long) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and its index in the original array and appends the results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.

*/
public inline fun <R, C : MutableCollection<R>> DoubleArray.mapIndexedTo(destination: C, transform: (index: Int, Double) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and its index in the original array and appends the results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.

*/
public inline fun <R, C : MutableCollection<R>> BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and its index in the original array and appends the results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself and returns the result of the transform applied to the element.

*/
public inline fun <R, C : MutableCollection<R>> CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Returns a list containing only the non-null results of applying the given [transform] function to each element in the original array.

*/
public inline fun <T, R : Any> Array<out T>.mapNotNull(transform: (T) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(), transform)
}

/**
 * Applies the given [transform] function to each element in the original array and appends only the non-null results to the given [destination].

*/
public inline fun <T, R : Any, C : MutableCollection<R>> Array<out T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
    forEach { element -> transform(element)?.let { destination.add(it) } }
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

*/
public inline fun <T, R, C : MutableCollection<R>> Array<out T>.mapTo(destination: C, transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

*/
public inline fun <R, C : MutableCollection<R>> ByteArray.mapTo(destination: C, transform: (Byte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

*/
public inline fun <R, C : MutableCollection<R>> ShortArray.mapTo(destination: C, transform: (Short) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

*/
public inline fun <R, C : MutableCollection<R>> IntArray.mapTo(destination: C, transform: (Int) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

*/
public inline fun <R, C : MutableCollection<R>> ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and appends the results to the given [destination].

*/
public inline fun <R, C : MutableCollection<R>> IntArray.mapIndexedTo(destination: C, transform: (index: Int, Int) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
element of the original array\n * and appends the results to the given [destination].\n */
public inline fun <R, C : 
MutableCollection<in R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */
public inline fun <R, C : 
MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */
public inline fun <R, C : 
MutableCollection<in R>> DoubleArray.mapTo(destination: C, transform: (Double) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */
public inline fun <R, C : 
MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */
public inline fun <R, C : 
MutableCollection<in R>> CharArray.mapTo(destination: C, transform: (Char) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */
public fun <T> Array<out T>.withIndex(): Iterable<IndexedValue<T>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */
public fun ByteArray.withIndex(): Iterable<IndexedValue<Byte>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */
public fun ShortArray.withIndex(): Iterable<IndexedValue<Short>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */
public fun IntArray.withIndex(): Iterable<IndexedValue<Int>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */
public fun LongArray.withIndex(): Iterable<IndexedValue<Long>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */
public fun FloatArray.withIndex(): Iterable<IndexedValue<Float>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */
public fun DoubleArray.withIndex(): Iterable<IndexedValue<Double>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */
public fun BooleanArray.withIndex(): Iterable<IndexedValue<Boolean>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun <T> Array<out T>.distinct(): List<T> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun ByteArray.distinct(): List<Byte> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun ShortArray.distinct(): List<Short> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun IntArray.distinct(): List<Int> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun LongArray.distinct(): List<Long> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun FloatArray.distinct(): List<Float> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun DoubleArray.distinct(): List<Double> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun BooleanArray.distinct(): List<Boolean> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.\n * The elements in the resulting list are in the same order as they were in the source array.\n */
public fun CharArray.distinct(): List<Char> {
    return this.toMutableSet().toList()
}
IntArray.distinct(): List<Int> {\n    return this.toMutableSet().toList().\n}

ByteArray.distinctBy(selector: (Byte) -> K): List<Byte> {
    val set = HashSet<K>()
    val list = ArrayList<Byte>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

/**
 * Returns a list containing only distinct elements from the
 * given array\n * having distinct keys returned by the given [selector] function.\n * The elements in the resulting list are in the same order as they were
 * in the source array.\n *\n */
public fun ByteArray.distinct(): List<Byte> {
    return this.toMutableSet().toList().\n}

LongArray.distinctBy(selector: (Long) -> K): List<Long> {
    val set = HashSet<K>()
    val list = ArrayList<Long>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

/**
 * Returns a list containing only distinct elements from the
 * given array\n * having distinct keys returned by the given [selector] function.\n * The elements in the resulting list are in the same order as they were
 * in the source array.\n */
public fun LongArray.distinct(): List<Long> {
    return this.toMutableSet().toList().\n}

IntArray.distinct(): List<Int> {
    return this.toMutableSet().toList().\n}

/**
 * Returns a list containing only distinct elements from the
 * given array\n * having distinct keys returned by the given [selector] function.\n * The elements in the resulting list are in the same order as they were
 * in the source array.\n */
public fun IntArray.distinct(): List<Int> {
    return this.toMutableSet().toList().\n}

/**
 * Returns a list containing only distinct elements from the
 * given array\n * having distinct keys returned by the given [selector] function.\n * The elements in the resulting list are in the same order as they were
 * in the source array.\n */
public fun CharArray.distinct(): List<Char> {
    return this.toMutableSet().toList().\n}

DoubleArray.distinctBy(selector: (Double) -> K): List<Double> {
    val set = HashSet<K>()
    val list = ArrayList<Double>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = HashSet<K>()
    val list = ArrayList<Char>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

BooleanArray.distinctBy(selector: (Boolean) -> K): List<Boolean> {
    val set = HashSet<K>()
    val list = ArrayList<Boolean>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ListOf<Double>.distinctBy(selector: (Double) -> K): List<Double> {
    val set = HashSet<K>()
    val list = ArrayList<Double>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ListOf<Long>.distinctBy(selector: (Long) -> K): List<Long> {
    val set = HashSet<K>()
    val list = ArrayList<Long>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Float>.distinctBy(selector: (Float) -> K): List<Float> {
    val set = HashSet<K>()
    val list = ArrayList<Float>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Short>.distinctBy(selector: (Short) -> K): List<Short> {
    val set = HashSet<K>()
    val list = ArrayList<Short>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Byte>.distinctBy(selector: (Byte) -> K): List<Byte> {
    val set = HashSet<K>()
    val list = ArrayList<Byte>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Char>.distinctBy(selector: (Char) -> K): List<Char> {
    val set = HashSet<K>()
    val list = ArrayList<Char>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Double>.distinctBy(selector: (Double) -> K): List<Double> {
    val set = HashSet<K>()
    val list = ArrayList<Double>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Long>.distinctBy(selector: (Long) -> K): List<Long> {
    val set = HashSet<K>()
    val list = ArrayList<Long>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Float>.distinctBy(selector: (Float) -> K): List<Float> {
    val set = HashSet<K>()
    val list = ArrayList<Float>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Short>.distinctBy(selector: (Short) -> K): List<Short> {
    val set = HashSet<K>()
    val list = ArrayList<Short>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Byte>.distinctBy(selector: (Byte) -> K): List<Byte> {
    val set = HashSet<K>()
    val list = ArrayList<Byte>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Char>.distinctBy(selector: (Char) -> K): List<Char> {
    val set = HashSet<K>()
    val list = ArrayList<Char>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Double>.distinctBy(selector: (Double) -> K): List<Double> {
    val set = HashSet<K>()
    val list = ArrayList<Double>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Long>.distinctBy(selector: (Long) -> K): List<Long> {
    val set = HashSet<K>()
    val list = ArrayList<Long>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Float>.distinctBy(selector: (Float) -> K): List<Float> {
    val set = HashSet<K>()
    val list = ArrayList<Float>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Short>.distinctBy(selector: (Short) -> K): List<Short> {
    val set = HashSet<K>()
    val list = ArrayList<Short>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Byte>.distinctBy(selector: (Byte) -> K): List<Byte> {
    val set = HashSet<K>()
    val list = ArrayList<Byte>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Char>.distinctBy(selector: (Char) -> K): List<Char> {
    val set = HashSet<K>()
    val list = ArrayList<Char>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Double>.distinctBy(selector: (Double) -> K): List<Double> {
    val set = HashSet<K>()
    val list = ArrayList<Double>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

ArrayOf<Long>.distinctBy(selector: (Long) -> K): List<Long> {
    val set = HashSet<K>()
    val list = ArrayList<Long>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
Open Source Used In DNAC Platform Antelope
set.removeAll(other) return set

The returned set preserves the element iteration order of the original array.

\( *\)\text{public infix fun ShortArray.subtract(other: Iterable<Short>): Set<Short> \{\text{val set = this.toMutableSet()_~}\}
\text{return set}removeAll(other) return set\}un/*\*un * \text{Returns a set containing all elements that are contained by this array and not contained by the specified collection.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public infix fun IntArray.subtract(other: Iterable<Int>): Set<Int> \{\text{val set = this.toMutableSet()_~}\}
\text{return set}removeAll(other) return set\}un/*\*un * \text{Returns a set containing all elements that are contained by this array and not contained by the specified collection.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public infix fun LongArray.subtract(other: Iterable<Long>): Set<Long> \{\text{val set = this.toMutableSet()_~}\}
\text{return set}removeAll(other) return set\}un/*\*un * \text{Returns a set containing all elements that are contained by this array and not contained by the specified collection.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public infix fun FloatArray.subtract(other: Iterable<Float>): Set<Float> \{\text{val set = this.toMutableSet()_~}\}
\text{return set}removeAll(other) return set\}un/*\*un * \text{Returns a set containing all elements that are contained by this array and not contained by the specified collection.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public infix fun CharArray.subtract(other: Iterable<Character>): Set<Character> \{\text{val set = this.toMutableSet()_~}\}
\text{return set}removeAll(other) return set\}un/*\*un * \text{Returns a set containing all elements that are contained by this array and not contained by the specified collection.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public fun ByteArray.toMutableSet(): MutableSet<Byte> \{\text{val set = LinkedHashSet<Byte>(mapCapacity(size))_~}\}
\text{return set}add(item) return set\}un/*\*un * \text{Returns a set containing all distinct elements from the given array.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public fun ShortArray.toMutableSet(): MutableSet<Short> \{\text{val set = LinkedHashSet<Short>(mapCapacity(size))_~}\}
\text{return set}add(item) return set\}un/*\*un * \text{Returns a set containing all distinct elements from the given array.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public fun IntArray.toMutableSet(): MutableSet<Int> \{\text{val set = LinkedHashSet<Int>(mapCapacity(size))_~}\}
\text{return set}add(item) return set\}un/*\*un * \text{Returns a set containing all distinct elements from the given array.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public fun LongArray.toMutableSet(): MutableSet<Long> \{\text{val set = LinkedHashSet<Long>(mapCapacity(size))_~}\}
\text{return set}add(item) return set\}un/*\*un * \text{Returns a set containing all distinct elements from the given array.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public fun FloatArray.toMutableSet(): MutableSet<Float> \{\text{val set = LinkedHashSet<Float>(mapCapacity(size))_~}\}
\text{return set}add(item) return set\}un/*\*un * \text{Returns a set containing all distinct elements from the given array.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public fun DoubleArray.toMutableSet(): MutableSet<Double> \{\text{val set = LinkedHashSet<Double>(mapCapacity(size))_~}\}
\text{return set}add(item) return set\}un/*\*un * \text{Returns a set containing all distinct elements from the given array.}\n\* * \text{The returned set preserves the element iteration order of the original array.}\n\* /\text{public fun BooleanArray.toMutableSet(): MutableSet<Boolean> \{\text{val set = LinkedHashSet<Boolean>(mapCapacity(size))_~}\}
\text{return set}add(item) return set\}un/*\*un *
Returns a mutable set containing all distinct elements from the given array.

*OrNull The returned set preserves the element iteration order of the original array.

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull

*OrNull
Returns `true` if all elements match the given [predicate].

```kotlin
public inline fun ByteArray.all(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

public inline fun ShortArray.all(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

public inline fun IntArray.all(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

public inline fun LongArray.all(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

public inline fun FloatArray.all(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

public inline fun DoubleArray.all(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

public inline fun BooleanArray.all(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}
```

```kotlin
public func any(): Boolean {
    return !isEmpty()
}

public func ByteArray.any(): Boolean {
    return !isEmpty()
}

public func ShortArray.any(): Boolean {
    return !isEmpty()
}

public func IntArray.any(): Boolean {
    return !isEmpty()
}

public func LongArray.any(): Boolean {
    return !isEmpty()
}

public func FloatArray.any(): Boolean {
    return !isEmpty()
}

public func DoubleArray.any(): Boolean {
    return !isEmpty()
}

public func BooleanArray.any(): Boolean {
    return !isEmpty()
}

public func CharArray.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
public inline fun <T> Array<out T>.any(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

public inline fun ByteArray.any(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

public inline fun ShortArray.any(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

public inline fun IntArray.any(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

public inline fun LongArray.any(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

public inline fun FloatArray.any(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

public inline fun DoubleArray.any(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

public inline fun BooleanArray.any(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

public inline fun CharArray.any(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
public func any(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

Open Source Used In DNAC Platform Antelope 3797
starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> LongArray.foldRight(initial: R, operation: (Long, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> DoubleArray.foldRight(initial: R, operation: (Double, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> CharArray.foldRight(initial: R, operation: (Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> BooleanArray.foldRight(initial: R, operation: (Boolean, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> ByteArray.foldRightIndexed(initial: R, operation: (index: Int, Byte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> IntArray.foldRightIndexed(initial: R, operation: (index: Int, Int, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> ShortArray.foldRightIndexed(initial: R, operation: (index: Int, Short, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> FloatArray.foldRight(initial: R, operation: (Float, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> DoubleArray.foldRightIndexed(initial: R, operation: (Double, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> CharArray.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> BooleanArray.foldRightIndexed(initial: R, operation: (index: Int, Boolean, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> EnumArray.foldRightIndexed(initial: R, operation: (index: Int, Enum, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.

```kotlin
*pubinline fun <R> EnumArray.foldRightIndexed(initial: R, operation: (index: Int, Enum, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
its index in the original array and current accumulator value.

* @param [operation] function that takes the index of
an element, the element itself and current accumulator value, and calculates the next accumulator value.

* @public inline fun <R> FloatArray.foldRightIndexed(initial: R, operation: (index: Int, Float, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

* @public inline fun <R> DoubleArray.foldRightIndexed(initial: R, operation: (index: Int, Double, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

* @public inline fun <R> BooleanArray.foldRightIndexed(initial: R, operation: (index: Int, Boolean, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

* @public inline fun <R> CharArray.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

* @public inline fun <T> Array<out T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

* @public inline fun ByteArray.forEachIndexed(action: (index: Int, Byte) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

* @public inline fun ShortArray.forEachIndexed(action: (index: Int, Short) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

* @public inline fun IntArray.forEachIndexed(action: (index: Int, Int) -> Unit): Unit {
    for (item in this) action(index++, item)
}

* @public inline fun LongArray.forEachIndexed(action: (index: Int, Long) -> Unit): Unit {
    for (item in this) action(index++, item)
}

* @public inline fun FloatArray.forEachIndexed(action: (index: Int, Float) -> Unit): Unit {
    for (item in this) action(index++, item)
}

* @public inline fun DoubleArray.forEachIndexed(action: (index: Int, Double) -> Unit): Unit {
    for (item in this) action(index++, item)
}

* @public inline fun BooleanArray.forEachIndexed(action: (index: Int, Boolean) -> Unit): Unit {
    for (item in this) action(index++, item)
}

* @public inline fun CharArray.forEachIndexed(action: (index: Int, Char) -> Unit): Unit {
    for (item in this) action(index++, item)
}
CharArray.max(): Char? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun max
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (max < e) max = e
    }
    return max
}

DoubleArray.max(): Double? {
    if (isEmpty()) return null
    var max = this[0]
    if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

LongArray.max(): Long? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

ByteArray.max(): Byte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

Array<out T>.max(): T? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

ArrayOfDouble.max(): Double? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

ArrayOfLong.max(): Long? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

ArrayOfChar.max(): Char? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

ArrayOfBoolean.max(): Boolean? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}
= this[i] \n    if (max < e) max = e
} \n
return max\n
\n\n\n* Returns the first element yielding the largest value of the given function or `null` if there are no elements.
\n\n@sample
\n\nsamples.collections.Collections.Aggregates.maxBy\n\n*/

\n\n
public inline fun <T, R : Comparable<R>> Array\n\n<T>.maxBy(selector: (T) -> R): T? { \n    \n    if (isEmpty()) return null
    \n    val lastIndex = this.lastIndexOf
    \n    if (lastIndex == 0) return maxElem
    \n    var maxValue = selector(maxElem) \n    \n    for (i in 1..lastIndex) { \n        val e = this[i] \n        val v = selector(e) \n        if (maxValue < v) { \n            maxElem = e
            maxValue = v
        } \n    } \n
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample
 * samples.collections.Collections.Aggregates.maxBy
 */

public inline fun <T, R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte? { \n    \n    if (isEmpty()) return null
    \n    var maxElem = this[0]
    \n    val lastIndex = this.lastIndexOf
    \n    if (lastIndex == 0) return maxElem
    \n    var maxValue = selector(maxElem) \n    \n    for (i in 1..lastIndex) { \n        val e = this[i]
        val v = selector(e) \n        if (maxValue < v) { \n            maxElem = e
            maxValue = v
        } \n    } \n
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample
 * samples.collections.Collections.Aggregates.maxBy
 */

public inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R): Short? { \n    \n    if (isEmpty()) return null
    \n    var maxElem = this[0]
    \n    val lastIndex = this.lastIndexOf
    \n    if (lastIndex == 0) return maxElem
    \n    var maxValue = selector(maxElem) \n    \n    for (i in 1..lastIndex) { \n        val e = this[i]
        val v = selector(e) \n        if (maxValue < v) { \n            maxElem = e
            maxValue = v
        } \n    } \n
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample
 * samples.collections.Collections.Aggregates.maxBy
 */

public inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int? { \n    \n    if (isEmpty()) return null
    \n    var maxElem = this[0]
    \n    val lastIndex = this.lastIndexOf
    \n    if (lastIndex == 0) return maxElem
    \n    var maxValue = selector(maxElem) \n    \n    for (i in 1..lastIndex) { \n        val e = this[i]
        val v = selector(e) \n        if (maxValue < v) { \n            maxElem = e
            maxValue = v
        } \n    } \n
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample
 * samples.collections.Collections.Aggregates.maxBy
 */

public inline fun <R : Comparable<R>> LongArray.maxBy(selector: (Long) -> R): Long? { \n    \n    if (isEmpty()) return null
    \n    var maxElem = this[0]
    \n    val lastIndex = this.lastIndexOf
    \n    if (lastIndex == 0) return maxElem
    \n    var maxValue = selector(maxElem) \n    \n    for (i in 1..lastIndex) { \n        val e = this[i]
        val v = selector(e) \n        if (maxValue < v) { \n            maxElem = e
            maxValue = v
        } \n    } \n
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample
 * samples.collections.Collections.Aggregates.maxBy
 */

public inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R): Float? { \n    \n    if (isEmpty()) return null
    \n    var maxElem = this[0]
    \n    val lastIndex = this.lastIndexOf
    \n    if (lastIndex == 0) return maxElem
    \n    var maxValue = selector(maxElem) \n    \n    for (i in 1..lastIndex) { \n        val e = this[i]
        val v = selector(e) \n        if (maxValue < v) { \n            maxElem = e
            maxValue = v
        } \n    } \n
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample
 * samples.collections.Collections.Aggregates.maxBy
 */

public inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) -> R): Double? { \n    \n    if (isEmpty()) return null
    \n    var maxElem = this[0]
    \n    val lastIndex = this.lastIndexOf
    \n    if (lastIndex == 0) return maxElem
    \n    var maxValue = selector(maxElem) \n    \n    for (i in 1..lastIndex) { \n        val e = this[i]
        val v = selector(e) \n        if (maxValue < v) { \n            maxElem = e
            maxValue = v
        } \n    } \n
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample
 * samples.collections.Collections.Aggregates.maxBy
 */

public inline fun <R : Comparable<R>> BooleanArray.maxBy(selector: (Boolean) -> R): Boolean? { \n    \n    if (isEmpty()) return null
    \n    var maxElem = this[0]
    \n    val lastIndex = this.lastIndexOf
    \n    if (lastIndex == 0) return maxElem
    \n    var maxValue = selector(maxElem) \n    \n    for (i in 1..lastIndex) { \n        val e = this[i]
        val v = selector(e) \n        if (maxValue < v) { \n            maxElem = e
            maxValue = v
        } \n    } \n
    return maxElem
}
{\n    maxElem = e\n    maxValue = v\n    }\n    return maxElem;\n}/** 
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n */@sample
samples.collections.Collections.Aggregates.maxBy\n*/\n
/** 
 * This is a function that finds the minimum element in an array. It iterates over the array, comparing each element to the current minimum. If the current element is smaller than the current minimum, it updates the minimum. The function returns the minimum element found.

```kotlin
public fun <T : Comparable<T>> Array<out T>.min(): T? {
    if (isEmpty()) return null
    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }
    return min
}
```

* 
* If any of elements is `NaN` returns `NaN`.

```kotlin
@SinceKotlin("1.1")
public fun Array<out Float>.min(): Float? {
    if (isEmpty()) return null
    var min = this[0]\n    if (min.isNaN()) return min
    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}
```

```kotlin
@SinceKotlin("1.1")
public fun Array<out Double>.min(): Double? {
    if (isEmpty()) return null
    var min = this[0]\n    if (min.isNaN()) return min
    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
/** 
 * Returns the largest value in the array. It iterates over the array, comparing each element to the current maximum. If the current element is larger than the current maximum, it updates the maximum. The function returns the maximum element found.

```kotlin
public fun ByteArray.maxWith(comparator: Comparator<in Byte>): Byte? {
    if (isEmpty()) return null
    max
}
```

```kotlin
/** 
 * Returns the largest value in the array. It iterates over the array, comparing each element to the current maximum. If the current element is larger than the current maximum, it updates the maximum. The function returns the maximum element found.

```kotlin
public fun IntArray.maxWith(comparator: Comparator<in Int>): Int? {
    if (isEmpty()) return null
    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
/** 
 * Returns the largest value in the array. It iterates over the array, comparing each element to the current maximum. If the current element is larger than the current maximum, it updates the maximum. The function returns the maximum element found.

```kotlin
public fun LongArray.maxWith(comparator: Comparator<in Long>): Long? {
    if (isEmpty()) return null
    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
/** 
 * Returns the largest value in the array. It iterates over the array, comparing each element to the current maximum. If the current element is larger than the current maximum, it updates the maximum. The function returns the maximum element found.

```kotlin
public fun ShortArray.maxWith(comparator: Comparator<in Short>): Short? {
    if (isEmpty()) return null
    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
/** 
 * Returns the largest value in the array. It iterates over the array, comparing each element to the current maximum. If the current element is larger than the current maximum, it updates the maximum. The function returns the maximum element found.

```kotlin
public fun BooleanArray.maxWith(comparator: Comparator<in Boolean>): Boolean? {
    if (isEmpty()) return null
    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
/** 
 * Returns the largest value in the array. It iterates over the array, comparing each element to the current maximum. If the current element is larger than the current maximum, it updates the maximum. The function returns the maximum element found.

```kotlin
public fun CharArray.maxWith(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null
    var maxElem = e\n    var maxValue = v\n    }\n    return maxElem;\n}/** 
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */@sample
CharArray.maxWith(\n    comparator: (Char) -> R): Char? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    var lastIndexOf = this.lastIndexOf\n    if (lastIndexOf == 0) return maxElem\n    var maxValue = selector(maxElem)\n    for (i in 1..lastIndexOf) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v
        }\n    }\n    return maxElem
}
```

```kotlin
public fun <T : Comparable<T>> Array<out T>\n.maxWith(\n    comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
public fun <T : Comparable<T>> Array<out T>\n.maxWith(\n    comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    var lastIndexOf = this.lastIndexOf\n    if (lastIndexOf == 0) return max\n    var maxValue = selector(max)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
public fun <T : Comparable<T>> Array<out T>\n.maxWith(\n    comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    var lastIndexOf = this.lastIndexOf\n    if (lastIndexOf == 0) return max\n    var maxValue = selector(max)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
public fun <T : Comparable<T>> Array<out T>\n.maxWith(\n    comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    var lastIndexOf = this.lastIndexOf\n    if (lastIndexOf == 0) return max\n    var maxValue = selector(max)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
public fun <T : Comparable<T>> Array<out T>\n.maxWith(\n    comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    var lastIndexOf = this.lastIndexOf\n    if (lastIndexOf == 0) return max\n    var maxValue = selector(max)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
public fun <T : Comparable<T>> Array<out T>\n.maxWith(\n    comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    var lastIndexOf = this.lastIndexOf\n    if (lastIndexOf == 0) return max\n    var maxValue = selector(max)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
public fun <T : Comparable<T>> Array<out T>\n.maxWith(\n    comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    var lastIndexOf = this.lastIndexOf\n    if (lastIndexOf == 0) return max\n    var maxValue = selector(max)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
public fun <T : Comparable<T>> Array<out T>\n.maxWith(\n    comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var max = this[0]\n    var lastIndexOf = this.lastIndexOf\n    if (lastIndexOf == 0) return max\n    var maxValue = selector(max)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }
    return max
}
```

```kotlin
@SinceKotlin("1.1")
public fun Array<out Double>.min(): Double? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

```kotlin
@SinceKotlin("1.1")
public fun Array<out Float>.min(): Float? {
    if (isEmpty()) return null
    var min = this[0]
    if (min.isNaN()) return min
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}
```

```kotlin
@SinceKotlin("1.1")
public fun Array<out Double>.min(): Double? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}
```
open source used in DNAC platform antelope 3806

/**
 * Returns the smallest element or `null` if there are no elements.
 */
public fun ByteArray.min(): Byte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        if (min > this[i]) min = this[i]
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 */
public fun ShortArray.min(): Short? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        if (min > this[i]) min = this[i]
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 */
public fun IntArray.min(): Int? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        if (min > this[i]) min = this[i]
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 */
public fun LongArray.min(): Long? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        if (min > this[i]) min = this[i]
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * 
 * If any of elements is `NaN` returns `NaN`.
 */
public fun FloatArray.min(): Float? {
    if (isEmpty()) return null
    var min = this[0]
    if (min.isNaN()) return min
    for (i in 1..lastIndex) {
        if (this[i].isNaN()) return this[i]
        if (min > this[i]) {
            min = this[i]
        }
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * 
 * If any of elements is `NaN` returns `NaN`.
 */
public fun DoubleArray.min(): Double? {
    if (isEmpty()) return null
    var min = this[0]
    if (min.isNaN()) return min
    for (i in 1..lastIndex) {
        if (this[i].isNaN()) return this[i]
        if (min > this[i]) {
            min = this[i]
        }
    }
    return min
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public fun ByteArray.minBy(selector: (Byte) -> R): Byte? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public fun ShortArray.minBy(selector: (Short) -> R): Short? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public fun IntArray.minBy(selector: (Int) -> R): Int? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public fun LongArray.minBy(selector: (Long) -> R): Long? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
minValue = v
\nreturn minElem

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public inline fun <R : Comparable<R>>
FloatArray.minBy(selector: (Float) -> R): Float? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public inline fun <R : Comparable<R>>
DoubleArray.minBy(selector: (Double) -> R): Double? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public inline fun <R : Comparable<R>>
CharArray.minBy(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun <T> Array<out T>.minWith(comparator: Comparator<in T>): T? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun ByteArray.minWith(comparator: Comparator<in Byte>): Byte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun ShortArray.minWith(comparator: Comparator<in Short>): Short? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun IntArray.minWith(comparator: Comparator<in Int>): Int? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun LongArray.minWith(comparator: Comparator<in Long>): Long? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun FloatArray.minWith(comparator: Comparator<in Float>): Float? {
    if (isEmpty()) return null
    var min =
this[0] for (i in 1..lastIndex) {\n    val e = this[i] if (comparator.compare(min, e) > 0) min = e\n} return min

* Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.

\n
/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun BooleanArray.minWith(comparator: Comparator<in Boolean>): Boolean? {
    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i] if (comparator.compare(min, e) > 0) min = e\n    } return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun CharArray.minWith(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i] if (comparator.compare(min, e) > 0) min = e\n    } return min
}

/**
 * Returns `true` if the array has no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.none
 */
public fun <T> Array<out T>.none(): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun ByteArray.none(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun ShortArray.none(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun IntArray.none(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun LongArray.none(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun FloatArray.none(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun DoubleArray.none(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun CharArray.none(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false\n    return true
}
samples.collections.Collections.Aggregates.noneWithPredicate
Boolean Array none(predicate: Boolean) -> Boolean { return true } Returns 'true' if no elements match the given [predicate].

/**
 * Returns `true` if no elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun BooleanArray none(predicate: Boolean) -> Boolean { return false } Returns 'false' if no elements match the given [predicate].

/**
 * Returns `true` if no elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline funCharArray none(predicate: Char) -> Boolean { return false } Returns 'false' if no elements match the given [predicate].
Array<out T>.reduceIndexed(operation: (index: Int, acc: S, T) -&gt; S): S {
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

Array<out T>.reduceIndexedIndexed(operation: (index: Int, acc: S, T) -&gt; S): S {
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

Array<out T>.reduceIndexedIndexedIndexed(operation: (index: Int, acc: S, T) -&gt; S): S {
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

Array<out T>.reduceIndexedIndexedIndexedIndexed(operation: (index: Int, acc: S, T) -&gt; S): S {
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

Array<out T>.reduceIndexedIndexedIndexedIndexedIndexed(operation: (index: Int, acc: S, T) -&gt; S): S {
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

Array<out T>.reduceIndexedIndexedIndexedIndexedIndexedIndexed(operation: (index: Int, acc: S, T) -&gt; S): S {
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
operation(index, accumulator, this![[index]])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {
    var accumulator = this[0] for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun <S, T : S> Array<out T>.reduceRight(operation: (T, acc: S) -> S): S {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun ByteArray.reduceRight(operation: (Byte, acc: Byte) -> Byte): Byte {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun ShortArray.reduceRight(operation: (Short, acc: Short) -> Short): Short {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun IntArray.reduceRight(operation: (Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun LongArray.reduceRight(operation: (Long, acc: Long) -> Long): Long {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun FloatArray.reduceRight(operation: (Float, acc: Float) -> Float): Float {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun DoubleArray.reduceRight(operation: (Double, acc: Double) -> Double): Double {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @return return accumulator
 */
public inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * and current accumulator value, and calculates the next accumulator value.
 * */
public inline fun <S, T : S> Array<out T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator: S = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * */
public inline fun ByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * */
public inline fun ShortArray.reduceRightIndexed(operation: (index: Int, Short, acc: Short) -> Short): Short {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * */
public inline fun IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * */
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * */
public inline fun FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * */
public inline fun DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element
 and applying [operation] from right to left
 * to each element with its index in the original array and current
 * @param [operation] function that takes the index of an element, the element itself
 * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun BooleanArray.reduceRightIndexed(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element
 and applying [operation] from right to left
 * to each element with its index in the original array and current
 * @param [operation] function that takes the index of an element, the element itself
 * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun CharArray.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Returns the sum of all values produced by [selector] function applied to
 * to each element in the array.
 */
public inline fun <T> Array<out T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to
each element in the array.
 */
public inline fun BooleanArray.sumBy(selector: (Boolean) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun CharArray.sumBy(selector: (Char) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ByteArray.sumByDouble(selector: (Byte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ShortArray.sumByDouble(selector: (Short) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun IntArray.sumByDouble(selector: (Int) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun LongArray.sumByDouble(selector: (Long) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun FloatArray.sumByDouble(selector: (Float) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun DoubleArray.sumByDouble(selector: (Double) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun CharArray.sumByDouble(selector: (Char) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun <T> Array<out T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {
    val first = ArrayList<T>()
    val second = ArrayList<T>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

Splits the original array into pair of lists, where *first* list contains elements for which [predicate] yielded *true*, while *second* list contains elements for which [predicate] yielded *false*. An original collection containing all the non-`null` elements, throwing an [IllegalArgumentOutOfRangeException] if there are any `null` elements.

```kotlin
public fun <T : Any> Array<T?>.requireNoNulls(): Array<T> {
    for (element in this)
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    return this as Array<T>
}
```

Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.

```kotlin
public inline fun CharArray.sumByDouble(selector: (Char) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun LongArray.sumByDouble(selector: (Long) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun DoubleArray.sumByDouble(selector: (Double) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun ByteArray.sumByDouble(selector: (Byte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun ShortArray.sumByDouble(selector: (Short) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun IntArray.sumByDouble(selector: (Int) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun FloatArray.sumByDouble(selector: (Float) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun Antelope<3814>.sumByDouble(selector: (Antelope<3814>) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        sum += selector(element)
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun {second.add(element) in this}
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun Boolean(: Boolean) -> Double): Double {
    var sum: Double = 0.0
    for (element in this)
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    return Pair(first, second)
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.

```kotlin
public inline fun {second.add(element) in this}
    return sum
}
```

Returns the sum of all values produced by [selector] function applied to each element in the array.
pair of lists, where *first* list contains elements for which [predicate] yielded `true`, while *second* list contains elements for which [predicate] yielded `false`. Splits the original array into pair of lists, where *first* list contains elements for which [predicate] yielded `true`, while *second* list contains elements for which [predicate] yielded `false`. Splits the original array into pair of lists, where *first* list contains elements for which [predicate] yielded `true`, while *second* list contains elements for which [predicate] yielded `false`.

```kotlin
/**
 * Splits the original array into pair of lists,
 * where *first* list contains elements for which [predicate] yielded `true`, while *second* list contains elements for which [predicate] yielded `false`.
 */
public inline fun DoubleArray.partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>> {
    val first = ArrayList<Double>()
    val second = ArrayList<Double>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original array into pair of lists,
 * where *first* list contains elements for which [predicate] yielded `true`, while *second* list contains elements for which [predicate] yielded `false`.
 */
public inline fun BooleanArray.partition(predicate: (Boolean) -> Boolean): Pair<List<Boolean>, List<Boolean>> {
    val first = ArrayList<Boolean>()
    val second = ArrayList<Boolean>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original array into pair of lists,
 * where *first* list contains elements for which [predicate] yielded `true`, while *second* list contains elements for which [predicate] yielded `false`.
 */
public inline fun CharArray.partition(predicate: (Char) -> Boolean): Pair<List<Char>, List<Char>> {
    val first = ArrayList<Char>()
    val second = ArrayList<Char>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterable
 */
public infix fun <T, R> Array<out T>.zip(other: Array<out R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterable
 */
public infix fun <R> ByteArray.zip(other: Array<out R>): List<Pair<Byte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterable
 */
public infix fun <R> ShortArray.zip(other: Array<out R>): List<Pair<Short, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterable
 */
public infix fun <R> IntArray.zip(other: Array<out R>): List<Pair<Int, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterable
 */
public infix fun <R> LongArray.zip(other: Array<out R>): List<Pair<Long, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterable
 */
public infix fun <R> FloatArray.zip(other: Array<out R>): List<Pair<Float, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterable
 */
public infix fun <R> DoubleArray.zip(other: Array<out R>): List<Pair<Double, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterable
 */
public infix fun <R> BooleanArray.zip(other: Array<out R>): List<Pair<Boolean, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public infix fun <R> CharArray.zip(other: Array<out R>): List<Pair<Char, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public infix fun <R> List.zip(other: List<out R>): List<Pair<R, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public infix fun <R> Array.zip(other: Array<out R>): List<Pair<R, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public infix fun <R> Iterable.zip(other: Iterable<out R>): List<Pair<R, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index using the provided [transform] function
 */
public infix fun <T> Iterable.transform(transform: (T) -> R): List<R> {
    return transform(this)
}
```
applied to each pair of elements. The returned list has length of the shortest collection. @sample

samples.collections.Iterables.Operations.zipIterableWithTransform

*\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zip(other: Array\-out R, transform: (a: T, b: R) -> V): List\<V> {\n val size = minOf(size, other.size)\n val list = ArrayList\<V\>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n } return list\}/**

\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zipIterableWithTransform

*\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zip(other: Array\-out R, transform: (a: T, b: R) -> V): List\<V> {\n val size = minOf(size, other.size)\n val list = ArrayList\<V\>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n } return list\}

\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zipIterableWithTransform

*\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zip(other: Array\-out R, transform: (a: T, b: R) -> V): List\<V> {\n val size = minOf(size, other.size)\n val list = ArrayList\<V\>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n } return list\}

\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zipIterableWithTransform

*\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zip(other: Array\-out R, transform: (a: T, b: R) -> V): List\<V> {\n val size = minOf(size, other.size)\n val list = ArrayList\<V\>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n } return list\}

\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zipIterableWithTransform

*\Apache inline fun \ skeptical <T, R, V> Array\-out T>.zip(other: Array\-out R, transform: (a: T, b: R) -> V): List\<V> {\n val size = minOf(size, other.size)\n val list = ArrayList\<V\>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n } return list\}
CharArray.zip(other: Array<out R>, transform: (a: Char, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
inline fun <T, R> Array<out T>.zip(other: Iterable<R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> Array<out T>.zip(other: Iterable<R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> ShortArray.zip(other: Iterable<R>): List<Pair<Short, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> IntArray.zip(other: Iterable<R>): List<Pair<Int, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> FloatArray.zip(other: Iterable<R>): List<Pair<Float, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> DoubleArray.zip(other: Iterable<R>): List<Pair<Double, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> BooleanArray.zip(other: Iterable<R>): List<Pair<Boolean, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> CharArray.zip(other: Iterable<R>): List<Pair<Char, R>> {
    return zip(other) { t1, t2 -> t1 to t2 } // A selection of the provided transform function
}
shortest collection.

*\n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

*\n*public inline fun \<R, V> ShortArray.zip(other: Iterable\\<\R>, transform: (a: Short, b: R) -> V): List\\<V> \{\n\narraySize = size\n\nval list = ArrayList\\<V\\>(minOf(other.collectionSizeOrDefault(10), arraySize))\n\nvar i = 0\n\nfor (element in other) {\n\nif (i >= arraySize) break\n\nlist.add(transform(this[i++], element))\n\n}\nreturn list\\n\n\n* Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* @sample
dsamples.collections.Iterables.Operations.zipIterableWithTransform

*/

public inline fun <R, V> ShortArray.zip(other: Iterable\\<R>, transform: (a: Short, b: R) -> V): List\\<V> \{\n\narraySize = size\n\nval list = ArrayList\\<V\\>(minOf(other.collectionSizeOrDefault(10), arraySize))\n\nvar i = 0\n\nfor (element in other) {\n\nif (i >= arraySize) break\n\nlist.add(transform(this[i++], element))\n\n}\nreturn list\\n\n\n* Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* @sample
dsamples.collections.Iterables.Operations.zipIterableWithTransform

*/

public inline fun <R, V> IntArray.zip(other: Iterable\\<R>, transform: (a: Int, b: R) -> V): List\\<V> \{\n\narraySize = size\n\nval list = ArrayList\\<V\\>(minOf(other.collectionSizeOrDefault(10), arraySize))\n\nvar i = 0\n\nfor (element in other) {\n\nif (i >= arraySize) break\n\nlist.add(transform(this[i++], element))\n\n}\nreturn list\\n\n\n* Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* @sample
dsamples.collections.Iterables.Operations.zipIterableWithTransform

*/

public inline fun <R, V> LongArray.zip(other: Iterable\\<R>, transform: (a: Long, b: R) -> V): List\\<V> \{\n\narraySize = size\n\nval list = ArrayList\\<V\\>(minOf(other.collectionSizeOrDefault(10), arraySize))\n\nvar i = 0\n\nfor (element in other) {\n\nif (i >= arraySize) break\n\nlist.add(transform(this[i++], element))\n\n}\nreturn list\\n\n\n* Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* @sample
dsamples.collections.Iterables.Operations.zipIterableWithTransform

*/

public inline fun <R, V> FloatArray.zip(other: Iterable\\<R>, transform: (a: Float, b: R) -> V): List\\<V> \{\n\narraySize = size\n\nval list = ArrayList\\<V\\>(minOf(other.collectionSizeOrDefault(10), arraySize))\n\nvar i = 0\n\nfor (element in other) {\n\nif (i >= arraySize) break\n\nlist.add(transform(this[i++], element))\n\n}\nreturn list\\n\n\n* Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* @sample
dsamples.collections.Iterables.Operations.zipIterableWithTransform

*/

public inline fun <R, V> DoubleArray.zip(other: Iterable\\<R>, transform: (a: Double, b: R) -> V): List\\<V> \{\n\narraySize = size\n\nval list = ArrayList\\<V\\>(minOf(other.collectionSizeOrDefault(10), arraySize))\n\nvar i = 0\n\nfor (element in other) {\n\nif (i >= arraySize) break\n\nlist.add(transform(this[i++], element))\n\n}\nreturn list\\n\n\n* Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* @sample
dsamples.collections.Iterables.Operations.zipIterableWithTransform

*/

public inline fun <R, V> BooleanArray.zip(other: Iterable\\<R>, transform: (a: Boolean, b: R) -> V): List\\<V> \{\n\narraySize = size\n\nval list = ArrayList\\<V\\>(minOf(other.collectionSizeOrDefault(10), arraySize))\n\nvar i = 0\n\nfor (element in other) {\n\nif (i >= arraySize) break\n\nlist.add(transform(this[i++], element))\n\n}\nreturn list\\n\n\n* Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* @sample
dsamples.collections.Iterables.Operations.zipIterableWithTransform

*/

public inline fun <R, V> CharArray.zip(other: Iterable\\<R>, transform: (a: Char, b: R) -> V): List\\<V> \{\n\narraySize = size\n\nval list = ArrayList\\<V\\>(minOf(other.collectionSizeOrDefault(10), arraySize))\n\nvar i = 0\n\nfor (element in other) {\n\nif (i >= arraySize) break\n\nlist.add(transform(this[i++], element))\n\n}\nreturn list\\n\n\n* Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* @sample
dsamples.collections.Iterables.Operations.zipIterableWithTransform

*/

open infix fun ByteArray.zip(other: ByteArray): List\\<Pair\\<Byte, Byte>> \{\n\nreturn zip(other) { t1, t2 -> t1 to t2 }\n\n\n* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n* The returned list has length of the shortest
collection.*  
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public infix fun ShortArray.zip(other: ShortArray): List<Pair<Short, Short>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}  

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public infix fun IntArray.zip(other: IntArray): List<Pair<Int, Int>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}  

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public infix fun LongArray.zip(other: LongArray): List<Pair<Long, Long>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}  

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public infix fun FloatArray.zip(other: FloatArray): List<Pair<Float, Float>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}  

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public infix fun DoubleArray.zip(other: DoubleArray): List<Pair<Double, Double>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}  

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public infix fun BooleanArray.zip(other: BooleanArray): List<Pair<Boolean, Boolean>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}  

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public infix fun CharArray.zip(other: CharArray): List<Pair<Char, Char>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}  

* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public inline fun <V> ByteArray.zip(other: ByteArray, transform: (a: Byte, b: Byte) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}  

* Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.
* The returned list has length of the shortest array.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public inline fun <V> ShortArray.zip(other: ShortArray, transform: (a: Short, b: Short) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}  

* Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.
* The returned list has length of the shortest array.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public inline fun <V> IntArray.zip(other: IntArray, transform: (a: Int, b: Int) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}  

* Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.
* The returned list has length of the shortest array.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public inline fun <V> LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}  

* Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.
* The returned list has length of the shortest array.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public inline fun <V> DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}  

* Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.
* The returned list has length of the shortest array.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public inline fun <V> BooleanArray.zip(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}  

* Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.
* The returned list has length of the shortest array.
* @sample samples.collections.Iterables.Operations.zipIterable
*/

public inline fun <V> CharArray.zip(other: CharArray, transform: (a: Char, b: Char) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}  

* Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.
* The returned list has length of the shortest array.
* @sample samples.collections.Iterables.Operations.zipIterable
*/
samples.collections.Iterables.Operations.zipIterableWithTransform\n*/\npublic inline fun <V> FloatArray.zip(other: FloatArray, transform: (a: Float, b: Float) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n
public inline fun <V> DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n
public inline fun <V> BooleanArray.zip(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n
public inline fun <V> CharArray.zip(other: CharArray, transform: (a: Char, b: Char) -> V): List<V> {\n    val size = minOf(size, other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}\n
public fun <T, A : Appendable> Array<out T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {\n    buffer.append(prefix)\n    var count = 0\n    for (element in this) {\n        if (++count > 1) buffer.append(separator)\n        if (limit < 0 || count <= limit) {\n            buffer.appendElement(element, transform)\n        } else break\n    }\n    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n    return buffer
}
```java
buffer.append(transform(element))
else
    buffer.append(element.toString())
}

if (limit >= 0 && count > limit) buffer.append(truncated())
buffer.append(postfix())
return

buffer.append(separator) if (limit < 0 || count <= limit) {
    buffer.append(transform(element))
    else
        buffer.append(element.toString())
}

if (limit >= 0 && count > limit) buffer.append(truncated())
buffer.append(postfix())
return

buffer.append(separator) if (limit < 0 || count <= limit) {
    buffer.append(transform(element))
    else
        buffer.append(element.toString())
}

if (limit >= 0 && count > limit) buffer.append(truncated())
buffer.append(postfix())
return

buffer.append(separator) if (limit < 0 || count <= limit) {
    buffer.append(transform(element))
    else
        buffer.append(element.toString())
}

if (limit >= 0 && count > limit) buffer.append(truncated())
buffer.append(postfix())
return
```

The code above is a method that appends the string from all the elements separated using a separator and using the given [prefix] and [postfix] if supplied. If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").

The method is defined as follows:

```java
public fun <A : Appendable> IntArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Int) -> CharSequence)? = null): A {
    // Method implementation...
}
```

```java
public fun <A : Appendable> LongArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Long) -> CharSequence)? = null): A {
    // Method implementation...
}
```

```java
public fun <A : Appendable> FloatArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Float) -> CharSequence)? = null): A {
    // Method implementation...
}
```

```java
public fun <A : Appendable> DoubleArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Double) -> CharSequence)? = null): A {
    // Method implementation...
}
```

```java
public fun <A : Appendable> BooleanArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Boolean) ->
```
A "buffer.append(prefix)\n    var count = 0\n    for (element in this) {
        if (++count > 1) buffer.append(separator)\n            if (limit < 0 || count <= limit) {\n                if (transform != null) {\n                    buffer.append(transform(element))\n                } else {\n                    buffer.append(element.toString())\n                }\n            } else break\n        }\n    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n    return buffer\n"

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...".)\n
 * @sample samples.collections.Collections.Transformations.joinTo\n */
public fun <A : Appendable> CharArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Char) -> CharSequence)? = null): A {
    buffer.append(prefix)\n    var count = 0\n    for (element in this) {
        if (++count > 1) buffer.append(separator)\n            if (limit < 0 || count <= limit) {\n                if (transform != null) {\n                    buffer.append(transform(element))\n                } else {\n                    buffer.append(element)\n                }\n            } else break\n        }
    if (limit >= 0 && count > limit) buffer.append(truncated)\n    buffer.append(postfix)\n    return buffer
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...".)\n
 * @sample samples.collections.Collections.Transformations.joinToString\n */
public fun <T> Array<out T>.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...".)\n
 * @sample samples.collections.Collections.Transformations.joinToString\n */
public fun ByteArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Byte) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...".)\n
 * @sample samples.collections.Collections.Transformations.joinToString\n */
public fun ShortArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Short) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...".)\n
 * @sample samples.collections.Collections.Transformations.joinToString\n */
public fun IntArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Int) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...".)\n
 * @sample samples.collections.Collections.Transformations.joinToString\n */
public fun LongArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Long) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
separators, prefix, postfix, limit, truncated, transform).toString() Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to \"...").

```kotlin
public fun FloatArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Float) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}
```

```kotlin
public fun DoubleArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Double) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}
```

```kotlin
public fun BooleanArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Boolean) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}
```

```kotlin
public fun CharArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Char) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}
```

```kotlin
public fun <T> Array<out T>.asIterable(): Iterable<T> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
```

```kotlin
public fun ByteArray.asIterable(): Iterable<Byte> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
```

```kotlin
public fun ShortArray.asIterable(): Iterable<Short> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
```

```kotlin
public fun IntArray.asIterable(): Iterable<Int> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
```

```kotlin
public fun LongArray.asIterable(): Iterable<Long> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
```

```kotlin
public fun FloatArray.asIterable(): Iterable<Float> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
```

```kotlin
public fun DoubleArray.asIterable(): Iterable<Double> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
```

```kotlin
public fun BooleanArray.asIterable(): Iterable<Boolean> {
    if (isEmpty()) return emptyList()
    return Iterable { this.iterator() }
}
```

```kotlin
public fun CharArray.asIterable(): Iterable<Char> {
    if
(isEmpty()) return emptyList() // return Iterable { this.iterator() } // \n // * @sample samples.collections.Sequences.Building.sequenceFromArray

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 * @sample samples.collections.Sequences.Building.sequenceFromArray
 */
public fun ByteArray.asSequence(): Sequence<Byte> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 * @sample samples.collections.Sequences.Building.sequenceFromArray
 */
public fun ShortArray.asSequence(): Sequence<Short> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 * @sample samples.collections.Sequences.Building.sequenceFromArray
 */
public fun IntArray.asSequence(): Sequence<Int> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 * @sample samples.collections.Sequences.Building.sequenceFromArray
 */
public fun LongArray.asSequence(): Sequence<Long> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 * @sample samples.collections.Sequences.Building.sequenceFromArray
 */
public fun FloatArray.asSequence(): Sequence<Float> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 * @sample samples.collections.Sequences.Building.sequenceFromArray
 */
public fun DoubleArray.asSequence(): Sequence<Double> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 * @sample samples.collections.Sequences.Building.sequenceFromArray
 */
public fun BooleanArray.asSequence(): Sequence<Boolean> {
    if (isEmpty()) return emptySequence()
    return Sequence { this.iterator() }
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfByte")
public fun Array<out Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfShort")
public fun Array<out Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfInt")
public fun Array<out Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Array<out Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Array<out Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfDouble")
public fun Array<out Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfByte")
public fun Array<out Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfShort")
public fun Array<out Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfInt")
public fun Array<out Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Array<out Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Array<out Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfDouble")
public fun Array<out Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}
this) { return if (count == 0) Double.NaN else sum / count /n}

/**
 * Returns an average value of elements in the array.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun ShortArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun IntArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun LongArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun FloatArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun DoubleArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Array<out Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Array<out Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Array<out Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Array<out Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Array<out Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Array<out Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns an element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Random.random(): Int {
    return random(Random)
}

/**
 * Returns an element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Random.random(): Long {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Random.random(): Double {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Random.random(): Float {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Random.random(): Long {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Random.random(): Double {
    return random(Random)
}
inline operator fun IntRange.contains(element: Int?): Boolean {
    return element != null && contains(element)
}

/**
 * Returns `true` if this range
 * contains the specified 
 * [element].
 * 
 * Always returns `false` if the 
 * [element] is `null`.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline operator fun LongRange.contains(element: Long?): Boolean {
    return element != null && contains(element)
}

/**
 * Returns `true` if this range
 * contains the specified 
 * [element].
 * 
 * Always returns `false` if the 
 * [element] is `null`.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline operator fun CharRange.contains(element: Char?): Boolean {
    return element != null && contains(element)
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Byte): Boolean {
    return contains(value.toInt())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Byte): Boolean {
    return contains(value.toLong())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Byte): Boolean {
    return contains(value.toShort())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("doubleRangeContains")
public operator fun ClosedRange<Double>.contains(value: Byte): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Byte): Boolean {
    return contains(value.toFloat())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Double): Boolean {
    return value.toIntExactOrNull().let { if (it != null) contains(it) else false }
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Double): Boolean {
    return value.toLongExactOrNull().let { if (it != null) contains(it) else false }
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Double): Boolean {
    return contains(value.toShort())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("doubleRangeContains")
public operator fun ClosedRange<Double>.contains(value: Double): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Double): Boolean {
    return contains(value.toFloat())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Float): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Float): Boolean {
    return contains(value.toLong())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Float): Boolean {
    return contains(value.toShort())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("doubleRangeContains")
public operator fun ClosedRange<Double>.contains(value: Float): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified 
 * [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Float): Boolean {
    return contains(value.toFloat())
}
and is going to be removed.

```kotlin
@kotlin.jvm.JvmName("byteRangeContains")
public operator fun ClosedRange<Byte>.contains(value: Double): Boolean {
    return value.toByteArray().let { if (it != null) contains(it) else false }
}

@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Double): Boolean {
    return value.toShort().let { if (it != null) contains(it) else false }
}

@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Double): Boolean {
    return contains(value.toFloat())
}

@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Float): Boolean {
    return value.toInt().let { if (it != null) contains(it) else false }
}

@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Float): Boolean {
    return contains(value.toLong())
}
```

Open Source Used In DNAC Platform Antelope 3827
operator fun ClosedRange<Short>.contains(value: Long): Boolean {
    return value.toShortExactOrNull().let { if (it != null) contains(it) else false }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("doubleRangeContains")
public operator fun
ClosedRange<Double>.contains(value: Long): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun
ClosedRange<Float>.contains(value: Long): Boolean {
    return contains(value.toFloat())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("intRangeContains")
public operator fun
ClosedRange<Int>.contains(value: Short): Boolean {
    return contains(value.toInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("longRangeContains")
public operator fun
ClosedRange<Long>.contains(value: Short): Boolean {
    return contains(value.toLong())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("byteRangeContains")
public operator
fun ClosedRange<Byte>.contains(value: Short): Boolean {
    return value.toByteExactOrNull().let { if (it != null) contains(it) else false }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("doubleRangeContains")
public operator fun
ClosedRange<Double>.contains(value: Short): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun
ClosedRange<Float>.contains(value: Short): Boolean {
    return contains(value.toFloat())
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 */
public infix fun Int.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this, to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 */
public infix fun Long.downTo(to: Byte): LongProgression {
    return LongProgression.fromClosedRange(this, to.toLong(), -1L)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 */
public infix fun Byte.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 */
public infix fun Short.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 */
public infix fun Char.downTo(to: Char): CharProgression {
    return CharProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 */
public infix fun Int.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 */
public infix fun Long.downTo(to: Int): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1L)
}
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public infix fun Byte.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1
}
```

Returns a progression from this value down to the specified [to] value with the step -1
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public infix fun Short.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1
}
```

Returns a progression from this value down to the specified [to] value with the step -1
* The [to] value should be less than or equal to 'this' value.
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public infix fun Int.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1L
}
```

Returns a progression from this value down to the specified [to] value with the step -1
* The [to] value should be less than or equal to 'this' value.
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public infix fun Long.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1L
}
```

Returns a progression from this value down to the specified [to] value with the step -1
* The [to] value should be less than or equal to 'this' value.
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public infix fun Byte.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this.toLong(), to, -1L
}
```

Returns a progression from this value down to the specified [to] value with the step -1
* The [to] value should be less than or equal to 'this' value.
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public infix fun Short.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this.toLong(), to, -1L
}
```

Returns a progression from this value down to the specified [to] value with the step -1
* The [to] value should be less than or equal to 'this' value.
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public infix fun Int.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this, to.toInt(), -1
}
```

Returns a progression from this value down to the specified [to] value with the step -1
* The [to] value should be less than or equal to 'this' value.
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public infix fun Long.downTo(to: Short): LongProgression {
    return LongProgression.fromClosedRange(this, to.toLong(), -1L
}
```

Returns a progression from this value down to the specified [to] value with the step -1
* The [to] value should be less than or equal to 'this' value.
* If the [to] value is greater than 'this' value the returned progression is empty.

```kotlin
public fun IntProgression.reversed(): IntProgression {
    return IntProgression.fromClosedRange(last, first, -step)
}
```

Returns a progression that goes over the same range in the opposite direction with the same step.

```kotlin
public fun LongProgression.reversed(): LongProgression {
    return LongProgression.fromClosedRange(last, first, -step)
}
```

Returns a progression that goes over the same range in the opposite direction with the same step.

```kotlin
public fun CharProgression.reversed(): CharProgression {
    return CharProgression.fromClosedRange(first, last, -step)
}
```

Returns a progression that goes over the same range in the opposite direction with the same step.

```kotlin
internal fun Int.toByteExactOrNull(): Byte? {
    return if (this in
```
Byte.MIN_VALUE.toInt().Byte.MAX_VALUE.toInt() this.toByteArray() else null

Long.toByteArrayOrNull(): Byte? { null return if (this in Byte.MIN_VALUE.toLong().Byte.MAX_VALUE.toLong()) this.toByteArray() else null

Short.toByteArrayOrNull(): Byte? { null return if (this in Byte.MIN_VALUE.toShort().Byte.MAX_VALUE.toShort()) this.toByteArray() else null

Double.toByteArrayOrNull(): Byte? { null return if (this in Byte.MIN_VALUE.toDouble().Byte.MAX_VALUE.toDouble()) this.toByteArray() else null

Float.toByteArrayOrNull(): Byte? { null return if (this in Byte.MIN_VALUE.toFloat().Byte.MAX_VALUE.toFloat()) this.toByteArray() else null

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 * *}
1).toInt()

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 */
public infix fun Int.until(to: Long): LongRange {
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 */
public infix fun Long.until(to: Long): LongRange {
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 */
public infix fun Byte.until(to: Long): IntRange {
    return this.toInt() .. (to.toInt() - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 */
public infix fun Short.until(to: Long): LongRange {
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toInt()
}

/**
 * Ensures that this value is not less than the specified [minimumValue].
 * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.
 */
public fun <T : Comparable<T>> T.coerceAtLeast(minimumValue: T): T {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not less than the specified [minimumValue].
 * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.
 */
public fun Byte.coerceAtLeast(minimumValue: Byte): Byte {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not less than the specified [minimumValue].
 * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.
 */
public fun Short.coerceAtLeast(minimumValue: Short): Short {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not less than the specified [minimumValue].
 * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.
 */
public fun Int.coerceAtLeast(minimumValue: Int): Int {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not less than the specified [minimumValue].
 * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.
 */
public fun Long.coerceAtLeast(minimumValue: Long): Long {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not less than the specified [minimumValue].
 * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.
 */
public fun Float.coerceAtLeast(minimumValue: Float): Float {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not less than the specified [minimumValue].
 * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.
 */
public fun Double.coerceAtLeast(minimumValue: Double): Double {
    return if (this < minimumValue) minimumValue else this
}
Ensures that this value is not greater than the specified \[\text{maximumValue}\].

### Byte

```kotlin
public fun Byte.coerceAtMost(maximumValue: Byte): Byte {
    return if (this > maximumValue) maximumValue else this
}
```

### Short

```kotlin
public fun Short.coerceAtMost(maximumValue: Short): Short {
    return if (this > maximumValue) maximumValue else this
}
```

### Int

```kotlin
public fun Int.coerceAtMost(maximumValue: Int): Int {
    return if (this > maximumValue) maximumValue else this
}
```

### Long

```kotlin
public fun Long.coerceAtMost(maximumValue: Long): Long {
    return if (this > maximumValue) maximumValue else this
}
```

### Float

```kotlin
public fun Float.coerceAtMost(maximumValue: Float): Float {
    return if (this > maximumValue) maximumValue else this
}
```

### Double

```kotlin
public fun Double.coerceAtMost(maximumValue: Double): Double {
    return if (this > maximumValue) maximumValue else this
}
```

### Comparable

```kotlin
public fun <T : Comparable<T>> T.coerceAtMost(maximumValue: T): T {
    return if (this > maximumValue) maximumValue else this
}
```

### In

```kotlin
public fun T.coerceIn(minimumValue: T?, maximumValue: T?): T {
    if (minimumValue !== null && maximumValue !== null) {
        if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
        if (this < minimumValue) return minimumValue
        if (this > maximumValue) return maximumValue
    } else {
        if (minimumValue !== null && this < minimumValue) return minimumValue
        if (maximumValue !== null && this > maximumValue) return maximumValue
    }
    return this
}
```

IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")

### Byte

```kotlin
public fun Byte.coerceIn(minimumValue: Byte, maximumValue: Byte): Byte {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    return if (this < minimumValue) minimumValue
    else if (this > maximumValue) maximumValue
    else this
}
```

IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")

### Short

```kotlin
public fun Short.coerceIn(minimumValue: Short, maximumValue: Short): Short {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    return if (this < minimumValue) minimumValue
    else if (this > maximumValue) maximumValue
    else this
}
```

IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")

### Int

```kotlin
public fun Int.coerceIn(minimumValue: Int, maximumValue: Int): Int {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    return if (this < minimumValue) minimumValue
    else if (this > maximumValue) maximumValue
    else this
}
```

IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")

### Long

```kotlin
public fun Long.coerceIn(minimumValue: Long, maximumValue: Long): Long {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    return if (this < minimumValue) minimumValue
    else if (this > maximumValue) maximumValue
    else this
}
```

IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue) return maximumValue\n return this

@SinceKotlin("1.1")
public fun <T : Comparable<T>> T.coerceIn(range: ClosedFloatingPointRange<T>): T {
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
    return when {
        range.lessThanOrEquals(this, range.start) && !range.lessThanOrEquals(range.start, this) -> range.start
        range.lessThanOrEquals(range.endInclusive, this) && !range.lessThanOrEquals(this, range.endInclusive) ->
            range.endInclusive
        else -> this // this < start equiv to this <= start && !(this >= start)
    }
}

public fun <T : Comparable<T>> T.coerceIn(range: ClosedRange<T>): T {
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }
}
*/
public fun Int.coerceIn(range: ClosedRange<Int>): Int {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<Int>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }
}

/**
 * Ensures that this value lies in the specified 
 * 
 * @return this value if it's in the [range], or 'range.start' if this value is less than 'range.start', or 'range.endInclusive' if this value is greater than 'range.endInclusive'
 * 
 * @sample
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

kotlin.experimental.*

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
Since Kotlin 1.3
@ExperimentalUnsignedTypes
public inline class UByte @PublishedApi internal constructor(@PublishedApi internal val data: Byte) : Comparable<UByte> {

    companion object {
        /**
         * A constant holding the minimum value an instance of UByte can have.
         */
        public const val MIN_VALUE: UByte = UByte(0)

        /**
         * A constant holding the maximum value an instance of UByte can have.
         */
        public const val MAX_VALUE: UByte = UByte(-1)

        /**
         * The number of bytes used to represent an instance of UByte in a binary form.
         */
        public const val SIZE_BYTES: Int = 1

        /**
         * The number of bits used to represent an instance of UByte in a binary form.
         */
        public const val SIZE_BITS: Int = 8

        @kotlin.internal.InlineOnly
        public inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())

        @kotlin.internal.InlineOnly
        public inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())

        @kotlin.internal.InlineOnly
        public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)

        @kotlin.internal.InlineOnly
        public inline operator fun compareTo(other: ULong): ULong = this.toULong().compareTo(other)

        @kotlin.internal.InlineOnly
        public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())

        @kotlin.internal.InlineOnly
        public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())

        @kotlin.internal.InlineOnly
        public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)

        @kotlin.internal.InlineOnly
        public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)

        @kotlin.internal.InlineOnly
        public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())

        @kotlin.internal.InlineOnly
        public inline operator fun minus(other: UShort): UInt = this.toUInt().minus(other.toUInt())

        @kotlin.internal.InlineOnly
        public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)

        @kotlin.internal.InlineOnly
        public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)

        @kotlin.internal.InlineOnly
        public inline operator fun times(other: UByte): UByte = this.toULong().times(other)

        @kotlin.internal.InlineOnly
        public inline operator fun times(other: UShort): UByte = this.toULong().times(other)

        @kotlin.internal.InlineOnly
        public inline operator fun times(other: UInt): UByte = this.toULong().times(other)

        @kotlin.internal.InlineOnly
        public inline operator fun times(other: ULong): UByte = this.toULong().times(other)

        @kotlin.internal.InlineOnly
        public inline operator fun div(other: UByte): UInt = this.toULong().div(other)

        @kotlin.internal.InlineOnly
        public inline operator fun div(other: UShort): UInt = this.toULong().div(other)

        @kotlin.internal.InlineOnly
        public inline operator fun div(other: UInt): UInt = this.toULong().div(other)

        @kotlin.internal.InlineOnly
        public inline operator fun div(other: ULong): ULong = this.toULong().div(other)

        @kotlin.internal.InlineOnly
        public inline operator fun mod(other: UByte): UInt = this.toULong().mod(other)

        @kotlin.internal.InlineOnly
        public inline operator fun mod(other: UShort): UInt = this.toULong().mod(other)

        @kotlin.internal.InlineOnly
        public inline operator fun mod(other: UInt): UInt = this.toULong().mod(other)

        @kotlin.internal.InlineOnly
        public inline operator fun mod(other: ULong): ULong = this.toULong().mod(other)
    }
}

Open Source Used In DNAC Platform Antelope 3834
@kotlin.internal.InlineOnly
public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt())
/** Multiplies this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun times(other: UShort): UInt = this.toUInt().times(other.toUInt())
/** Multiplies this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)
/** Multiplies this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun times(other: ULong): ULong = this.toULong().times(other)
/** Divides this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())
/** Divides this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())
/** Divides this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UInt = this.toUInt().div(other)
/** Divides this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): ULong = this.toULong().div(other)
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other.toUInt())
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other)
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun inc(): UByte = UByte(data.inc())
/** Increments this value. */

@kotlin.internal.InlineOnly
public inline operator fun dec(): UByte = UByte(data.dec())
/** Decrements this value. */

@kotlin.internal.InlineOnly
operator fun rangeTo(other: UByte): UIntRange = UIntRange(this.toUInt(), other.toUInt())
/** Creates a range from this value to the specified [other] value. */

@kotlin.internal.InlineOnly
infix fun and(other: UByte): UByte = UByte(this.data and other.data)
/** Performs a bitwise AND operation between the two values. */

@kotlin.internal.InlineOnly
infix fun or(other: UByte): UByte = UByte(this.data or other.data)
/** Performs a bitwise OR operation between the two values. */

@kotlin.internal.InlineOnly
infix fun xor(other: UByte): UByte = UByte(this.data xor other.data)
/** Performs a bitwise XOR operation between the two values. */

@kotlin.internal.InlineOnly
public inline fun inv(): UByte = UByte(data.inv())
/** Inverts the bits in this value. */

@kotlin.internal.InlineOnly
public inline fun toByte(): Byte = data
/** Converts this [UByte] value to [Byte]. */

@kotlin.internal.InlineOnly
public inline fun toShort(): Short = data.toShort() and 0xFF
/** Converts this [UByte] value to [Short]. */

@kotlin.internal.InlineOnly
public inline fun toInt(): Int = data.toInt() and 0xFF
/** Converts this [UByte] value to [Int]. */

@kotlin.internal.InlineOnly
public inline fun toLong(): Long = data.toLong() and 0xFF
/** Converts this [UByte] value to [Long]. */

@kotlin.internal.InlineOnly
public inline fun toUByte(): UByte = this
/** Converts this [UByte] value to [UByte]. */

@kotlin.internal.InlineOnly
public inline fun toUShort(): UShort = this.toUByte().toUShort() and 0xFFFF
/** Converts this [UByte] value to [UShort]. */

@kotlin.internal.InlineOnly
public inline fun toULong(): ULong = this.toUByte().toULong() and 0xFFFFFFFF
/** Converts this [UByte] value to [ULong]. */
'UShort' value are the same as the bits of this 'UByte' value, whereas the most significant 8 bits are filled with zeros.

@kotlin.internal.InlineOnly
public inline fun toUShort(): UShort = UShort(data.toShort() and 0xFF)

/**
 * Converts this [UByte] value to [UInt].
 *
 * The resulting `UInt` value represents the same numerical value as this `UByte`.
 * The least significant 8 bits of the resulting `UInt` value are the same as the bits of this `UByte` value, whereas the most significant 24 bits are filled with zeros.
 */
@kotlin.internal.InlineOnly
public inline fun toUInt(): UInt = UInt(data.toInt() and 0xFF)

/**
 * Converts this [UByte] value to [ULong].
 *
 * The resulting `ULong` value represents the same numerical value as this `UByte`.
 * The least significant 8 bits of the resulting `ULong` value are the same as the bits of this `UByte` value, whereas the most significant 56 bits are filled with zeros.
 */
@kotlin.internal.InlineOnly
public inline fun toULong(): ULong = ULong(data.toLong() and 0xFF)

/**
 * Converts this [UByte] value to [Float].
 */
@kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toInt().toFloat()

/**
 * Converts this [UByte] value to [Double].
 */
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double = this.toInt().toDouble()

public override fun toString(): String = toInt().toString()

/**
 * Converts this [Byte] value to [UByte].
 *
 * If this value is positive, the resulting `UByte` value represents the same numerical value as this `Byte`.
 *
 * The resulting `UByte` value has the same binary representation as this `Byte` value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Byte.toUByte(): UByte = UByte(this)

/**
 * Converts this [Short] value to [UByte].
 *
 * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `Short`.
 * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `Int`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Short.toUByte(): UByte = UByte(this.toByte())

/**
 * Converts this [Int] value to [UByte].
 *
 * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `Long`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Int.toUByte(): UByte = UByte(this.toByte())

/**
 * Converts this [Long] value to [UByte].
 *
 * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `Long`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Long.toUByte(): UByte = UByte(this.toByte())

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// Auto-generated file. DO NOT EDIT!

package kotlin

import kotlin.experimental.*

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UInt @PublishedApi internal constructor(@PublishedApi internal val data: Int) : Comparable<UInt> {

    companion object {
        /**
         * A constant holding the minimum value an instance of UInt can have.
         */
        public const val MIN_VALUE: UInt = UInt(0)

        /**
         * A constant holding the maximum value an instance of UInt can have.
         */
        public const val MAX_VALUE: UInt = UInt(-1)

        /**
         * The number of bytes used to represent an instance of UInt in a binary form.
         */
        public const val SIZE_BYTES: Int = 4

        /**
         * The number of bits used to represent an instance of UInt in a binary form.
         */
        public const val SIZE_BITS: Int = 32

        @kotlin.internal.InlineOnly
        public inline operator fun compareTo(other: UByte): Int = this.compareTo(other.toUInt())

    }

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it’s less than other, or a positive number if it’s greater than other.
     */
    @kotlin.internal.InlineOnly
    public fun compareTo(other: UByte): Int = this.compareTo(other.toUInt())

    public override fun toString(): String = toInt().toString()

}
public inline operator fun compareTo(other: UShort): Int = this.compareTo(other.toUInt())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UInt): Int = uintCompare(this.data, other.data)

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    @Suppress("OVERRIDE_BY_INLINE")
    public override inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)

    /**
     * Adds the other value to this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UByte): UInt = this.plus(other.toUInt())
    /**
     * Adds the other value to this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UShort): UInt = this.plus(other.toUInt())
    /**
     * Adds the other value to this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UInt): UInt = UInt(this.data.plus(other.data))
    /**
     * Adds the other value to this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)

    /**
     * Subtracts the other value from this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UByte): UInt = this.minus(other.toUInt())
    /**
     * Subtracts the other value from this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UShort): UInt = this.minus(other.toUInt())
    /**
     * Subtracts the other value from this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UInt): UInt = UInt(this.data.minus(other.data))
    /**
     * Subtracts the other value from this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)

    /**
     * Multiplies this value by the other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UByte): UInt = this.times(other.toUInt())
    /**
     * Multiplies this value by the other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UShort): UInt = this.times(other.toUInt())
    /**
     * Multiplies this value by the other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UInt): UInt = UInt(this.data.times(other.data))
    /**
     * Multiplies this value by the other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: ULong): ULong = this.toULong().times(other)

    /**
     * Calculates the remainder of dividing this value by the other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UByte): UInt = this.rem(other.toUInt())
    /**
     * Calculates the remainder of dividing this value by the other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UShort): UInt = this.rem(other.toUInt())
    /**
     * Calculates the remainder of dividing this value by the other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UInt): UInt = uintRemainder(this, other)
    /**
     * Calculates the remainder of dividing this value by the other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)

    /**
     * Increments this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun inc(): UInt = UInt(data.inc())
    /**
     * Decrements this value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun dec(): UInt = UInt(data.dec())

    /**
     * Creates a range from this value to the specified other value.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun rangeTo(other: UInt): UIntRange = UIntRange(this, other)

    /**
     * Shifts this value left by the [bitCount] number of bits, filling the leftmost bits with zeros.
     */
    * /\n@kotlin.internal.InlineOnly
    public inline infix fun shl(bitCount: Int): UInt = UInt(data shl bitCount)
    /**
     * Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros.
     */
    * /\n@kotlin.internal.InlineOnly
    public inline infix fun shr(bitCount: Int): UInt = UInt(data shr bitCount)
bits are filled with the sign bit of this value.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly

public inline fun Short.toUInt(): UInt = UInt(this.toInt())

*/

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Int.toUInt(): UInt = UInt(this)

/**
 * Converts this \[Int\] value to \[UInt\].
 *
 * If this value is positive, the resulting `UInt` value represents the same numerical value as this `Int`.
 * The resulting `UInt` value has the same binary representation as this `Int` value.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Long.toUInt(): UInt = UInt(this)

/**
 * Converts this \[Long\] value to \[UInt\].
 *
 * If this value is positive and less than or equal to \[UInt.MAX_VALUE\], the resulting `UInt` value represents the same numerical value as this `Long`.
 * The resulting `UInt` value is represented by the least significant 32 bits of this `Long` value.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Float.toUInt(): UInt = doubleToUInt(this)

/**
 * Converts this \[Float\] value to \[UInt\].
 *
 * The fractional part, if any, is rounded down towards zero.
 * Returns zero if this `Float` value is negative or `NaN`, \[UInt.MAX_VALUE\] if it's bigger than \[UInt.MAX_VALUE\].
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Double.toUInt(): UInt = doubleToUInt(this)

/**
 * Converts this \[Double\] value to \[UInt\].
 *
 * The fractional part, if any, is rounded down towards zero.
 * Returns zero if this `Double` value is negative or `NaN`, \[UInt.MAX_VALUE\] if it's bigger than \[UInt.MAX_VALUE\].
 */

package kotlin

import kotlin.experimental.*

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UShort @PublishedApi internal constructor(@PublishedApi internal val data: Short) : Comparable<UShort> {

    companion object {
        /**
         * A constant holding the minimum value an instance of UShort can have.
         */
        public const val MIN_VALUE: UShort = UShort(0)

        /**
         * A constant holding the maximum value an instance of UShort can have.
         */
        public const val MAX_VALUE: UShort = UShort(-1)

        /**
         * The number of bytes used to represent an instance of UShort in a binary form.
         */
        public const val SIZE_BYTES: Int = 2

        /**
         * The number of bits used to represent an instance of UShort in a binary form.
         */
        public const val SIZE_BITS: Int = 16
    }

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = this.toDouble().compareTo(other.toDouble())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other.toUInt())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: ULong): UInt = this.toULong().plus(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UByte): Int = this.toDouble().compareTo(other.toDouble())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UShort): Int = this.toDouble().compareTo(other.toDouble())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UInt): Int = this.toDouble().compareTo(other.toDouble())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = this.toDouble().compareTo(other.toDouble())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = this.toDouble().compareTo(other.toDouble())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
     * or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = this.toDouble().compareTo(other.toDouble())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,
value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: ULong): ULong =
    this.toULong().plus(other) /* Subtract the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UByte): UInt =
    this.toUInt().minus(other.toUInt()) /* Subtract the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UShort): UInt =
    this.toUInt().minus(other.toUInt()) /* Subtract the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UShort): UInt =
    this.toUInt().plus(other) /* Subtract the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: ULong): ULong =
    this.toULong().minus(other).toULong() /* Multiply this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UByte): UInt =
    this.toUInt().div(other.toUInt()) /* Divide this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UShort): UInt =
    this.toUInt().div(other.toUInt()) /* Divide this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: ULong): ULong =
    this.toULong().div(other) /* Divide this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UByte): UInt =
    this.toUInt().rem(other.toUInt()) /* Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UShort): UInt =
    this.toUInt().rem(other.toUInt()) /* Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: ULong): ULong =
    this.toULong().rem(other) /* Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun inc(): UShort = UShort(data.inc()) /* Increments this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun dec(): UShort = UShort(data.dec()) /* Decrements this value. */

    /**
     * Converts this `UShort` value to `Byte`.
     *
     * If this value is less than or equals to `Byte.MAX_VALUE`, the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * The resulting `Byte` value may be negative.
     * * This value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
     * * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value
     * represents the same numerical value as this `UShort`.
     * * Note that the resulting `Byte` value may be negative.
value are the same as the bits of this `UShort` value,
* whereas the most significant 16 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 16 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 48 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
* whereas the most significant 8 bits are filled with zeros.
* A `UShort` value is valid if its bits are the same as the bits of this `UShort` value,
override fun hashCode(): Int =
    if (isEmpty()) -1 else (31 * first.toInt() + last.toInt())

override fun toString(): String = "$first..$last"

companion object {
    /** An empty range of values of type Int. */
    public val EMPTY: CharRange = CharRange(1.toChar(), 0.toChar())
}

/**
 * A range of values of type `Int`. *
 */
public class IntRange(start: Int, endInclusive: Int) : IntProgression(start, endInclusive, 1), ClosedRange<Int> {
    override val start: Int get() = first
    override val endInclusive: Int get() = last

    override fun contains(value: Int): Boolean = first <= value && value <= last

    override fun isEmpty(): Boolean = first > last

    override fun equals(other: Any?): Boolean =
        other is IntRange && (isEmpty() && other.isEmpty() ||
        first == other.first && last == other.last)

    override fun hashCode(): Int =
        if (isEmpty()) -1 else (31 * first + last)

    override fun toString(): String = "$first..$last"

    companion object {
        /** An empty range of values of type Int. */
        public val EMPTY: IntRange = IntRange(1, 0)
    }
}

/**
 * A range of values of type `Long`. *
 */
public class LongRange(start: Long, endInclusive: Long) : LongProgression(start, endInclusive, 1), ClosedRange<Long> {
    override val start: Long get() = first
    override val endInclusive: Long get() = last

    override fun contains(value: Long): Boolean = first <= value && value <= last

    override fun isEmpty(): Boolean = first > last

    override fun equals(other: Any?): Boolean =
        other is LongRange && (isEmpty() && other.isEmpty() ||
        first == other.first && last == other.last)

    override fun hashCode(): Int =
        if (isEmpty()) -1 else (31 * (first xor (first ushr 32)) + (last xor (last ushr 32))).toInt()

    override fun toString(): String = "$first..$last"

    companion object {
        /** An empty range of values of type Long. */
        public val EMPTY: LongRange = LongRange(1, 0)
    }
}

/**
 * Returns an empty read-only list. The returned list is
 * serializable (JVM).
 * @sample samples.collections.Collections.emptyList
 */
public fun <T> emptyList(): List<T> = EmptyList
serializable (JVM).

* @sample samples.collections.Collections.Lists.readOnlyList

/public fun <T> listOf(vararg elements: T): List<T> = if (elements.size > 0) elements.asList() else emptyList()

* @sample samples.collections.Collections.Lists.emptyList

public fun <T> listOf(vararg elements: T): List<T> = if (elements.size > 0) elements.asList() else emptyList()

/**
 * Returns an empty read-only list. The returned list is serializable (JVM).
 * @sample samples.collections.Collections.Lists.readOnlyList
 */
public fun <T> List(size: Int, init: (index: Int) -> T): List<T> = MutableList(size, init)

/**
 * Returns a new read-only list either of single given element, if it is not null, or empty list if the element is null. The returned list is serializable (JVM).
 * @sample samples.collections.Collections.Lists.emptyList
 */
public fun <T> MutableList(vararg elements: T): MutableList<T> = ArrayList(ArrayAsCollection(elements, isVarargs = true)).asList() * @sample samples.collections.Collections.Lists.emptyList

/**
 * Returns a new mutable list with the specified [size], where each element is calculated by calling the specified function.
 * The function [init] is called for each list element sequentially starting from the first one. It should return the value for a list element given its index. 
 * @sample samples.collections.Collections MutableList
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> MutableList(size: Int, init: (index: Int) -> T): MutableList<T> {
    val list = ArrayList<T>(size)
    repeat(size) { index -> list.add(init(index)) }
    return list
}

/**
 * Returns an [IntRange] of the valid indices for this collection.
 * @sample samples.collections.Collections.Collections.indicesOfCollection
 */
public val Collection<*>.indices: IntRange get() = 0..size - 1

/**
 * Returns the index of the last item in the list or -1 if the list is empty.
 * @sample samples.collections.Collections.Collections.lastIndex
 */
public val <T> List<T>.lastIndex: Int get() = this.size - 1

/**
 * Returns `true` if the collection is not empty or the empty list otherwise.
 * @sample samples.collections.Collections.Collections.isNotEmpty
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Collection<T>.isEmpty(): Boolean = isNullOrEmpty() * @sample samples.collections.Collections.Collections.isNotNull

/**
 * Returns this Collection if it's not `null` or the result of calling [defaultValue] function if the collection is
 * @sample samples.collections.Collections.Collections.orEmpty
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> Collection<T>?.orEmpty(): Collection<T> = this ?: emptyList()
empty.

@sample samples.collections.Collections.Collections.collectionIfEmpty

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Collection<*>, C : R =
    if (isEmpty()) defaultValue() else this

/**
 * Checks if all elements in the specified collection are contained in this collection.
 * @sample samples.collections.Collections.Collections.containsAll
 */
@kotlin.internal.InlineOnly
public inline fun <@kotlin.internal.OnlyInputTypes T>
    Collection<T>.containsAll(elements: Collection<T>): Boolean = this.containsAll(elements)

internal fun <T>
    List<T>.optimizeReadOnlyList() = when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else ->
            this
        }

/**
 * Searches this list or its range for the provided [element] using the binary search algorithm.
 * The list is expected to be sorted into ascending order according to the Comparable natural ordering of its elements,
 * otherwise the result is undefined.
 * If the list contains multiple elements equal to the specified [element], there is no guarantee which one will be found.
 * The null value is considered to be less than any non-null value.
 * @return the index of the element, if it is contained in the list within the specified range;
 * otherwise, the inverted insertion point `(insertion point - 1)`.
 * The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted.
 * @sample samples.collections.Collections.Collections.binarySearchOnComparable
 */
public fun <T : Comparable<T>>
    List<T?>.binarySearch(element: T?, fromIndex: Int = 0, toIndex: Int = size): Int {
        rangeCheck(size, fromIndex, toIndex)

        var low = fromIndex
        var high = toIndex - 1
        while (low <= high) {
            val mid = (low + high).ushr(1) // safe from overflows
            val midVal = get(mid)
            val cmp = midVal.compare(element)

            if (cmp < 0)
                low = mid + 1
            else if (cmp > 0)
                high = mid - 1

            else
                return mid // key found
        }
        return -(low + 1)  // key not found
    }

/**
 * Searches this list or its range for an element having the key returned by the specified [selector] function equal to the provided [key] value using the binary search algorithm.
 * The list is expected to be sorted into ascending order according to the Comparable natural ordering of keys of its elements,
 * otherwise the result is undefined.
 * If the list contains multiple elements with the specified [key], there is no guarantee which one will be found.
 * The null value is considered to be less than any non-null value.
 * @return the index of the element with the specified [key], if it is contained in the list within the specified range;
 * otherwise, the inverted insertion point `(insertion point - 1)`.
 * The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted.
 * @sample samples.collections.Collections.Collections.binarySearchByKey
 */
public inline fun <T, K : Comparable<K>>
    List<T>.binarySearchBy(
        key: K?,
        fromIndex: Int = 0,
        toIndex: Int = size,
        crossinline selector: (T) -> K?
    ): Int =
        binarySearch(fromIndex, toIndex) { selector(it) }

// do not introduce this
public fun <T, K> List<T>.binarySearchBy(key: K, comparator: Comparator<K>, fromIndex: Int = 0, toIndex: Int = size(), selector: (T) -> K): Int =
        binarySearch(fromIndex, toIndex) {
            comparator.compare(selector(it), key) }

/**
 * Searches this list or its range for an element for which the given [comparison] function returns zero using the binary search algorithm.
 * The list is expected to be sorted so that the signs of the [comparison] function's return values ascend on the list elements.
 * i.e. negative values come before zero and zeroes come before positive values.
 * Otherwise, the result is undefined.
 * If the list contains multiple elements for which [comparison] returns zero, there is no guarantee which one will be found.
 * @param comparison function that returns zero when called on the list element being searched.
 * On the elements coming before the target element, the function must return negative values.
 * On the elements coming after the target element, the function must return positive values.
 * @return the index of the found element, if it is contained in the list within the specified range; otherwise, the inverted insertion point `(-insertion point - 1)`.
 * The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted.
 * @sample samples.collections.Lists.binarySearchWithComparisonFunction
 */
public fun <T, K> List<T>.binarySearch(fromIndex: Int = 0, toIndex: Int = size, comparison: (T) -> Int): Int {
    rangeCheck(size, fromIndex, toIndex)

    var low = fromIndex
    var high = toIndex - 1
    while (low <= high) {
        val mid = (low + high).ushr(1) // safe from overflows
        val midVal = get(mid)
        val cmp = comparison(midVal)

        if (cmp < 0)
            low = mid + 1
        else if (cmp > 0)
            high = mid - 1
        else
            return mid // key found
    }
    return -(low + 1)  // key not found
}

private fun rangeCheck(size: Int, fromIndex: Int, toIndex: Int) {
    when {
        fromIndex > toIndex -> throw IllegalArgumentException("fromIndex ($fromIndex) is greater than toIndex ($toIndex).")
        fromIndex < 0 ->
            throw IndexOutOfBoundsException("fromIndex ($fromIndex) is less than zero.")
        toIndex > size -> throw IndexOutOfBoundsException("toIndex ($toIndex) is greater than size ($size).")
    }
}

@PublishedApi
@SinceKotlin("1.3")
internal expect fun checkIndexOverflow(index: Int): Int

@PublishedApi
@SinceKotlin("1.3")
internal expect fun checkCountOverflow(count: Int): Int

@PublishedApi
@SinceKotlin("1.3")
internal fun throwIndexOverflow() { throw ArithmeticException("Index overflow has happened.") }

@PublishedApi
@SinceKotlin("1.3")
internal fun throwCountOverflow() { throw ArithmeticException("Count overflow has happened.") }
emptyMap() * Returns an empty read-only map. * The returned map is serializable (JVM).
@sample samples.collections.Maps.Instantiation.emptyReadOnlyMap
*/
@kotlin.internal.InlineOnly
public inline fun <K, V> mapOf(): Map<K, V> = emptyMap() * Returns a new [MutableMap] with the specified contents, given as a list of pairs * where the first component is the key and the second is the value. * If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs. * Entries of the map are iterated in the order they were specified.
@sample samples.collections.Maps.Instantiation.mutableMapFromPairs
*/
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <K, V> hashMapOf(): HashMap<K, V> = HashMap<K, V>() * Returns an empty new [HashMap].
@sample samples.collections.Maps.Instantiation.hashMapFromPairs
*/
public fun <K, V> hashMapOf(vararg pairs: Pair<K, V>): HashMap<K, V> = LinkedHashSet<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) } * Returns a new [LinkedHashSet] with the specified contents, given as a list of pairs * where the first component is the key and the second is the value. * If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs. * Entries of the map are iterated in the order they were specified.
@sample samples.collections.Maps.Instantiation.linkedMapFromPairs
*/
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
private fun mapCapacity(expectedSize: Int): Int {
    if (expectedSize < 3) {
        return expectedSize + 1
    }
    if (expectedSize < INT_MAX_POWER_OF_TWO) {
        return expectedSize + expectedSize / 3
    }
    return Int.MAX_VALUE // any large value
}
private const val INT_MAX_POWER_OF_TWO: Int = Int.MAX_VALUE / 2 + 1

containsKey(key):\n\n/**
 * Returns the value corresponding to the given [key], or `null` if such a key is not present in the map.
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline operator fun <@kotlin.internal.OnlyInputTypes K, V> Map\langle out K, V\rangle\.get(key: K): V? = @n@ Suppress("UNCHECKED_CAST") (this as Map\langle K, V\rangle).get(key)\n
get(key):\n\n/**
 * Returns the value corresponding to the given key.
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline operator fun <@kotlin.internal.OnlyInputTypes K, V> Map\langle out K, V\rangle\.getOrDefault(key: K, defaultValue: () -> V): V = get(key) ?: defaultValue()\n
containsValue(value):\n\n/**
 * Returns `true` if the map maps one or more keys to the specified [value], or requires to pass a key of type `V`.
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes V> Map\langle out K, V\rangle\.containsValue(value: V): Boolean = this\.containsValue(value)\n
containsKey(key):\n\n/**
 * Returns `true` if the map contains the specified [key].
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline operator fun <@kotlin.internal.OnlyInputTypes K> Map\langle out K, V\rangle\.containsKey(key: K): Boolean = @n@ Suppress("UNCHECKED_CAST") (this as Map\langle K, V\rangle).containsKey(key)\n
containsEntry(key, value):\n\n/**
 * Returns `true` if the map contains an entry whose key is equal to the specified key and whose value is equal to the specified value.
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes K, V> Map\langle out K, V\rangle\.containsEntry(key: K, value: V): Boolean = this\.containsEntry(key, value)\n
remove(key):\n\n/**
 * Removes the specified key and its corresponding value from this map.
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes K, V> Map\langle out K, V\rangle\.remove(key: K): V? = this\.remove(key)\n
removeAll():\n\n/**
 * Removes all of the mappings from this map, if it is backed by an implementer of the `Set` interface, or causes a re-organization of mappings in some other manner if it is not.
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes K, V> Map\langle out K, V\rangle\.removeAll(): Unit = this\.removeAll()\n
removeAll(containsKey):\n\n/**
 * Removes key-value pairs for all keys found in the specified `Predicate`.
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes K> Map\langle out K, V\rangle\.removeAll(containsKey: Boolean): Unit = this\.removeAll { key, _ -> containsKey(key) }\n
entrySet():\n\n/**
 * Returns an `EntrySet` view of the map entries.
 */
@kotlin.internal.InlineOnly\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes K, V> Map\langle out K, V\rangle\.entrySet(): Set\langle Entry\rangle = this\.entrySet()\n
entries.iterator() * Populates the given [destination] map with entries having the keys of this map and the
values obtained by applying the [transform] function to each entry in this [Map].

/**
 * Populates the given [destination] map with entries having the keys obtained by applying the [transform] function to each entry in this [Map] and the values of this map.
 * In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.
 * @sample samples.collections.Maps.Transformations.mapValues
 */
    return entries.associateByTo(destination, { it.key }, transform) * Populates the given [destination] map with entries having the keys obtained by applying the [transform] function to each entry in this [Map] and the values of this map.
}

/**
 * Populates the given [destination] map with entries having the keys obtained by applying the [transform] function to each entry in this [Map] and the values of this map.
 * In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.
 * @sample samples.collections.Maps.Transformations.mapKeys
 */
    return entries.associateByTo(destination, transform, { it.value }) * Puts all the given [pairs] into this [MutableMap] with the first component in the pair being the key and the second the value.
}

/**
 * Puts all the elements of the given collection into this [MutableMap] with the first component in the pair being the key and the second the value.
 * In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.
 * @sample samples.collections.Maps.Filtering.filterTo
 */
    return destination.associateByTo(element, predicate) * Puts all the elements of the given collection into this [MutableMap] with the first component in the pair being the key and the second the value.
}

/**
 * Returns a new map containing all key-value pairs matching the given [predicate].
 * The returned map preserves the entry iteration order of the original map.
 * @sample samples.collections.Maps.Filtering.filter
 */
    return filterTo(LinkedHashMap<K, V>(), predicate) * Returns a new map containing all key-value pairs matching the given [predicate].
    
}

/**
 * Returns a new map containing all key-value pairs matching the given [predicate].
 * The returned map preserves the entry iteration order of the original map.
 * @sample samples.collections.Maps.Filtering.filter
 */
    return filterTo(LinkedHashMap<K, V>(), predicate) * Returns a new map containing all key-value pairs matching the given [predicate].
    
}

/**
 * Puts all the elements of the given collection into this [MutableMap] with the first component in the pair being the key and the second the value.
 * In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.
 * @sample samples.collections.Maps.Filtering.filterTo
 */
    return filterTo(destination, element, predicate) * Appends all entries matching the given [predicate] into the mutable map given as [destination].
}

/**
 * Puts all the elements of the given collection into this [MutableMap] with the first component in the pair being the key and the second the value.
 * In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.
 * @sample samples.collections.Maps.Filtering.filterTo
 */
    return filterTo(destination, element, predicate) * Appends all entries matching the given [predicate] into the mutable map given as [destination].
}
samples.collections.Maps.Filtering.filterNotTo

```kotlin
    return this.filterNotTo(LinkedHashMap<K, V>(), predicate)
}
```

```kotlin
public inline fun <K, V, M : MutableMap<in K, in V>>
    Map<out K, V>.filterNotTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}
```

```
/**
 * Returns a new map containing all key-value pairs not matching the given [predicate].
 * The returned map preserves the entry iteration order of the original map.
 */
@sample

samples.collections.Maps.Filtering.filterNotTo

```kotlin
    return filterNotTo(destination, predicate)
}
```

```
/**
 * Returns a new map containing all key-value pairs matching the given [predicate].
 * The returned map preserves the entry iteration order of the original map.
 */
```
pairs.toMap() else LinkedHashMap(this).apply { putAll(pairs) }  

/**
 * Creates a new read-only map by replacing or adding entries to this map from another [map].
 * Those entries of another [map] that are missing in this map are iterated in the end in the order of that [map].
 * @param map
 * @return A new read-only map.
 */
public operator fun <K, V> Map<out K, V>.plus(map: Map<out K, V>) =
    LinkedHashMap(this).apply { putAll(map) }

/**
 * Creates a new read-only map by replacing or adding entries to this map from another [map].
 * @param map
 * @return A new read-only map.
 */
public operator fun <K, V> Map<out K, V>.plusAssign(map: Map<out K, V>) =
    LinkedHashMap(this).apply { putAll(map) }

/**
 * Appends or replaces all pairs from the given collection of [pairs] in this mutable map.
 * @param pairs
 * @return A new mutable map.
 */
    MutableMap<in K, V> =
    this.toMutableMap().apply { putAll(pairs) }.optimizeReadOnlyMap()

/**
 * Appends or replaces all pairs from the given collection of [pairs] in this mutable map.
 * @param pairs
 * @return A new mutable map.
 */
    MutableMap<in K, V> =
    this.toMutableMap().apply { putAll(pairs) }.optimizeReadOnlyMap()

/**
 * Appends or replaces all entries from the given [map] in this mutable map.
 * @param map
 * @return A new mutable map.
 */
    MutableMap<in K, V> =
    this.toMutableMap().apply { putAll(map) }.optimizeReadOnlyMap()

/**
 * Removes the entry with the given [key] from this mutable map.
 * @param key
 * @return A new mutable map.
 */
    MutableMap<in K, V> =
    this.toMutableMap().apply { remove(key) }

/**
 * Removes all entries the keys of which are contained in the given [keys] collection from this mutable map.
 * @param keys
 * @return A new mutable map.
 */
    MutableMap<in K, V> =
    this.toMutableMap().apply { keys.removeAll(keys) }

/**
 * Removes all entries the keys of which are contained in the given [keys] sequence from this mutable map.
 * @param keys
 * @return A new mutable map.
 */
    MutableMap<in K, V> =
    this.toMutableMap().apply { keys.removeAll(keys) }

/**
 * Removes all entries from the keys of which are contained in the given [keys] array from this mutable map.
 * @param keys
 * @return A new mutable map.
 */
    MutableMap<in K, V> =
    this.toMutableMap().apply { keys.removeAll(keys) }

/**
 * Removes all entries from the keys of which are contained in the given [keys] sequence from this mutable map.
 * @param keys
 * @return A new mutable map.
 */
    MutableMap<in K, V> =
    this.toMutableMap().apply { keys.removeAll(keys) }
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SetsKt")

package kotlin.collections

internal object EmptySet : Set<Nothing>, Serializable {
    private const val serialVersionUID: Long = 340660374387020532

    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty() && isSingletonMapOrSelf() || this
    override fun hashCode(): Int = 0
    override fun toString(): String = "{[]}"

    override val size: Int get() = 0
    override fun isEmpty(): Boolean = true
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun iterator(): Iterator<Nothing> = EmptyIterator

    private fun readResolve(): Any = EmptySet
}

/**
 * Returns an empty read-only set.
 * The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.emptyReadOnlySet
 */
public fun <T> emptySet(): Set<T> = EmptySet

/**
 * Returns a new read-only set with the given elements.
 * Elements of the set are preserved in the order they were specified.
 * The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.emptySet
 */
public fun <T> setOf(vararg elements: T): Set<T> = if (elements.size > 0) LinkedHashSet(elements.toList()) else emptySet()

/**
 * Returns an empty read-only set. The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.emptyReadOnlySet
 */
@kotlin.internal.InlineOnly
public inline fun <T> setOf(): Set<T> = emptySet()

/**
 * Returns a new [MutableSet].
 * The returned set preserves the element iteration order.
 * @sample samples.collections.Collections.Sets.mutableSet
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> mutableSetOf(): MutableSet<T> = HashSet()

/** Returns a new [MutableSet] with the given elements.
 * @sample samples.collections.Collections.Sets.mutableSet
 */
public fun <T> mutableSetOf(vararg elements: T): MutableSet<T> = elements.toSet()

/** Returns an empty new [HashSet].
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> hashSetOf(): HashSet<T> = HashSet()

/** Returns a new [HashSet] with the given elements.
 */
public fun <T> hashSetOf(vararg elements: T): HashSet<T> = elements.toSet()

/** Returns this [Set] if it's not `null` and the empty set otherwise.
 * @sample Samples.Collections.Sets.setOf
 */
@kotlin.internal.InlineOnly
public inline fun <T, K> Set<T>.orEmpty(): Set<K> = optimizeReadOnlySet() when (size) {
    0 -> emptySet()
    else -> this
}

internal fun <T> Set<Nothing>.optimizeReadOnlySet() = when (size) {
    0 -> emptySet()
    1 -> toSingletonMapOrSelf() || this
    else -> this
}

package kotlin.collections

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

/**
 * Parses the string as a signed [Byte] number and returns the result or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> String.toByteOrNull(): Byte? = toByteOrNull()

/**
 * Parses the string as a signed [Byte] number and returns the result or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> String.toByteOrNull(radix: Int): Byte? = toByteOrNull()

/**
 * Parses the string as a [Short] number and returns the result or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> String.toShortOrNull(): Short? = toShortOrNull()

/**
 * Parses the string as a [Short] number and returns the result or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> String.toShortOrNull(radix: Int): Short? = toShortOrNull()

*/
string is not a valid representation of a number.

* `@throws IllegalArgumentException when \[radix\] is not a valid radix for string to number conversion.*

* `@SinceKotlin("1.1")` *\n
```kotlin
@SinceKotlin("1.1")
public fun String.toShortOrNull(radix: Int): Short? {
    val int = this.toIntOrNull(radix) ?: return null
    if (int < Short.MIN_VALUE || int > Short.MAX_VALUE) return null
    return int.toShort()
}
```

```kotlin
/**
 * Parses the string as an \[Int\] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.1")
public fun String.toIntOrNull(): Int? = toIntOrNull(radix = 10)

/**
 * Parses the string as an \[Int\] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 *
 * @throws IllegalArgumentException when \[radix\] is not a valid radix for string to number conversion.*

* `@SinceKotlin("1.1")` *\n
```kotlin
@SinceKotlin("1.1")
public fun String.toLongOrNull(): Long? = toLongOrNull(radix = 10)

/**
 * Parses the string as a \[Long\] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 *
 * @throws IllegalArgumentException when \[radix\] is not a valid radix for string to number conversion.*

* `@SinceKotlin("1.1")` *\n
```kotlin
@file:kotlin.jvm.JvmName("UnsignedKt")
@file:UseExperimental(ExperimentalUnsignedTypes::class)

@PublishedApi
internal fun uintCompare(v1: Int, v2: Int): Int = (v1 xor Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)

@PublishedApi

@PublishedApi
internal fun uintDivide(v1: UInt, v2: UInt): UInt = (v1.toLong() / v2.toLong()).toUInt()

@PublishedApi
internal fun uintRemainder(v1: UInt, v2: UInt): UInt = (v1.toLong() % v2.toLong()).toUInt()
```

---

*Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.*

@file:kotlin.jvm.JvmName("UnsignedKt")
@file:UseExperimental(ExperimentalUnsignedTypes::class)

package kotlin

@PublishedApi
internal fun uintCompare(v1: Int, v2: Int): Int = (v1 xor Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)

@PublishedApi

@PublishedApi
internal fun uintDivide(v1: UInt, v2: UInt): UInt = (v1.toLong() / v2.toLong()).toUInt()

@PublishedApi
internal fun uintRemainder(v1: UInt, v2: UInt): UInt = (v1.toLong() % v2.toLong()).toUInt()

---

*Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.*

*Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.*

---

*Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.*

---

*Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.*

---

*Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.*
val divisor = v2.toLong() if (divisor < 0) {
    // i.e., divisor >= 2^63
    return if (v1 < v2) ULong(0) else ULong(1)
}
// Optimization - use signed division if both dividend and divisor < 2^63
if (dividend >= 0) {
    return ULong(dividend / divisor)
}
// Otherwise, approximate the quotient, check, and correct if necessary.
val quotient = ((dividend ushr 1) / divisor) shl 1
val rem = dividend - quotient * divisor
return ULong(quotient + if (ULong(rem) >= ULong(divisor)) 1 else 0)

// Optimization - use signed modulus if both dividend and divisor < 2^63
if (dividend >= 0) {
    return ULong(dividend % divisor)
}
// Otherwise, approximate the quotient, check, and correct if necessary.
val quotient = ((dividend ushr 1) / divisor) shl 1
val rem = dividend - quotient * divisor
return ULong(rem - if (ULong(rem) >= ULong(divisor)) divisor else 0)

@PublishedApi
internal fun ulongRemainder(v1: ULong, v2: ULong): ULong {
    val dividend = v1.toLong()
    val divisor = v2.toLong()
    if (divisor < 0) { // i.e., divisor >= 2^63
        return if (v1 < v2) v1 // dividend < divisor
        else (v1 - v2) // dividend < divisor
    }

    // Optimization - use signed division if both dividend and divisor < 2^63
    if (dividend >= 0) {
        return ULong(dividend / divisor)
    }

    // Otherwise, approximate the quotient, check, and correct if necessary.
    val quotient = ((dividend ushr 1) / divisor) shl 1
    val rem = dividend - quotient * divisor
    return ULong(quotient + if (ULong(rem) >= ULong(divisor)) 1 else 0)
}

@PublishedApi
internal fun doubleToUInt(v: Double): UInt = when {
    v.isNaN() -> 0u
    v <= UInt.MIN_VALUE.toDouble() -> UInt.MIN_VALUE
    v >= UInt.MAX_VALUE.toDouble() -> UInt.MAX_VALUE
    v < Long.MAX_VALUE -> v.toInt().toUInt()
    else -> v - 9223372036854775808.0).toInt().toUInt() + Int.MAX_VALUE.toUInt()      // Int.MAX_VALUE < v <
}

@PublishedApi
internal fun doubleToULong(v: Double): ULong = when {
    v.isNaN() -> 0u
    v <= ULong.MIN_VALUE.toDouble() -> ULong.MIN_VALUE
    v >= ULong.MAX_VALUE.toDouble() -> ULong.MAX_VALUE
    v < Long.MAX_VALUE ->
        v.toDouble().toULong() // Real values from Long.MAX_VALUE to (Long.MAX_VALUE + 1) are not representable in Double, so don't handle them.
    else -> v - 9223372036854775808.0).toLong().toULong() + 9223372036854775808uL      // Long.MAX_VALUE + 1 < v <
}

@PublishedApi
internal fun uintToDouble(v: Int): Double = (v and Int.MAX_VALUE).toDouble() + (v ushr 31 shl 30).toDouble() * 2

@PublishedApi
internal fun ulongToDouble(v: Long): Double = (v ushr 11).toDouble() * 2048 + (v and 2047)

internal fun ulongToString(v: Long): String = ulongToString(v, 10)

internal fun ulongToString(v: Long, base: Int): String {
    if (v >= 0) return v.toString(base)

    var quotient = ((v ushr 1) / base) shl 1
    var rem = v - quotient * base
    if (rem >= base) {
        rem -= base
        quotient += 1
    }
    return quotient.toString(base) + rem.toString(base)
}

}
Converts this collection to a set, when it's worth so and it doesn't change contains method behavior. */

internal fun <T> Iterable<T>.convertToSetForSetOperation(): Collection<T> =
    when (this) {
        is Set -> this
        is Collection -> if (this.safeToConvertToSet()) toHashSet() else this
    }

    */

    internal fun <T> Iterable<T>.convertToSetForSetOperation(): Collection<T> =
    when (this) {
        is Set -> this
        is Collection -> if (this.safeToConvertToSet()) toHashSet() else this
        else -> toHashSet()
    }

    /**
     *
     * Returns all elements from all collections in the given collection.
     * @sample samples.collections.Iterables.Operations.flattenIterable
     */
    public fun <T> Iterable<Iterable<T>>.flatten(): List<T> {
        val result = ArrayList<T>()
        for (element in this) {
            result.addAll(element)
        }
        return result
    }

    /**
     * Returns a single list of all elements from all pairs in this collection.
     * @sample samples.collections.Iterables.Operations.unzipIterable
     */
    public fun <T, R> Iterable<Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {
        val listT = ArrayList<T>(expectedSize)
        val listR = ArrayList<R>(expectedSize)
        for (pair in this) {
            listT.add(pair.first)
            listR.add(pair.second)
        }
        return listT to listR
    }

    private object EmptySequence : Sequence<Nothing>,
    DropTakeSequence<Nothing> {
        override fun iterator(): Iterator<Nothing> = EmptyIterator
        override fun drop(n: Int) = EmptySequence
        override fun take(n: Int) = EmptySequence
    }

    /**
     * Returns this sequence if it's not `null` and the empty sequence otherwise.
     * @sample samples.collections.Sequences.Usage.sequenceOrEmpty
     */
    @SinceKotlin("1.3")
    @kotlin.internal.InlineOnly
    public inline fun <T> Sequence<T>?.orEmpty(): Sequence<T> = this ?: emptySequence()

    /**
     * Returns a sequence that iterates through the elements either of
     * this sequence or, if this sequence turns out to be empty, of the sequence returned by 
     * `defaultValue` function.
     * @sample samples.collections.Sequences.Usage.sequenceIfEmpty
     */
    @SinceKotlin("1.3")
    public fun <T> Sequence<T>.ifEmpty(defaultValue: () -> Sequence<T>): Sequence<T> = sequence {
        this@ifEmpty.iterator()
        if (iterator.hasNext()) {
            yieldAll(iterator)
        } else {
            yieldAll(defaultValue())
        }
    }

    /**
     * Returns a sequence of all elements from all sequences in this
     * sequence.
     */
    public fun <T> Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
        } else if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return FlatteningSequence(this, { it }, iterator)
        } else {
            return Collection.<R, Sequence<*>>.
            flatten(iterator) {
                sequence {
                    yieldAll(iterator)
                }
            }
        }
    }

    /**
     * Returns a sequence of all elements from all iterables in this
     * sequence.
     */
    @kotlin.jvm.JvmName("flattenSequenceOfIterable")
    public fun <T> Sequence<Iterable<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
        } else if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return FlatteningSequence(this, { it }, iterator)
        } else {
            return Collection.<R, Sequence<*>>.
            flatten(iterator) {
                sequence {
                    yieldAll(iterator)
                }
            }
        }
    }

    /**
     * Returns a sequence of all elements from all collections in this
     * sequence, where
     */
    public fun <T> Sequence<Collection<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
        } else if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return FlatteningSequence(this, { it }, iterator)
        } else {
            return Collection.<R, Sequence<*>>.
            flatten(iterator) {
                sequence {
                    yieldAll(iterator)
                }
            }
        }
    }

    /**
     * Returns a sequence of all elements from all iterables in this
     * sequence.
     */
    @kotlin.jvm.JvmName("flattenSequenceOfIterable")
    public fun <T> Sequence<Iterable<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
        } else if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return FlatteningSequence(this, { it }, iterator)
        } else {
            return Collection.<R, Sequence<*>>.
            flatten(iterator) {
                sequence {
                    yieldAll(iterator)
                }
            }
        }
    }

    /**
     * Returns a sequence of all elements from all sequences in this
     * sequence.
     */
    public fun <T> Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
        } else if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return FlatteningSequence(this, { it }, iterator)
        } else {
            return Collection.<R, Sequence<*>>.
            flatten(iterator) {
                sequence {
                    yieldAll(iterator)
                }
            }
        }
    }

    /**
     * Returns a sequence of all elements from all iterables in this
     * sequence.
     */
    @kotlin.jvm.JvmName("flattenSequenceOfIterable")
    public fun <T> Sequence<Iterable<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
        } else if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return FlatteningSequence(this, { it }, iterator)
        } else {
            return Collection.<R, Sequence<*>>.
            flatten(iterator) {
                sequence {
                    yieldAll(iterator)
                }
            }
        }
    }

    /**
     * Returns a sequence of all elements from all sequences in this
     * sequence.
     */
    public fun <T> Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
        } else if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return FlatteningSequence(this, { it }, iterator)
        } else {
            return Collection.<R, Sequence<*>>.
            flatten(iterator) {
                sequence {
                    yieldAll(iterator)
                }
            }
        }
    }

    /**
     * Returns a sequence of all elements from all iterables in this
     * sequence.
     */
    @kotlin.jvm.JvmName("flattenSequenceOfIterable")
    public fun <T> Sequence<Iterable<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
        } else if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return FlatteningSequence(this, { it }, iterator)
        } else {
            return Collection.<R, Sequence<*>>.
            flatten(iterator) {
                sequence {
                    yieldAll(iterator)
                }
            }
        }
    }

    /**
     * Returns a sequence of all elements from all sequences in this
     * sequence.
     */
    public fun <T> Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

    private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {
        if (this is TransformingSequence<*, *> || this is Sequence<*>.
        flatten(iterator)) {
            return (this as TransformingSequence<*,,
class FlatteningSequence<T, R, E> {
    constructor(
        private val sequence: Sequence<T>,
        private val transformer: (T) -> R,
        private val iterator: (R) -> Iterator<E>
    ) : Sequence<E> {
        override fun iterator(): Iterator<E> = object : Iterator<E> {
            override fun next(): E {
                val iterator = sequence.iterator()
                var itemIterator: Iterator<E>? = null
                override fun hasNext(): Boolean {
                    return ensureItemIterator()
                }
                private fun ensureItemIterator(): Boolean {
                    if (itemIterator == null) {
                        if (!iterator.hasNext()) {
                            return false
                        } else {
                            itemIterator = iterator.next()
                            return true
                        }
                    } else {
                        if (!itemIterator.hasNext()) {
                            itemIterator = null
                            return false
                        } else {
                            return true
                        }
                    }
                }
            }
        }
    }
}

/**
 * A sequence which combines values from the underlying [sequence] with their indices and returns them as
 * [IndexedValue] objects. 
 */
internal class IndexingSequence<T> {
    constructor(private val sequence: Sequence<T>) : Sequence<IndexedValue<T>> {
        override fun iterator(): Iterator<IndexedValue<T>> = object : Iterator<IndexedValue<T>> {
            val iterator = sequence.iterator()
            var index = 0
            override fun next(): IndexedValue<T> {
                return IndexedValue(index++, iterator.next())
            }
        }
    }
}

/**
 * A sequence which returns the values from the underlying [sequence] that either match or do not
 * match the specified [predicate]. 
 */
internal class FilteringSequence<T> {
    constructor(
        private val sequence: Sequence<T>,
        private val sendWhen: Boolean = true
    ) : Sequence<T> {
        override fun iterator(): Iterator<T> = object : Iterator<T> {
            private val predicate: (T) -> Boolean
        }
    }
}

/**
 * A sequence which returns the results of applying the given [transformer] function to the values in the underlying
 * [sequence]. 
 */
internal class TransformingSequence<T, R> {
    constructor(private val sequence: Sequence<T>, private val transformer: (T) -> R) : Sequence<R> {
        override fun iterator(): Iterator<R> = object : Iterator<R> {
            val iterator = sequence.iterator()
            var index = 0
            override fun next(): R {
                return transformer(iterator.next())
            }
        }
    }
}

/**
 * A sequence which combines values from the underlying [sequence] with their indices and returns them as
 * [IndexedValue] objects. 
 */
internal class TransformingIndexedSequence<T, R> {
    constructor(private val sequence: Sequence<T>, private val transformer: (Int, T) -> R) : Sequence<R> {
        override fun iterator(): Iterator<R> = object : Iterator<R> {
            val iterator = sequence.iterator()
            var index = 0
            override fun next(): R {
                return transformer(iterator.next())
            }
        }
    }
}

interface Container<T> {
    override fun hasNext(): Boolean
    override fun next(): T
}

interface Iterable<T> {
    override fun hasNext(): Boolean
    override fun next(): T
}

interface Iterator<T> {
    override fun hasNext(): Boolean
    override fun next(): T
}

interface Sequence<T> {
    override fun iterator(): Iterator<T>
}

interface Sequential<T> {
    override fun iterator(): Sequential<T>
}

interface Transformable<T> {
    override fun iterator(): Transformable<T>
}

class ArrayList<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class LinkedList<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Stack<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Queue<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class PriorityQueue<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Deque<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Set<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Map<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Array<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class ArrayList<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class LinkedList<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Stack<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Queue<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class PriorityQueue<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Deque<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Set<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Map<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Array<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class ArrayList<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class LinkedList<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Stack<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Queue<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class PriorityQueue<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Deque<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Set<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Map<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}

class Array<T> {
    override fun iterator(): Iterator<T> = { Iterator { this } }
}
val element = iterator.next()\nif (nextItemIterator.hasNext()) \n  nextItemIterator = iterator(transformer(element))\nelse \n  iterator = nextItemIterator\nif (nextItemIterator.hasNext()) \n  return true\nelse \n  return false

fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else SubSequence(sequence, startIndex + n, endIndex)\n
fun take(n: Int): Sequence<T> = if (n >= count) this else SubSequence(sequence, startIndex, startIndex + n)\n
fun override fun iterator() = object : Iterator<T> {
  val iterator = sequence.iterator()
  var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue
  var nextItem: T? = null\n
  override fun hasNext(): Boolean {
    return nextState == 1
  }

  override fun next(): T {
    if (nextState == -1)
      throw NoSuchElementException()\n
    if (predicate(item)) {
      nextState = 1
      nextItem = item
      return
    }

    if (nextItemIterator.hasNext()) {
      val nextItemIterator = iterator(transformer(element))\n      nextItemIterator.hasNext() \n
      if (predicate(item)) {
        nextState = 1
        nextItem = item\n        return
      }\n
      val nextState = 0\n      override fun next(): T {
        if (nextState == -1)
          return (position < startIndex && & & iterator.hasNext()) \n
        return position++ as T
      }

      override fun hasNext(): Boolean {
        return position < startIndex && iterator.hasNext()
      }

      override fun next(): T {
        return position++ as T
      }
    }

    val result = nextItem as T\n    nextItem = null

    if ((nextState == 0) \n
    when noSuchElementException() \n
    @Suppress("UNCHECKED_CAST")\n    val element = iterator.next()\n    val nextItemIterator = iterator(transformer(element))\n
    if (nextItemIterator.hasNext()) {
      nextItemIterator = iterator(transformer(element))\n    } else {
      iterator = nextItemIterator\n    }

    if (nextState == 0)
      throw NoSuchElementException()"}

    override fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else SubSequence(sequence, startIndex + n, endIndex)\n
    override fun take(n: Int): Sequence<T> = if (n >= count) this else SubSequence(sequence, startIndex, startIndex + n)\n
    override fun iterator() = object : Iterator<T> {
      val iterator = sequence.iterator()
      var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue
      var nextItem: T? = null

      override fun hasNext(): Boolean {
        return nextState == 1
      }

      override fun next(): T {
        return nextState++ as T
      }

      override fun hasNext(): Boolean {
        return nextState < startIndex && iterator.hasNext()
      }

      override fun next(): T {
        return nextState++ as T
      }
    }

    override fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else SubSequence(sequence, startIndex + n, endIndex)\n
    override fun take(n: Int): Sequence<T> = if (n >= count) this else SubSequence(sequence, startIndex, startIndex + n)
DropWhileSequence<T> constructor(T sequence, (T) -> Boolean) : Sequence<T> {
    override fun iterator(): Iterator<T> = object : Iterator<T> {
        val iterator = sequence.iterator()
        var dropState: Int = -1 // -1 for not dropping, 1 for nextItem, 0 for normal iteration
        var nextItem: T? = null

        private fun drop() {
            while (iterator.hasNext()) {
                val item = iterator.next()
                if (!predicate(item)) {
                    nextItem = item
                    dropState = 1
                    return
                }
            }
            dropState = 0
        }

        override fun next(): T {
            if (dropState == -1)
                drop()

            if (dropState == 1) {
                @Suppress("UNCHECKED_CAST")
                val result = nextItem as T
                nextItem = null
                dropState = 0
                return result
            }
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (dropState == -1)
                drop()
            return dropState == 1 || iterator.hasNext()
        }
    }
}

DistinctSequence<T, K>(source, keySelector) : Sequence<T> {
    override fun iterator(): Iterator<T> = DistinctIterator(source.iterator(), keySelector)
}

private class DistinctIterator<T, K>(source: Iterator<T>, keySelector: (T) -> K) : AbstractIterator<T>() {
    private val observed = HashSet<K>()

    override fun computeNext() {
        while (source.hasNext()) {
            val next = source.next()
            val key = keySelector(next)

            if (observed.add(key)) {
                setNext(next)
                return
            }
        }
        done()
    }
}

GeneratorSequence<T : Any>(getInitialValue: () -> T?, getNextValue: (T) -> T?) : Sequence<T> {
    override fun iterator(): Iterator<T> = object : Iterator<T> {
        var nextItem: T? = null
        var nextState: Int = -2 // -2 for initial unknown, -1 for next unknown, 0 for done, 1 for continue

        private fun calcNext() {
            nextItem = if (nextState == -2) getInitialValue() else getNextValue(nextItem!!)
            nextState = if (nextItem == null) 0 else 1
        }

        override fun next(): T {
            if (nextState < 0)
                calcNext()

            if (nextState == 0)
                throw NoSuchElementException()
            val result = nextItem as T
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState < 0)
                calcNext()
            return nextState == 1
        }
    }
}

constrainOnce(): Sequence<T> {
    return if (this is ConstrainedOnceSequence<T>) this else ConstrainedOnceSequence(this)
}

fun generateSequence(nextFunction: () -> T?): Sequence<T> {
    return GeneratorSequence(nextFunction, { nextFunction() }, constraintOnce: { sequence, constrainOnce: { sequence } })
}

fun generateSequence(seed: T?, nextFunction: (T) -> T?): Sequence<T> = if (seed == null)

EmptySequence else


GeneratorSequence({ seed }, nextFunction)\n\n* Returns a sequence defined by the function [seedFunction], which is invoked to produce the starting value, \n* and the [nextFunction], which is invoked to calculate the next value based on the previous one on each iteration. \n* The sequence produces values until it encounters first `null` value. \n* If [seedFunction] returns `null`, an empty sequence is produced. \n* The sequence can be iterated multiple times. \n* \n@see kotlin.sequences.sequence
\n@sample samples.collections.Sequences.Building.generateSequenceWithLazySeed

/**
 * Returns a sequence defined by the function \[seedFunction],
 * which is invoked to produce the starting value,
 * and the \[nextFunction], which is invoked to calculate the next
 * value based on the previous one on each iteration. \n * The sequence produces values until it encounters first
 * `null` value. \n * If \[seedFunction] returns `null`, an empty sequence is produced. \n * The sequence can be iterated multiple times. \n * \n@see kotlin.sequences.sequence
@sample samples.collections.Sequences.Building.generateSequenceWithLazySeed\n*/

public fun <T : Any> generateSequence(seedFunction: () -> T?, nextFunction: (T) -> T?): Sequence<T> = GeneratorSequence(seedFunction, nextFunction)\n
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("PreconditionsKt")

package kotlin

import kotlin.contracts.contract

/**
 * Throws an \[IllegalArgumentException\] if the \[value\] is false.\n * \n * @sample samples.miss.Preconditions.failRequireWithLazyMessage\n */
@kotlin.internal.InlineOnly
public inline fun require(value: Boolean): Unit {
    contract {
        returns() implies value
    }
    require(value) { "Failed requirement." }
}

/**
 * Throws an \[IllegalArgumentException\] if the \[value\] is null. Otherwise returns the not null value. \n * \n * @sample samples.miss.Preconditions.failRequireWithLazyMessage\n */
@kotlin.internal.InlineOnly
public inline fun requireNotNull(value: T?): T {
    contract {
        returns() implies value != null
    }
    return requireNotNull(value) { "Required value was null." }
}

/**
 * Throws an \[IllegalStateException\] if the \[value\] is false.\n * \n * @sample samples.miss.Preconditions.failCheckWithLazyMessage\n */
@kotlin.internal.InlineOnly
public inline fun check(value: Boolean): Unit {
    contract {
        returns() implies value
    }
    check(value) { "Check failed." }
}

/**
 * Throws an \[IllegalStateException\] if the \[value\] is null. Otherwise returns the not null value. \n * \n * @sample samples.miss.Preconditions.failCheckWithLazyMessage\n */
@kotlin.internal.InlineOnly
public inline fun checkNotNull(value: T?): T {
    contract {
        returns() implies value != null
    }
    return checkNotNull(value) { "Required value was null." }
}

/**
 * Throws an \[IllegalStateException\] with the given \[message\].\n * \n * @sample samples.miss.Preconditions.failWithError\n */
@kotlin.internal.InlineOnly
public inline fun error(message: Any): Nothing = throw IllegalStateException(message.toString())

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors. \n* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n* NOTE: THIS FILE IS AUTO-GENERATED
 */
public actual fun CharArray.asList(): List<Char> {
    return object : AbstractList<Char>(), RandomAccess {
        override val size: Int = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()

        override fun contains(element: Char): Boolean = this@asList.contains(element)
        override fun get(index: Int): Char {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: Char): Int {
            if ((element as Any?) !is Char) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: Char): Int {
            if ((element as Any?) !is Char) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}

/**
 * Returns a [List] that wraps the original array.
 */
@SinceKotlin("1.1")
@library("arrayDeepEquals")
public actual infix fun <T> Array<out T>.contentDeepEquals(other: Array<out T>): Boolean {
    definedExternally
}

/**
 * Returns a hash code based on the contents of this array as if it is [List].
 * Nested arrays are treated as lists too.
 * If any of arrays contains itself on any nesting level the behavior is undefined.
 * The elements of other types are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 */
@SinceKotlin("1.1")
@library("arrayDeepHashCode")
public actual fun <T> Array<out T>.contentDeepHashCode(): Int {
    definedExternally
}

/**
 * Returns a string representation of the contents of this array as if it is a [List].
 * Nested arrays are treated as lists too.
 * If any of arrays contains itself on any nesting level the behavior is undefined.
 */
@SinceKotlin("1.1")
@library("arrayDeepToString")
public actual fun <T> Array<out T>.contentDeepToString(): String {
    definedExternally
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 * The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 */
@SinceKotlin("1.1")
@library("arrayEquals")
public actual infix fun ByteArray.contentEquals(other: ByteArray): Boolean {
    definedExternally
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 * The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 */
@SinceKotlin("1.1")
@library("arrayEquals")
public actual infix fun ShortArray.contentEquals(other: ShortArray): Boolean {
   definedExternally
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 * The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 */
@SinceKotlin("1.1")
@library("arrayEquals")
public actual infix fun IntArray.contentEquals(other: IntArray): Boolean {
    definedExternally
}
*\@SinceKotlin("1.1")\n@library("arrayEquals")\npublic actual infix fun LongArray.contentEquals(\nother: \nLongArray): Boolean {\n    definedExternally\n}\n\n* Returns `true` if the two specified arrays are \n* structurally equal to one another,\n* i.e. contain the same number of the same elements in the same order.\n* The elements are compared for equality with the \[equals\] function.\n* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n*

*\@SinceKotlin("1.1")\n@library("arrayEquals")\npublic actual infix fun FloatArray.contentEquals(\nother: \nFloatArray): Boolean {\n    definedExternally\n}\n\n* Returns `true` if the two specified arrays are \n* structurally equal to one another,\n* i.e. contain the same number of the same elements in the same order.\n* The elements are compared for equality with the \[equals\] function.\n* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n*

*\@SinceKotlin("1.1")\n@library("arrayEquals")\npublic actual infix fun BooleanArray.contentEquals(\nother: \nBooleanArray): Boolean {\n    definedExternally\n}\n\n* Returns `true` if the two specified arrays are \n* structurally equal to one another,\n* i.e. contain the same number of the same elements in the same order.\n* The elements are compared for equality with the \[equals\] function.\n* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n*

*\@SinceKotlin("1.1")\n@library("arrayEquals")\npublic actual infix fun CharArray.contentEquals(\nother: \nCharArray): Boolean {\n    definedExternally\n}\n\n* Returns `true` if the two specified arrays are \n* structurally equal to one another,\n* i.e. contain the same number of the same elements in the same order.\n* The elements are compared for equality with the \[equals\] function.\n* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun Array<out T>.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun ByteArray.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun ShortArray.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun IntArray.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun LongArray.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun FloatArray.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun DoubleArray.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun BooleanArray.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayHashCode")\npublic actual fun CharArray.contentHashCode(): Int {\n    definedExternally\n}\n\n* Returns a hash code based on the contents of this array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun Array<out T>.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun ByteArray.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun ShortArray.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun IntArray.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun LongArray.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun FloatArray.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun DoubleArray.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun BooleanArray.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*

*\@SinceKotlin("1.1")\n@library("arrayToString")\npublic actual fun CharArray.contentToString(): String {\n    definedExternally\n}\n\n* Returns a string representation of the contents of the specified array as if it is \[List\].\n*
@SinceKotlin("1.1")
@library("arrayToString")
public actual fun ShortArray.contentToString(): String {
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
 * \[List\].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.1")
@library("arrayToString")
public actual fun IntArray.contentToString(): String {
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
 * \[List\].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.1")
@library("arrayToString")
public actual fun LongArray.contentToString(): String {
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
 * \[List\].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.1")
@library("arrayToString")
public actual fun FloatArray.contentToString(): String {
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
 * \[List\].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.1")
@library("arrayToString")
public actual fun DoubleArray.contentToString(): String {
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
 * \[List\].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.1")
@library("arrayToString")
public actual fun BooleanArray.contentToString(): String {
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
 * \[List\].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.1")
@library("arrayToString")
public actual fun CharArray.contentToString(): String {
    definedExternally
}

/**
 * Copies this array or its subrange into the \[destination\] array and returns that
 * array.
 * 
 * It's allowed to pass the same array in the \[destination\] and even specify the subrange so that it
 * overlaps with the destination range.
 * 
 * @param destination the array to copy to.
 * @param destinationOffset the position in the \[destination\] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or \[IllegalArgumentException\] when \[startIndex\]
 * or \[endIndex\] is out of range of this array indices or when `startIndex > endIndex`.
 * 
 * @return the \[destination\] array.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T> {
    arrayCopy(this, destination, destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the \[destination\] array and returns that array.
 * 
 * It's allowed to pass the same array in the \[destination\] and even specify the subrange so that it
 * overlaps with the destination range.
 * 
 * @param destination the array to copy to.
 * @param destinationOffset the position in the \[destination\] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or \[IllegalArgumentException\] when \[startIndex\]
 * or \[endIndex\] is out of range of this array indices or when `startIndex > endIndex`.
 * 
 * @return the \[destination\] array.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray {
    arrayCopy(this.unsafeCast<Array<Byte>>(), destination.unsafeCast<Array<Byte>>(), destinationOffset, startIndex, endIndex)
    return destination
}
@param destination the array to copy to.
 * @param destinationOffset the position in the [destination] array to
 copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [ endIndex] is out of range of this
 array indices or when `startIndex > endIndex`.
 * @return the [destination] array.

 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray {
    arrayCopy(this.unsafeCast<Array<Short>>(), destination.unsafeCast<Array<Short>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to.
 * @param destinationOffset the position in the [destination] array to
 copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [ endIndex] is out of range of this
 array indices or when `startIndex > endIndex`.
 * @return the [destination] array.

 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray {
    arrayCopy(this.unsafeCast<Array<Int>>(), destination.unsafeCast<Array<Int>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to.
 * @param destinationOffset the position in the [destination] array to
 copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [ endIndex] is out of range of this
 array indices or when `startIndex > endIndex`.
 * @return the [destination] array.

 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray {
    arrayCopy(this.unsafeCast<Array<Long>>(), destination.unsafeCast<Array<Long>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to.
 * @param destinationOffset the position in the [destination] array to
 copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [ endIndex] is out of range of this
 array indices or when `startIndex > endIndex`.
 * @return the [destination] array.

 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): FloatArray {
    arrayCopy(this.unsafeCast<Array<Float>>(),
    destination.unsafeCast<Array<Float>>(), destinationOffset, startIndex, endIndex)
    return destination
}  
* Copies this array or its subrange into the [destination] array and returns that array.
* In all other cases, [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @param destination the array to copy to.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset].
* @return the [destination] array.
*
*n @SinceKotlin("1.3")@kotlin.internal.InlineOnlyn @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\n\n/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * In all other cases, [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @param destination the array to copy to, 0 by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
*
*n @SinceKotlin("1.3")@kotlin.internal.InlineOnlyn @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\n\n/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * In all other cases, [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @param destination the array to copy to, 0 by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
*
*n @SinceKotlin("1.3")@kotlin.internal.InlineOnlyn @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\n\n/**
 * Returns new array which is a copy of the original array.
 * @param samples collections.Arrays.CopyOfOperations.copyOfOn  
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
*
*n @SinceKotlin("1.3")@kotlin.internal.InlineOnlyn @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\n\n/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * In all other cases, [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @param destination the array to copy to, 0 by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
*
*n @SinceKotlin("1.3")@kotlin.internal.InlineOnlyn @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\n\n/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * In all other cases, [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @param destination the array to copy to, 0 by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
*
*n @SinceKotlin("1.3")@kotlin.internal.InlineOnlyn @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\n\n/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * In all other cases, [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @param destination the array to copy to, 0 by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
*
array which is a copy of the original array.

```kotlin
public actual fun LongArray.copyOf(newSize: Int): LongArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("LongArray", arrayCopyResize(this, newSize, 0L))
}
```

```kotlin
public actual fun DoubleArray.copyOf(): DoubleArray {
    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n\n    return this.asDynamic().slice()\n```
in the copy array are filled with zero values.

```kotlin
public actual fun FloatArray.copyOf(newSize: Int): FloatArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, FloatArray(newSize))
}
```

```kotlin
public actual fun DoubleArray.copyOf(newSize: Int): DoubleArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, DoubleArray(newSize))
}
```

```kotlin
public actual fun BooleanArray.copyOf(newSize: Int): BooleanArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("BooleanArray", arrayCopyResize(this, newSize, false))
}
```

```kotlin
public actual fun CharArray.copyOf(newSize: Int): CharArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("CharArray", fillFrom(this, CharArray(newSize)))
}
```

```kotlin
public actual fun <T> Array<out T>.copyOf(newSize: Int): Array<T?> {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return arrayCopyResize(this, newSize, null)
}
```

```kotlin
public actual fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray {
    return this.asDynamic().slice(fromIndex, toIndex)
}
```

```kotlin
public actual fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray {
    return this.asDynamic().slice(fromIndex, toIndex)
}
```

```kotlin
public actual fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray {
    return this.asDynamic().slice(fromIndex, toIndex)
}
```

```kotlin
public actual fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray {
    return this.asDynamic().slice(fromIndex, toIndex)
}
```
IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public actual fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public actual fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public actual fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public actual fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * 
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IndexOutOfBoundsException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * 
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * 
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}
AbstractList.checkRangeIndexes(fromIndex, toIndex, size))  this.asDynamic().fill(element, fromIndex, toIndex));
}  \n}\n\n/* Fills this array or its subrange with the specified [element] value. */

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {

AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex);
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit {

AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex);
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {

AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex);
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {

AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex);
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {

AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex);
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit {

AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex);
}
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex);
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline operator fun <T> Array<out T>.plus(element: T): Array<T> {
    return this.asDynamic().concat(arrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ByteArray.plus(element: Byte): ByteArray {
    return plus(byteArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ShortArray.plus(element: Short): ShortArray {
    return plus(shortArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun IntArray.plus(element: Int): IntArray {
    return plus(intArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun LongArray.plus(element: Long): LongArray {
    return plus(longArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun FloatArray.plus(element: Float): FloatArray {
    return plus(floatArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun DoubleArray.plus(element: Double): DoubleArray {
    return plus(doubleArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun BooleanArray.plus(element: Boolean): BooleanArray {
    return plus(booleanArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun <T> Array<out T>.plus(elements: Collection<T>): Array<T> {
    return arrayPlusCollection(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun ShortArray.plus(elements: Collection<Short>): ShortArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun IntArray.plus(elements: Collection<Int>): IntArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun LongArray.plus(elements: Collection<Long>): LongArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun FloatArray.plus(elements: Collection<Float>): FloatArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray {
    return arrayPlusCollection(this, elements)
}
then all elements of the given [elements] collection.
*/
public actual operator fun CharArray.plus(elements: Collection<Char>): CharArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
} //

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] array.
 */
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual operator fun <T> Array<out T>.plus(elements: Array<out T>): Array<T> {
    return this.asDynamic().concat(elements)
} //

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ShortArray.plus(elements: ShortArray): ShortArray {
    return primitiveArrayConcat(this, elements)
} //

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun IntArray.plus(elements: IntArray): IntArray {
    return primitiveArrayConcat(this, elements)
} //

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun LongArray.plus(elements: LongArray): LongArray {
    return primitiveArrayConcat(this, elements)
} //

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {
    return primitiveArrayConcat(this, elements)
} //

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {
    return primitiveArrayConcat(this, elements)
} //

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {
    return primitiveArrayConcat(this, elements)
} //

/**
 * Returns an array containing all elements of the original array and then all elements of the
 * given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun CharArray.plus(elements: CharArray): CharArray {
    return primitiveArrayConcat(this, elements)
} //

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline fun <T> Array<out T>.plusElement(element: T): Array<T> {
    return this.asDynamic().concat(arrayOf(element))
} //

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun IntArray.sort(): Unit {
    definedExternally
} //

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
public actual fun LongArray.sort(): Unit {
    if (size > 1) sort {
        a: Long, b: Long -> a.compareTo(b)
    }
} //

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun ByteArray.sort(): Unit {
    definedExternally
} //

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun ShortArray.sort(): Unit {
    definedExternally
} //

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun DoubleArray.sort(): Unit {
    definedExternally
} //

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun FloatArray.sort(): Unit {
    definedExternally
} //

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun CharArray.sort(): Unit {
    definedExternally
} //

/**
 * Sorts the array in-place according to the natural order
of its elements. The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

```kotlin
/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
public actual fun <T : Comparable<T>> Array<out T>.sort(): Unit {
    if (size > 1) sortArray(this)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
public fun <T> Array<out T>.sort(comparison: (a: T, b: T) -> Int): Unit {
    if (size > 1) sortArrayWith(this, comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.sort(noinline comparison: (a: Byte, b: Byte) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.sort(noinline comparison: (a: Short, b: Short) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.sort(noinline comparison: (a: Int, b: Int) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.sort(noinline comparison: (a: Long, b: Long) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.sort(noinline comparison: (a: Float, b: Float) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.sort(noinline comparison: (a: Double, b: Double) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.sort(noinline comparison: (a: Char, b: Char) -> Int): Unit {
    asDynamic().sort(comparison)
}
```

```kotlin
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
@file:kotlin.jvm.JvmName("ComparisonsKt")
@file:kotlin.jvm.JvmMultifileClass

package kotlin.comparisons

/**
 * Compares two values using the specified functions [selectors] to calculate the result
 * of the comparison. The functions are called sequentially, receive the given values [a] and [b] and return
 * [Comparable] objects. As soon as the [Comparable] instances returned by a function for [a] and [b] values do
```
no\n* compare as equal, the result of that comparison is returned.\n* @sample
samples.comparisons.Comparisons.compareValuesByWithSelectors\n* /\n\npublic fun <T> compareValuesBy(a: T, b: T, vararg selectors: (T) -> Comparable<*?>?): Int {\n    require(selectors.size > 0)\n    return compareValuesByImpl(a, b, selectors)\n}\n\nprivate fun <T> compareValuesByImpl(a: T, b: T, selectors: Array<out (T) -> Comparable<*?>?>?): Int {\n    for (fn in selectors) {\n        val v1 = fn(a)\n        val v2 = fn(b)\n        val diff = compareValues(v1, v2)\n        if (diff != 0) return diff\n    }\n    return 0\n}\n\n/**\n * Compares two values using the specified \[selector\] function to calculate the result of the comparison.\n* The function is applied to the given values \[a\] and \[b\] and return \[Comparable\] objects.\n* The result of comparison of these \[Comparable\] instances is returned.\n* @sample samples.comparisons.Comparisons.compareValuesByWithSingleSelector\n* @\n* kotlin.internal.InlineOnly\n\npublic inline fun <T> compareValuesBy(a: T, b: T, selector: (T) -> Comparable<*?>?): Int {\n    return compareValues(selector(a), selector(b))\n}\n\n/**\n * Compares two values using the specified \[selector\] function to transform a \[Comparable\] value to another \[Comparable\] value for comparison.\n* The function is applied to the given values \[a\] and \[b\] and return \[Comparable\] objects. As soon as the \[Comparable\] instances returned by a function for \`a` and \`b` values do not\n* compare as equal, the result of that comparison is returned from the \[Comparable\]\n* @sample
samples.comparisons.Comparisons.compareValuesByWithSelectors\n* /\n\npublic fun <T, K> compareValuesBy(a: T, b: T, comparator: Comparator<in K>, selector: (T) -> K): Int {\n    return comparator.compare(selector(a), selector(b))\n}\n\n/**\n * Creates a comparator using the function to transform value to a \[Comparable\] instance for comparison.\n* @sample
samples.comparisons.Comparisons.compareValuesByWithSingleSelector\n* @\n* kotlin.internal.InlineOnly\n\npublic inline fun <T> compareBy(crossinline selector: (T) -> Comparable<*?>?): Comparator<T> =
    Comparator { a, b ->
        compareValuesBy(a, b, selector)\n    }\n\n/**\n * Creates a comparator using the \[selector\] function to transform values being compared and then applying\n* the specified \[comparator\] to compare transformed values.\n* @sample
samples.comparisons.Comparisons.compareValuesByWithComparator\n* @\n* kotlin.internal.InlineOnly\n\npublic inline fun <T, K> compareByDescending(crossinline selector: (T) -> Comparable<*?>?): Comparator<T> =
    Comparator { a, b ->
        compareValuesByDescending(a, b, selector)\n    }\n\n/**\n * Creates a descending comparator using the function to transform value to a \[Comparable\] instance for comparison.\n* @sample
samples.comparisons.Comparisons.compareValuesByDescendingWithSingleSelector\n* @\n* kotlin.internal.InlineOnly\n\npublic inline fun <T> compareValuesByDescending(crossinline selector: (T) -> Comparable<*?>?): Comparator<T> =
    Comparator { a, b ->
        compareValuesByDescending(a, b, selector)\n    }\n\n/**\n * Creates a descending comparator using the \[selector\] function to transform values being compared and then applying\n* the specified \[comparator\] to compare transformed values.\n* Note that an order of \[comparator\] is reversed by this wrapper.\n* @sample
samples.comparisons.Comparisons.compareValuesByDescendingWithComparator\n* @\n* kotlin.internal.InlineOnly\n\npublic inline fun <T, K> compareValuesByDescending(compparator: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =
    Comparator { a, b ->
        compareValuesByDescending(a, b, comparator, selector)\n    }\n\n/**
*Note: The text below is a natural representation of the provided code snippet.*

```kotlin
/**
 * Creates a comparator comparing values after the primary comparator defined them equal. It uses
 * the [selector] function to transform values and then compares them with the given [comparator].
 */
@kotlin.internal.InlineOnly
public inline fun <T>
    Comparator<T>.thenBy(crossinline selector: (T) -> Comparable<*?>?): Comparator<T> =
    Comparator { a, b -
        val previousCompare = this@thenBy.compare(a, b)
        if (previousCompare != 0) previousCompare else
            compareValuesBy(a, b, selector)
    }

/**
 * Creates a comparator comparing values after the primary comparator defined them equal. It uses
 * the [selector] function to transform values and then compares them with the given [comparator].
 */
@kotlin.internal.InlineOnly
public inline fun <T, K> Comparator<T>.thenBy(comparator: Comparator<in K>,
    crossinline selector: (T) -> K):
    Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenBy.compare(a, b)
        if (previousCompare != 0) previousCompare else compareValuesBy(a, b, comparator, selector)
    }

/**
 * Creates a descending comparator using the primary comparator and
 * the [selector] function to transform value to a [Comparable] instance for comparison.
 */
@kotlin.internal.InlineOnly
public inline fun <T>
    Comparator<T>.thenByDescending(crossinline selector: (T) -> Comparable<*?>?): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenByDescending.compare(a, b)
        if (previousCompare != 0) previousCompare else compareValuesBy(b, a, selector)
    }

/**
 * Creates a descending comparator comparing values after the primary comparator defined them equal. It uses
 * the [selector] function to transform values and then compares them with the given [comparator].
 */
@kotlin.internal.InlineOnly
public inline fun <T, K> Comparator<T>.thenByDescending(comparator: Comparator<in K>,
    crossinline selector: (T) -> K):
    Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenByDescending.compare(a, b)
        if (previousCompare != 0) previousCompare else compareValuesBy(b, a, comparator, selector)
    }

/**
 * Creates a comparator using the primary comparator and function to calculate a result of comparison.
 */
@kotlin.internal.InlineOnly
public inline
    fun <T> Comparator<T>.thenComparator(crossinline comparison: (a: T, b: T) -> Int): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenComparator.compare(a, b)
        if (previousCompare != 0) previousCompare else comparison(a, b)
    }

/**
 * Combines this comparator and the given [comparator] such that the latter is applied only
 * when the former considered values equal.
 */
public infix fun <T> Comparator<T>.then(comparator:
    Comparator<in T>): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@then.compare(a, b)
        if (previousCompare != 0) previousCompare else comparator.compare(a, b)
    }

/**
 * Combines this comparator and the given [comparator] such that the latter is applied only
 * when the former considered values equal.
 */
public infix fun <T> Comparator<T>.thenDescending(comparator: Comparator<in T>): Comparator<T> =
    this@thenDescending.compare(a, b)

/**
 * Extends the given [comparator] of non-nullable values to a comparator
 * considering `null` value less than any other value.
 */
@kotlin.internal.InlineOnly
public fun <T : Any>
    nullsFirst(comparator: Comparator<in T>): Comparator<T?> =
    Comparator { a, b ->
        when {
            a === b -> 0
            a == null -> -1
            b == null -> 1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Provides a comparator of nullable [Comparable] values* considering ‘null’ value less than any other value.
 */
@kotlin.internal.InlineOnly
public infix fun <T : Any>
    nullsFirstLastWithComparator
    nullsFirstLastWithComparator
    Comparator<T> =
    Comparator { a, b ->
        when {
            a == b -> 0
            a === b -> -1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Extends the given [comparator] of non-nullable values to a comparator
 * of nullable values* considering `null` value greater than any other value.
 */
@kotlin.internal.InlineOnly
public fun <T : Any>
    nullsFirstLastComparator
    nullsFirstLastComparator
    Comparator<T> =
    Comparator { a, b ->
        when {
            a == b -> 0
            a === b -> -1
            else -> comparator.compare(a, b)
        }
    }
```

---

**Open Source Used In DNAC Platform Antelope 3873**
Antelope  3874

```kotlin
fun nullsLast(comparator: Comparator<T>): Comparator<T?> =
    Comparator { a, b ->
        when {
            a == b -> 0
            a == null -> 1
            b == null -> -1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Provides a comparator of nullable [Comparable] values
 * considering `null` value greater than any other value.
 * @sample samples.comparisons.Comparisons.nullsLast
 */
@kotlin.internal.InlineOnly
public inline fun <T : Comparable<T>> nullsLast(): Comparator<T?> =
    nullsLast(naturalOrder())

/**
 * Returns a comparator that compares [Comparable] objects in natural order.
 * @sample samples.comparisons.Comparisons.naturalOrderComparator
 */
public fun <T : Comparable<T>> naturalOrder(): Comparator<T> =
    @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as Comparator<T>)

/**
 * Returns a comparator that compares [Comparable] objects in reversed natural order.
 * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator
 */
public fun <T : Comparable<T>> reverseOrder(): Comparator<T> =
    @Suppress("UNCHECKED_CAST") (ReverseOrderComparator as Comparator<T>)

/**
 * Returns a comparator that imposes the reverse ordering of this comparator.
 * @sample samples.comparisons.Comparisons.reversed
 */
public fun <T> Comparator<T>.reversed(): Comparator<T> =
    when (this) {
        is ReversedComparator -> this.comparator
        NaturalOrderComparator -> @Suppress("UNCHECKED_CAST") (ReverseOrderComparator as Comparator<T>)
        ReverseOrderComparator -> @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as Comparator<T>)
        else -> ReversedComparator(this)
    }

private class ReversedComparator<T>(public val comparator: Comparator<T>) : Comparator<T> {
    override fun compare(a: T, b: T): Int = comparator.compare(b, a)
    override fun reversed(): Comparator<T> = comparator
}

private object NaturalOrderComparator : Comparator<Comparable<Any>> {
    override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)
    override fun reversed(): Comparator<Comparable<Any>> = ReverseOrderComparator
}

private object ReverseOrderComparator : Comparator<Comparable<Any>> {
    override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)
    override fun reversed(): Comparator<Comparable<Any>> = NaturalOrderComparator
}
```

```
/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in
 * the license/LICENSE.txt file.
 */

package kotlin

import kotlin.contracts.*

/**
 * An exception is thrown to indicate that a method body remains to be implemented.
 */
public class NotImplementedError(message: String = "An operation is not implemented.") :
    Error(message)

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 * @param reason a string explaining why the implementation is missing.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(reason: String) : Nothing = throw NotImplementedError("An operation is not implemented: $reason")

/**
 * Calls the specified function [block] and returns its result.
 * @param block a function returning the result.
 */
@kotlin.internal.InlineOnly
public inline fun <R> run(block: () -> R): R =
    run(block, InvocationKind.EXACTLY_ONCE)

/**
 * Calls the specified function [block] with `this` value as its receiver and returns its result.
 * @param block a function.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R> T.run(block: T.() -> R): R =
    run(block, InvocationKind.ONE)
```

---

**Open Source Used In DNAC Platform Antelope**

3874
callsInPlace(block, InvocationKind.EXACTLY_ONCE) | \n return block() | \n
*/

*Calls the specified function [block] with the given [receiver] as its receiver and returns its result. *

*For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#callsinplace).

 */

*In this

* For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#apply).

 */

//
// NOTE: THIS FILE IS AUTO-GENERATED by the
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.js.*

/**
 * Returns the greater of two values.
 * If values are equal, returns the first one.
 */
@SinceKotlin("1.1.1")
public actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {
    return if (a >= b) a
    else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Byte, b: Byte): Byte {
    return Math.max(a.toInt(), b.toInt()).unsafeCast<Byte>()
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Short, b: Short): Short {
    return Math.max(a.toInt(), b.toInt()).unsafeCast<Short>()
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Int, b: Int): Int {
    return Math.max(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Long, b: Long): Long {
    return if (a >= b) a
    else b
}

```
inline fun maxOf(a: Float, b: Float): Float {
    return Math.max(a, b)
} /* Returns the greater of two values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Double, b: Double): Double {
    return Math.max(a, b)
} /* Returns the greater of three values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Byte, b: Byte, c: Byte): Byte {
    return Math.max(a.toInt(), b.toInt(),
    c.toInt()).unsafeCast<Byte>()
} /* Returns the greater of three values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Short, b: Short, c: Short): Short {
    return Math.max(a.toInt(), b.toInt(),
    c.toInt()).unsafeCast<Short>()
} /* Returns the greater of three values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Int, b: Int, c: Int): Int {
    return Math.max(a, b, c)
} /* Returns the greater of three values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Long, b: Long, c: Long): Long {
    return Math.max(a, maxOf(b, c))
} /* Returns the greater of three values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Float, b: Float, c: Float): Float {
    return Math.max(a, b, c)
} /* Returns the greater of three values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Double, b: Double, c: Double): Double {
    return Math.max(a, b, c)
} /* Returns the greater of three values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual fun <T : Comparable<T>> minOf(a: T, b: T): T {
    return if (a <= b) a else b
} /* Returns the smaller of two values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun minOf(a: Byte, b: Byte): Byte {
    return Math.min(a.toInt(), b.toInt()).unsafeCast<Byte>()
} /* Returns the smaller of two values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun minOf(a: Short, b: Short): Short {
    return Math.min(a.toInt(), b.toInt()).unsafeCast<Short>()
} /* Returns the smaller of two values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun minOf(a: Int, b: Int): Int {
    return Math.min(a, b)
} /* Returns the smaller of two values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR", "NOTHING_TO_INLINE")
public actual inline fun minOf(a: Long, b: Long): Long {
    return if (a <= b) a else b
} /* Returns the smaller of two values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun minOf(a: Float, b: Float): Float {
    return Math.min(a, b)
} /* Returns the smaller of two values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun minOf(a: Double, b: Double): Double {
    return Math.min(a, b)
} /* Returns the smaller of two values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun minOf(a: Byte, b: Byte, c: Byte): Byte {
    return Math.min(a.toInt(), b.toInt(),
    c.toInt()).unsafeCast<Byte>()
} /* Returns the smaller of three values.*/

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun minOf(a: Short, b: Short, c: Short): Short {
    return Math.min(a.toInt(), b.toInt(),
    c.toInt()).unsafeCast<Short>()
} /* Returns the smaller of three values.*/
public actual inline fun minOf(a: Int, b: Int, c: Int): Int {
    return Math.min(a, b, c)
}

/**
 * Returns the smaller of three values.
 */
public actual inline fun minOf(a: Long, b: Long, c: Long): Long {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the smaller of three values.
 */
public actual inline fun minOf(a: Float, b: Float, c: Float): Float {
    return Math.min(a, b, c)
}

/**
 * Returns the smaller of three values.
 */
public actual inline fun minOf(a: Double, b: Double, c: Double): Double {
    return Math.min(a, b, c)
}

private @SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
@kotlin.RangeOrientation
private actual inline fun minOf(a: Int, b: Int, c: Int): Int {
    return Math.min(a, b, c)
}

/**
 * Returns 1st *element* from the collection.
 */
public inline operator fun <T> List<T>.component1(): T {
    return get(0)
}

/**
 * Returns 2nd *element* from the collection.
 */
public inline operator fun <T> List<T>.component2(): T {
    return get(1)
}

/**
 * Returns 3rd *element* from the collection.
 */
public inline operator fun <T> List<T>.component3(): T {
    return get(2)
}

/**
 * Returns 4th *element* from the collection.
 */
public inline operator fun <T> List<T>.component4(): T {
    return get(3)
}

/**
 * Returns 5th *element* from the collection.
 */
public inline operator fun <T> List<T>.component5(): T {
    return get(4)
}

/**
 * Returns `true` if *element* is found in the collection.
 */
public operator fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.contains(element: T): Boolean {
    if (this is Collection)
        return contains(element)
    return indexOf(element) >= 0
}

/**
 * Returns an element at the given *index* or throws an *IndexOutOfBoundsException* if the *index* is out of bounds of this list.
 * 
 * @sample samples.collections.Collections.Elements.elementAt
 */
public fun <T> List<T>.elementAtOrNull(index: Int): T? {
    if (this is List)
        return getOrNull(index)
    if (index < 0)
        return null
    val iterator = iterator()
    var count = 0
    while (iterator.hasNext()) {
        val element = iterator.next()
        if (index == count++)
            return element
    }
    return null
}

/**
 * Returns an element at the given *index* or throws an *IndexOutOfBoundsException* if the *index* is out of bounds of this list.
 * 
 * @sample samples.collections.Collections.Elements.elementAtOrNull
 */
public fun <T> List<T>.elementAtOrElse(index: Int): T {
    if (this is List)
        return getOrElse(index)
    return defaultValueOrNull(index)
}

/**
 * Returns an element at the given *index* or throws an *IndexOutOfBoundsException* if the *index* is out of bounds of this list.
 * 
 * @sample samples.collections.Collections.Elements.elementAt
 */
@kotlin.RangeOrientation
public inline operator fun <T> List<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {
    if (this is List)
        return getOrElse(index)
    if (index < 0)
        return defaultValue(index)
    val iterator = iterator()
    var count = 0
    while (iterator.hasNext()) {
        val element = iterator.next()
        if (index == count++)
            return element
    }
    return defaultValue(index)
}
samples.collections.Collections.Elements.elementAtOrNull

@kotlin.internal.InlineOnly
public inline fun <T> List<T>.elementAtOrNull(index: Int): T? {
    return this.getOrNull(index)
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if no such element was found.
 */
public inline fun <T> Iterable<T>.find(predicate: (T) -> Boolean): T? {
    return this.findLast(predicate)
}

/**
 * Returns the last element matching the given [predicate], or ‘null’ if no such element was found.
 */
public inline fun <T> List<T>.findLast(predicate: (T) -> Boolean): T? {
    return this.getOrNull(this.lastIndex)
}

/**
 * Returns the first element.
 * @throws [NoSuchElementException] if the collection is empty.
 */
public fun <T> List<T>.first(): T {
    if (this.isEmpty())
        throw NoSuchElementException("List is empty.")
    return this[0]
}

/**
 * Returns the first element, or `null` if the list is empty.
 */
public fun <T> List<T>.firstOrNull(): T? {
    return if (this.isEmpty()) null else this[0]
}

/**
 * Returns first index of [element], or -1 if the collection does not contain such element.
 */
public inline fun <T> Iterable<T>.indexOfFirst(predicate: (T) -> Boolean): Int {
    var index = 0
    for (item in this) {
        if (predicate(item))
            return index
        index++
    }
    return -1
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the collection does not contain such element.
 */
public inline fun <T> List<T>.indexOfFirst(predicate: (T) -> Boolean): Int {
    var index = 0
    for (item in this) {
        if (predicate(item))
            return index
        index++
    }
    return -1
}

/**
 * Returns the last index of [element], or -1 if the list does not contain such element.
 */
public fun <T> List<T>.lastIndex(): Int {
    return this.size - 1
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the collection does not contain such element.
 */
public inline fun <T> Iterable<T>.indexOfLast(predicate: (T) -> Boolean): Int {
    var lastIndex = -1
    var index = 0
    for (item in this) {
        if (predicate(item))
            lastIndex = index
        index++
    }
    return lastIndex
}
returns the last element matching the given [predicate].

@Throws(NoSuchElementException, IndexOutOfBoundsException)

/**
 * Returns last index of [element], or -1 if the collection does not contain element.
 * \n */
public fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.lastIndexOf(element: T): Int {
    return lastIndexOf(element)
}

/**
 * Returns last index of [element], or -1 if the list does not contain element.
 * \n */
@Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning, extension takes precedence in some cases
public fun <@kotlin.internal.OnlyInputTypes T> List<T>.lastIndexOf(element: T): Int {
    return lastIndexOf(element)
}

/**
 * Returns the last element, or null if the collection does not contain element.
 * \n */
public fun <T> Iterable<T>.lastOrNull(): T? {
    return if (isEmpty()) null else lastElement
}

/**
 * Returns the last element, or null if the list is empty.
 * \n */
public fun <T> List<T>.lastOrNull(): T? {
    return if (isEmpty()) null else lastElement
}

/**
 * Returns the last element matching the given [predicate], or null if no such element was found.
 * \n */
public inline fun <T> Iterable<T>.lastOrNull(predicate: (T) -> Boolean): T? {
    return lastOrNull()
}

/**
 * Returns the last element matching the given [predicate], or null if no such element was found.
 * \n */
public inline fun <T> List<T>.lastOrNull(predicate: (T) -> Boolean): T? {
    return lastOrNull()
}

/**
 * Returns a random element from this collection.
 * \n */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Collection<T>.random(): T {
    return random()
}

/**
 * Returns a random element from this collection using the specified source of randomness.
 * \n */
@SinceKotlin("1.3")
public fun <T> Collection<T>.random(random: Random): T {
    return random.nextElement()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> List<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Collection<T>.single(): T {
    return single()
}

/**
 * Returns single element, or throws an exception if the collection is empty or has more than one element.
 * \n */
public fun <T> Iterable<T>.single(): T }

Open Source Used In DNAC Platform Antelope 3879
element and the element itself. * and returns the result of predicate evaluation on the element. *

```kotlin
@kotlin.internal.NoInfer R: List<T> -> Boolean): C {
    for (index in indices)
        if (predicate(index, element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are specified type parameter R to the given [destination]. *

```kotlin
fun <T, C : MutableCollection<in T>> Iterable<T>.filterIsInstanceTo(destination: C): C {
    for (element in this)
        if (element is R)
            destination.add(element)
    return destination
}
```

Returns a list containing elements that are instances of specified type parameter R to the given [destination]. *

```kotlin
fun <T, C :MutableCollection<in T>> Iterable<T>.filterIndexedTo(destination: C, predicate: (Int, T) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element))
            destination.add(element)
    } return destination
}
```

Returns a list containing all elements that are instances of specified type parameter R to the given [destination]. *

```kotlin
fun <T, C : MutableCollection<in T>> Iterable<T>.filterNot(predicate: (T) -> Boolean): List<T> {
    val list = ArrayList<T>()
    for (item in this)
        if (!predicate(item))
            list.add(item)
    return list
}
```

Returns a list containing all elements that are not `null` to the given [destination]. *

```kotlin
fun <T> List<T>.take(n: Int): List<T> {
    if (n >= size)
        return toList()
    if (n == 1)
        return listOf(last())
    val list = ArrayList<T>(n)
    if (this is RandomAccess)
        for (index in size - n until size)
            list.add(this[index])
    else
        for (item in this)
            if (item != null)
                list.add(item)
    return list
}
```

Returns a list containing last elements.

```kotlin
fun <T> List<T>.takeLast(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing `null` elements.

```kotlin
fun <T> Iterable<T>.filterIsInstance(): List<R> {
    return filterIsInstanceTo(ArrayList<R>())
}
```

Returns a list containing all elements that are instances of specified type parameter R to the given [destination]. *

```kotlin
fun <T> Iterable<T>.filterNot(): List<T> {
    return filterNotTo(ArrayList<T>(), predicate)
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {
    for (it in this)
        if (predicate(it))
            add(it)
    return toList()
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> List<T>.takeFirstWhile(predicate: (T) -> Boolean): List<T> {
    for (it in this)
        if (predicate(it))
            add(it)
    return toList()
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> List<T>.takeFirstWhile(predicate: (T) -> Boolean): List<T> {
    for (it in this)
        if (predicate(it))
            add(it)
    return toList()
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.

```kotlin
fun <T> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this)
        if (!predicate(element))
            destination.add(element)
    return destination
}
```

Returns a list containing all elements that are not matching the given [predicate]. *

```kotlin
fun <T> List<T>.takeFirst(n: Int): List<T> {
    return emptyList()
}
```

Returns a list containing first elements.
expect fun <T> MutableList<T>.reverse(): Unit

/**
 * Returns a list with elements in reversed order.
 */
public fun <T> mutableListOf<T>.reverse(): Unit

/**
 * Sorts elements in the list in-place according to natural sort order of the value returned by specified [selector] function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T : Comparable<T>> MutableList<T>.sortDescending(): Unit

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T : Comparable<T>> MutableList<T>.sortBy(): Unit

/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T : Comparable<T>> MutableList<T>.sortByDescending(): Unit

/**
 * Returns a list all elements sorted according to their natural sort order.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T : Comparable<T>> MutableList<T>.sorted(): List<T>

/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T : Comparable<T>> MutableList<T>.sortedWith(Comparator<in T>): List<T>

/**
 * Returns an array of Boolean containing all of the elements of this collection.
 */
public fun Collection<Boolean>.toBooleanArray(): BooleanArray

/**
 * Returns an array of Byte containing all of the elements of this collection.
 */
public fun Collection<Byte>.toByteArray(): ByteArray

/**
 * Returns an array of Char containing all of the elements of this collection.
 */
public fun Collection<Char>.toCharArray(): CharArray

/**
 * Returns an array of Double containing all of the elements of this collection.
 */
public fun Collection<Double>.toDoubleArray(): DoubleArray

/**
 * Returns an array of Float containing all of the elements of this collection.
 */
public fun Collection<Float>.toFloatArray(): FloatArray

/**
 * Returns an array of Int containing all of the elements of this collection.
 */
public fun Collection<Int>.toIntArray(): IntArray

/**
 * Returns an array of Any contains all of the elements of this collection.
 */
public fun Collection<Any>.toArray(): Array<Any?>

/**
 * Returns an array of Any contains all of the elements of this collection.
 */
public fun Collection<*>.toArray(): Array<*>
result[index++] = element
    return result
}

/**
 * Returns an array of Long containing all of the elements of this collection.
 */
@<public fun Collection<Long>.toLongArray(): LongArray {
    val result = LongArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given collection.
 * If any of two pairs would have the same key the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 */
@<public inline fun <T, K, V> Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

/**
 * Returns an array of Short containing all of the elements of this collection.
 */
@<public fun Collection<Short>.toShortArray(): ShortArray {
    val result = ShortArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns a [Map] containing the elements from the given collection indexed by the key returned from [keySelector] function applied to each element.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 */
@<public inline fun <T, K:MutableMap<in K, in V>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

/**
 * Returns a [Map] where keys are elements from the given collection and values are produced by the [valueSelector] function applied to each element.
 */
@<SinceKotlin("1.3") fun <K, V> Iterable<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {
    val result = LinkedHashMap<K, V>(mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

// Sample usage
fun main() {
    val elements = listOf(1L, 2L, 3L, 4L)
    val longArray = elements.toLongArray()
    val shortArray = elements.toShortArray()
    val map = elements.associate { it * 2 }
    val hashMap = elements.associateWith { it * 2 }
    println(longArray)
    println(shortArray)
    println(map)
    println(hashMap)
}

// Output
[1, 2, 3, 4]
[1, 2, 3, 4]
{(1, 2), (2, 4), (3, 6), (4, 8)}
{(1, 2), (2, 4), (3, 6), (4, 8)}
Iterable<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {
    for (element in this) {
        return destination
    }
    return destination
}

Iterable<T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

Iterable<T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>(mapCapacity(collectionSizeOrDefault(12))))
}

Iterable<T>.toList(): List<T> {
    if (this is Collection) {
        return when (size) {
            0 -> emptyList()
            1 -> listOf(if (this is List) get(0) else iterator().next())
            else -> this.toMutableList()
        }
    }
    return this.toMutableList().optimizeReadOnlyList()
}

Iterable<T>.flatMap(transform: (T) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Iterable<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)
}

Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<T>()
        }
        list.add(element)
    }
    return destination
}

Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<T>()
        }
        list.add(element)
    }
    return destination
}

Iterable<T>.flatMapTo(destination: M, keySelector: (T) -> K, valueSelector: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<T>()
        }
        list.add(element)
    }
    return destination
}

/**
 * Appends all elements to
 * the given [destination] collection.
 * @return [destination]
 */
public fun <T, M : MutableCollection<T>> Iterable<T>.toMutableList(): M {
    return toCollection(ArrayList<T>())
}

/**
 * Returns a [List] containing all elements.
 * @return a [List] containing all elements.
 */
public fun <K> Iterable<K>.toList(): List<K> {
    if (this is Collection) {
        return when (size) {
            0 -> emptyList()
            1 -> listOf(if (this is List) get(0) else iterator().next())
            else -> this.toMutableList()
        }
    }
    return this.toMutableList().optimizeReadOnlyList()
}

/**
 * Returns a [Set] of all elements.
 * @return a [Set] of all elements.
 */
public fun <K> Iterable<K>.toSet(): Set<K> {
    if (this is Collection) {
        return when (size) {
            0 -> emptySet()
            1 -> setOf(if (this is List) this[0] else iterator().next())
            else -> toCollection(LinkedHashSet<K>(mapCapacity(size)))
        }
    }
    return toCollection(LinkedHashSet<K>()).optimizeReadOnlySet()
}

/**
 * Groups elements of the original collection by the key returned by the given [keySelector] function
 * and returns a map where each key is associated with a list of corresponding elements.
 * @return a map where each key is associated with a list of corresponding elements.
 */
public fun <K, M : MutableMap<K, MutableList<T>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<T>()
        }
        list.add(element)
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original collection
 * by the key returned by the given [keySelector] function and puts to the [destination] map
 * each group key associated with a list of corresponding values.
 * @return a map where each key is associated with a list of corresponding values.
 */
public fun <K, M : MutableMap<K, MutableList<T>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<T>()
        }
        list.add(element)
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original collection
 * by the key returned by the given [keySelector] function and puts to the [destination] map
 * each group key associated with a list of corresponding values.
 * @return a map where each key is associated with a list of corresponding values.
 */
public fun <K, M : MutableMap<K, MutableList<T>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<T>()
        }
        list.add(element)
    }
    return destination
}
destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return
}

/**
 * Creates a [Grouping] source from a collection to be used later with one of group-and-fold
 * operations
 * using the specified [keySelector] function to extract a key from each element.
 * @param sample samples.collections.Grouping.groupingByEachCount
 * @return
 */
public inline fun <T, K> Iterable<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {
    var override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
    var override fun keyOf(element: T): K = keySelector(element)

    return
}

/**
 * Returns a list containing the results of applying the given [transform] function to each
 * element in the original collection.
 * @param sample samples.collections.Collections.Transformations.map
 */
public inline fun <T, R> Iterable<T>.map(transform: (T) -> R): List<R> {
    return mapTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

/**
 * Applies the given [transform] function to each element and its index in
 * the original collection.
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 */
public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {
    forEachIndexed { index, element -> transform(checkIndexOverflow(index), element)?.let { destination.add(it) } }
    return destination
}

/**
 * Applies the given [transform] function to each element and its index in
 * the original collection. It appends the results to an indexable collection
 * with an indexable transformation
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 */
public inline fun <T, R> Iterable<T>.mapIndexedTo(destination: MutableList<R>, transform: (index: Int, T) -> R): MutableList<R> {
    var index = 0
    for (item in this)
        destination.add(transform(checkIndexOverflow(index++), item))
    return destination
}

/**
 * Returns a list containing only non-null results of applying the given [transform] function
 * to each element in the original collection.
 */
public inline fun <T, R : Any> Iterable<T>.mapNotNull(transform: (T) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(), transform)
}

/**
 * Applies the given [transform] function to each element of the original collection
 * and appends the results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 */
public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.mapTo(destination: C, transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original collection
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun <T> Iterable<T>.withIndex(): Iterable<IndexedValue<T>> {
    return return IndexingIterable { iterator() }
}

/**
 * Returns a list containing only distinct elements from a collection.
 * The elements in the resulting list are in the same order as they were in the source collection.
 * @param [selector] function that takes a single parameter and returns true if the parameter
 * is distinct.
 */
public fun <T> Iterable<T>.distinct(): List<T> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only elements from a collection
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source collection.
 */
public fun <T, K> Iterable<T>.distinctBy(selector: (T) -> K): List<T> {
    return this.toMutableSet().toList()
}
-> K): List<T> {
    val set = HashSet<K>()
    val list = ArrayList<T>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

/**
 * Returns a set containing all elements that are contained by this collection and not contained by the specified collection.
 */
public infix fun <T> Iterable<T>.subtract(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

/**
 * Returns a mutable set containing all distinct elements from the given collection.
 */
public fun <T> List<T>.toMutableSet(): MutableList<T> {
    return when (this) {
        is Collection<T> ->
        else -> toCollection(LinkedHashSet<T>())
    }
}

/**
 * Returns a set containing all elements that are contained by both collections.
 */
public infix fun <T> Iterable<T>.union(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns `true` if all elements match the given 
 * predicate.
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun <T> Iterable<T>.all(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return true
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if this collection has at least one element.
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun <T> Iterable<T>.any(): Boolean {
    if (this is Collection) return !isEmpty()
    return iterator().hasNext()
}

/**
 * Returns `true` if at least one element matches the given 
 * predicate.
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun <T> Iterable<T>.any(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return false
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns the number of elements in this collection.
 */
public fun <T> Iterable<T>.count(): Int {
    if (this is Collection) return size
    var count = 0
    for (element in this) checkCountOverflow(++count)
    return count
}

/**
 * Returns the number of elements matching the given 
 * predicate.
 */
public fun <T> List<T>.count(predicate: (T) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) checkCountOverflow(++count)
    return count
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right.
 * @param operation
 */
public inline fun <T, R> List<T>.foldRight(initial: R, operation: (T, R) -> R): R {
    var accumulator = initial
    if (!isEmpty()) {
        val iterator = listIterator(size)
        while (iterator.hasPrevious()) {
            accumulator = operation(iterator.previous(), accumulator)
        }
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left.
 * @param operation
 */
public inline fun <T, R> List<T>.foldLeft(initial: R, operation: (R, T) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right.
 * @param operation
 */
public fun <T, R> List<T>.foldIndexed(initial: R, operation: (Int, R, T) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(checkIndexOverflow(index++, accumulator, element))
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left.
 * @param operation
 */
public fun <T, R> List<T>.foldInitial(initial: R, operation: (R, T) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right.
 * @param operation
 */
public fun <T, R> List<T>.fold(initial: R, operation: (R, T) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
Open Source Used In DNAC Platform Antelope

```
@SinceKotlin("1.1")
public fun Iterable<Float>.min(): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    if (min.isNaN()) return min
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * 
 * If any of elements is `NaN` returns `NaN`.
 */
@SinceKotlin("1.1")
public fun Iterable<Double>.min(): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    if (min.isNaN()) return min
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (e.isNaN()) return e
        if (min < e) min = e
    }
    return min
}

/**
 * Returns the first element having the largest value according to the provided 
 * [comparator] or `null` if there are no elements.
 */
public fun <T> Iterable<T>.maxWith(comparator: Comparator<in T>): T? {
    var max = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there
 * are no elements.
 * 
 * If any of elements is `NaN` returns `NaN`.
 */
@SinceKotlin("1.1")
public fun Iterable<Float>.max(): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var max = iterator.next()
    if (max.isNaN()) return max
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (max > e) max = e
    }
    return max
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there
 * are no elements.
 * 
 * If any of elements is `NaN` returns `NaN`.
 */
@SinceKotlin("1.1")
public fun Iterable<Double>.max(): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var max = iterator.next()
    if (max.isNaN()) return max
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the smallest element or `null` if there are no elements.
 */
@SinceKotlin("1.1")
public fun Iterable<Float>.minBy(selector: (Float) -> Float): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minValue = selector(minElement)
    do {
        val e = iterator.next()
        val v = selector(e)
        if (minValue > v) minValue = v
    } while (iterator.hasNext())
    return minValue
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.1")
public fun Iterable<Double>.minBy(selector: (Double) -> Double): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var maxValue = selector(maxElement)
    do {
        val e = iterator.next()
        val v = selector(e)
        if (maxValue < v) maxValue = v
    } while (iterator.hasNext())
    return maxValue
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * 
 * If any of elements is `NaN` returns `NaN`.
 */
@SinceKotlin("1.1")
public fun Iterable<Double>.minBy(selector: (Double) -> Double): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minValue = selector(minElement)
    do {
        val e = iterator.next()
        val v = selector(e)
        if (minValue > v) minValue = v
    } while (iterator.hasNext())
    return minValue
}
```

function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.\n
```
v) { minElem = e
    minValue = v
} while (iterator.hasNext())

returns the first element having the smallest value according to the provided [comparator] or ‘null’ if there are no elements.

public fun <T> Iterable<T>.minWith(comparator: Comparator<T>): T? {
  val iterator = iterator()
  var minValue = iterator.next()
  while (iterator.hasNext()) {
    if (comparator.compare(minValue, iterator.next()) < 0) {
      minValue = iterator.next()
    }
  }
  return minValue
}

 Returns `true` if the collection has no elements.

inline fun <T> Iterable<T>.minWith(comparator: Comparator<T>): T? {
  var iterator = iterator()
  if (!iterator.hasNext()) return null
  var min = iterator.next()
  while (iterator.hasNext()) {
    var e = iterator.next()
    if (comparator.compare(min, e) > 0) min = e
  }
  return min
}

/**
 * Returns the first element having the smallest value according to the provided 
 * [comparator] or `null` if there are no elements.
 */
public fun <T> Iterable<T>.minWith(comparator: Comparator<T>): T? {
  var iterator = iterator()
  if (!iterator.hasNext()) return null
  var min = iterator.next()
  while (iterator.hasNext()) {
    var e = iterator.next()
    if (comparator.compare(min, e) > 0) min = e
  }
  return min
}

/**
 * Returns `true` if the collection has no elements.
 */
public fun <T> Iterable<T>.none(): Boolean {
  if (this is Collection) return isEmpty()
  return !iterator().hasNext()
}

/**
 * Returns `true` if no elements match the given 
 * [predicate].
 */
public inline fun <T> Iterable<T>.none(predicate: (T) -> Boolean): Boolean {
  if (this is Collection && isEmpty())
    return true
  for (element in this) if (predicate(element)) return false
  return true
}

/**
 * Performs the given 
 * [action] on each element and returns the collection itself afterwards.
 */
@SinceKotlin("1.1")
public inline fun <T, C : Iterable<T>> C.onEach(action: (T) -> Unit): C {
  apply { for (element in this) action(element) }
}

/**
 * Accumulates value starting with the first element and applying [operation] from left
 * to right to current accumulator value and each element in the original collection.
 */
public inline fun <S, T : S> Iterable<T>.reduce(operation: (acc: S, T) -> S): S {
  val iterator = this.iterator()
  if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")
  var accumulator: S = iterator.next()
  while (iterator.hasNext()) {
    accumulator = operation(accumulator, iterator.next())
  }
  return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element in the original collection.
 */
public inline fun <S, T : S> List<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {
  val iterator = listIterator(size)
  if (!iterator.hasPrevious()) throw UnsupportedOperationException("Empty collection can't be reduced.")
  var index = 1
  var accumulator: S = iterator.previous()
  while (iterator.hasPrevious()) {
    val index = iterator.previousIndex()
    accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.previous())
  }
  return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 */
public inline fun <S, T : S> List<T>.reduceRight(operation: (T, acc: S) -> S): S {
  val iterator = listIterator(size)
  if (!iterator.hasPrevious()) throw UnsupportedOperationException("Empty collection can't be reduced.")
  var accumulator: S = iterator.previous()
  while (iterator.hasPrevious()) {
    accumulator = operation(iterator.previous(), accumulator)
  }
  return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and its index in the original list and current accumulator value.
 */
  val iterator = listIterator(size)
  if (!iterator.hasPrevious()) throw UnsupportedOperationException("Empty collection can't be reduced.")
  var accumulator: S = iterator.previous()
  while (iterator.hasPrevious()) {
    val index = iterator.previousIndex()
    accumulator = operation(index, iterator.previous(), accumulator)
  }
  return accumulator
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the collection.
 */
public inline fun <T> Iterable<T>.sumBy(selector: (T) -> Int): Int {
  var sum: Int = 0
  for (element in this) {
    sum += selector(element)
  }
  return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the collection.
 */
public inline fun <T> Iterable<T>.sumByDouble(selector: (T) -> Double): Double {
  var sum: Double = 0.0
  for (element in this) {
    sum += selector(element)
  }
  return sum
}

/**
 * Returns an original collection containing all the non-`null` elements, throwing an 
 * [IllegalArgumentException] if there are any `null` elements.
 */
public fun <T : Any> Iterable<T?>.requireNoNulls(): Iterable<T> {
  for (element in this) {
    if (element == null) throw IllegalArgumentException("null element found in Sths.")
  }
  return this as Iterable<T>
}

@Suppress("UNCHECKED_CAST")

/**
 * Returns a list containing all elements of the original collection and then all elements of the given [elements] array.
 */
public operator fun <T> Collection<T>.plus(element: T): List<T> {
    val result = ArrayList<T>(size + 1)
    result.addAll(this)
    result.add(element)
    return result
}

/**
 * Returns a list containing all elements of the original collection and then all elements of the given [elements] array.
 */
public operator fun <T> Iterable<T>.plus(elements: Array<T>): List<T> {
    val result = ArrayList<T>(size + elements.size)
    result.addAll(this)
    result.addAll(elements)
    return result
}

/**
 * Returns a list containing all elements of the original collection and then all elements of the given [elements] collection.
 */
public operator fun <T> Collection<T>.plus(elements: Iterable<T>): List<T> {
    val result = ArrayList<T>(size + elements.size)
    result.addAll(this)
    result.addAll(elements)
    return result
}

/**
 * Returns a list containing all elements of the original collection and then all elements of the given [elements] collection.
 */
public operator fun <T> Iterable<T>.plus(elements: Iterable<T>): List<T> {
    if (this is RandomAccess && this is List)
        return ArrayList<T>(this.size + elements.size)
    val result = ArrayList<T>()
    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = false).forEach {
        result.add(it)
    }
    return result
}

/**
 * Returns a list of results of applying the given [transform] function to an each list representing a view over the window of the given [size] sliding along this collection with the given [step].
 */
    if (this is RandomAccess && this is List)
        return ArrayList<T>(this.size + 10)
    val checkWindowSizeStep(size, step, size) if (this is RandomAccess && this is List) {
        val thisSize = thisSize
        return result
    }
    val result = ArrayList<List<T>>()
    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = false).forEach {
        result.add(it)
    }
    return result
}

*/

public inline fun <T> Collection<T>.plusElement(element: T): List<T> {
    return plus(element)
}

public inline fun <T> Iterable<T>.plusElement(element: T): List<T> {
    return plus(element)
}

/**
 * Returns a list of snapshots of the window of the given [size] sliding along this collection with the given [step], where each [snapshot] is a list.
 * Several last lists may have less elements than the given [size].
 */
    if (this is RandomAccess && this is List)
        return ArrayList<T>(this.size + 10)
    val checkWindowSizeStep(size, step, size) if (this is RandomAccess && this is List) {
        val thisSize = thisSize
        return result
    }
    val result = ArrayList<List<T>>()
    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = false).forEach {
        result.add(it)
    }
    return result
}

*/
any.
" by default 'false' which means partial windows won't be preserved
@sample samples.collections.Sequences.Transformations.averageWindows
@sample public fun <T, R> Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): List<R> { if (this is RandomAccess && this is List) { val thisSize = this.size
  val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1
  val result = ArrayList<R>(resultCapacity)
  var window = MovingSubList(this)
  var index = 0
  while (index in 0 until thisSize) {
    val windowSize = size.coerceAtMost(thisSize - index)
    if (!partialWindows && windowSize < size) break
    window.move(index, index + windowSize)
    result.add(transform(window))
    index += step
  }
  return result
}

/**
 * Returns a list of pairs built from the elements of this collection and the other array with the same index.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public fun <T, R> Iterable<T>.zip(other: Array<out T>): List<Pair<T, T>> { var i = 0
  while (i < other.size) {
    result.add(transform(element, other[i++]))
  }
  return result
}

/**
 * Returns a list of pairs built from the elements of this collection and the other collection with the same index.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> { val first = other.iterator()
  val second = other.iterator()
  val list = ArrayList<Pair<T, R>>(minOf(collectionSizeOrDefault(10), other.collectionSizeOrDefault(10)))
  while (first.hasNext() && second.hasNext()) {
    list.add(transform(first.next(), second.next()))
  }
  return list
}

/**
 * Returns a list of pairs of each two adjacent elements in this collection.
 * The returned list is empty if this collection contains less than two elements.
 * @sample samples.collections.Transformations.zipWithNext
 */
public fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> { val iterator = iterator()
  if (!iterator.hasNext()) return emptyList()
  val result = mutableListOf<Pair<T, T>>()
  var current = iterator.next()
  while (iterator.hasNext()) {
    val next = iterator.next()
    result.add(transform(current, next))
    current = next
  }
  return result
}

/**
 * Appends the string from all the elements separated using separator and using the given prefix and postfix if supplied.
 * If the collection could be huge, you can specify a non-negative value of limit, in which case only the first limit elements will be appended, followed by the truncated string (which defaults to "...").
 * @sample samples.Collections.Transformations.zipWithNextToFindDeltas
 */
public fun <T, A : Appendable> Iterable<T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "\n", transform: ((T) -> CharSequence)? = null): A {
  if (!iterator.hasNext()) return buffer
  val first = iterator.next()
  if (first is RandomAccess && first is List) {
    val list = ArrayList(first)
    return list.zipWithNext { transform(first) }{ transform(second) }
  }
  val result = mutableListOf<String>()
  while (iterator.hasNext()) {
    val current = iterator.next()
    result.add(transform(current))
    result = result.joinToString(separator, prefix, postfix)
  }
  if (result.size > limit) {
    val truncated = result.take(limit - 1) + "...
    return buffer.append(truncated)
  }
  return buffer.append(result)
}

/**
 * Returns a list containing the results of applying the given function to an each pair of two adjacent elements in this collection.
 * The returned list is empty if this collection contains less than two elements.
 * @sample samples.collections.Transformations.zipWithNextToFindDeltas
 */
public fun <T, R> Iterable<T>.zipWithNext(transform: (a: T, b: R) -> R): List<R> { val iterator = other.iterator()
  val result = mutableListOf<R>()
  while (iterator.hasNext()) {
    val current = iterator.next()
    result.add(transform(first, current))
    first = current
  }
  return result
}

/**
 * Returns a list of pairs built from the elements of this collection and the other array with the same index.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public fun <T, R> Iterable<T>.zip(other: Array<out T>): List<Pair<T, R>> { var i = 0
  while (i < other.size) {
    result.add(transform(element, other[i++]))
  }
  return result
}

/**
 * Returns a list of pairs built from the elements of this collection and the other collection with the same index.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Transformations.zipWithNext
 */
public fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> { val first = other.iterator()
  val second = other.iterator()
  val list = ArrayList<Pair<T, R>>(minOf(collectionSizeOrDefault(10), other.collectionSizeOrDefault(10)))
  while (first.hasNext() && second.hasNext()) {
    list.add(transform(first.next(), second.next()))
  }
  return list
}

/**
 * Returns a list of pairs of each two adjacent elements in this collection.
 * The returned list is empty if this collection contains less than two elements.
 * @sample samples.collections.Transformations.zipWithNext
 */
public fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> { val iterator = iterator()
  if (!iterator.hasNext()) return emptyList()
  val result = mutableListOf<Pair<T, T>>()
  var current = iterator.next()
  while (iterator.hasNext()) {
    val next = iterator.next()
    result.add(transform(current, next))
    current = next
  }
  return result
}

/**
 * Appends the string from all the elements separated using separator and using the given prefix and postfix if supplied.
 * If the collection could be huge, you can specify a non-negative value of limit, in which case only the first limit elements will be appended, followed by the truncated string (which defaults to "...").
 * @sample samples.Collections.Transformations.zipWithNextToFindDeltas
 */
public fun <T, A : Appendable> Iterable<T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "\n", transform: ((T) -> CharSequence)? = null): A {
  if (!iterator.hasNext()) return buffer
  val first = iterator.next()
  if (first is RandomAccess && first is List) {
    val list = ArrayList(first)
    return list.zipWithNext { transform(first) }{ transform(second) }
  }
  val result = mutableListOf<String>()
  while (iterator.hasNext()) {
    val current = iterator.next()
    result.add(transform(current))
    result = result.joinToString(separator, prefix, postfix)
  }
  if (result.size > limit) {
    val truncated = result.take(limit - 1) + "...
    return buffer.append(truncated)
  }
  return buffer.append(result)
}

/**
 * Returns a list of pairs built from the elements of this collection and the other array with the same index.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public fun <T, R> Iterable<T>.zip(other: Array<out T>): List<Pair<T, R>> { var i = 0
  while (i < other.size) {
    result.add(transform(element, other[i++]))
  }
  return result
}

/**
 * Returns a list of pairs built from the elements of this collection and the other collection with the same index.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Transformations.zipWithNext
 */
public fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> { val first = other.iterator()
  val second = other.iterator()
  val list = ArrayList<Pair<T, R>>(minOf(collectionSizeOrDefault(10), other.collectionSizeOrDefault(10)))
  while (first.hasNext() && second.hasNext()) {
    list.add(transform(first.next(), second.next()))
  }
  return list
}

/**
 * Returns a list of pairs of each two adjacent elements in this collection.
 * The returned list is empty if this collection contains less than two elements.
 * @sample samples.collections.Transformations.zipWithNext
 */
public fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> { val iterator = iterator()
  if (!iterator.hasNext()) return emptyList()
  val result = mutableListOf<Pair<T, T>>()
  var current = iterator.next()
  while (iterator.hasNext()) {
    val next = iterator.next()
    result.add(transform(current, next))
    current = next
  }
  return result
}

/**
 * Appends the string from all the elements separated using separator and using the given prefix and postfix if supplied.
 * If the collection could be huge, you can specify a non-negative value of limit, in which case only the first limit elements will be appended, followed by the truncated string (which defaults to "...").
 * @sample samples.Collections.Transformations.zipWithNextToFindDeltas
 */
public fun <T, A : Appendable> Iterable<T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "\n", transform: ((T) -> CharSequence)? = null): A {
  if (!iterator.hasNext()) return buffer
  val first = iterator.next()
  if (first is RandomAccess && first is List) {
    val list = ArrayList(first)
    return list.zipWithNext { transform(first) }{ transform(second) }
  }
  val result = mutableListOf<String>()
  while (iterator.hasNext()) {
    val current = iterator.next()
    result.add(transform(current))
    result = result.joinToString(separator, prefix, postfix)
  }
  if (result.size > limit) {
    val truncated = result.take(limit - 1) + "...
    return buffer.append(truncated)
  }
  return buffer.append(result)
}
var count = 0
for (element in this) {
    if (++count > 1) buffer.append(separator)
    var count = 0
    for (element in this) {
        if (++count > 1) buffer.append(separator)
        if (limit < 0 || count <= limit) {
            buffer.appendElement(element, transform)
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * Returns a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 */
public fun <T> Iterable<T>.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Returns this collection as an [Iterable].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.asIterable(): Iterable<T> {
    return this
}

/**
 * Creates a [Sequence] instance that wraps the original collection returning its elements when being iterated.
 * Returns a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 */
@kotlin.jvm.JvmName("averageOfByte")
public fun Iterable<Byte>.average(): Double {
    var sum = 0.0
    var count = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Iterable<Byte>.sum(): Int {
    var sum = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Iterable<Int>.sum(): Int {
    var sum = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Iterable<Long>.sum(): Long {
    var sum = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Iterable<Float>.sum(): Float {
    var sum = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Iterable<Double>.sum(): Double {
    var sum = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}
for (element in this) {
    sum += element
}
return sum
}

for (element in this) {
    sum += element
}
return sum
}

for (element in this) {
    sum += element
}
return sum
}

for (element in this) {
    sum += element
}
return sum
}

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the
 * Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

import kotlin.comparisons.naturalOrder
import kotlin.random.Random

/** Returns the array if it's not `null`, or an empty array otherwise. */
@kotlin.internal.InlineOnly
public actual inline fun <T> Array<out T>?.orEmpty(): Array<out T> = copyToArrayList(this)

/**
 * Randomly shuffles elements in this list.
 *
 * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.shuffle(): Unit = shuffle(Random)

/**
 * Sorts elements in the list in-place according to their natural sort order.
 *
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public actual fun <T : Comparable<T>> MutableList<T>.sort(): Unit {
    sortArrayWith(array, naturalOrder())
}

/**
 * Sorts elements in the list in-place according to the order
 * specified with [comparator].
 *
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public actual fun <T> MutableList<T>.sortWith(comparator: Comparator<in T>): Unit {
    sortArrayWith(array, comparator)
}

private fun <T> sortArrayWith(array: Array<T>, comparator: Comparator<in T>) {
    if (array.size <= 1) return

    val rangeSize = array.size
    for (i in 0 until array.size) {
        array[i] = array[rangeSize - 1 - i]
    }
}

internal actual fun <T> arrayOfNulls(reference: Array<T>, size: Int): Array<T> {
    return arrayOfNulls<Any>(size).unsafeCast<Array<T>>()
}

@SinceKotlin("1.3")
@PublishedApi
@JsName("arrayCopy")
internal fun <T> arrayCopy(source: Array<out T>, destination: Array<in T>, destinationOffset: Int, startIndex: Int, endIndex: Int) {
    AbstractList.checkRangeIndexes(startIndex, endIndex, source.size) { 
        val rangeSize = endIndex - startIndex
        AbstractList.checkRangeIndexes(destinationOffset, destination + rangeSize, destination.size) { 
            if (source is ArrayBufferView) {
                AbstractList.checkRangeIndexes(startIndex, endIndex, source.size)
                for (i in 0 until rangeSize) {
                    destination[destinationOffset + i] = source[startIndex + i]
                }
            } else {
                destinationOffset += startIndex
                for (i in 0 until rangeSize) {
                    destination[destinationOffset + i] = source[startIndex + i]
                }
            }
        }
    }
}

Open Source Used In DNAC Platform Antelope 3893
null

/*
 * Copyright 2010-2018 JetBrains s.r.o.
 * Use of this source code is governed by the Apache 2.0 license that can be
 * found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CollectionsKt")

package kotlin.collections

/**
 * Returns the given iterator itself. This allows to use an instance of iterator in a `'for` loop.
 * @sample samples.collections.Enumerators.iterator
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> Iterator<T>.iterator(): Iterator<T> = this

/**
 * Returns an [Iterator] that wraps each element produced by the original iterator into an
 * [IndexedValue] containing the index of that element and the element itself.
 * @sample samples.collections.Enumerators.indexed
 */
public fun <T> Iterator<T>.withIndex(): Iterator<IndexedValue<T>> = IndexingIterator(this)

/**
 * Performs the given [operation] on each element of this [Iterator].
 * @sample samples.collections.Enumerators.forEach
 */
public inline fun <T> Iterator<T>.forEach(operation: (T) -> Unit): Unit {
    for (element in this) operation(element)
}

/**
 * Iterators transforming original `iterator` into iterator of [IndexedValue], counting index from zero.
 */
internal class IndexingIterator<out T>(private val iterator: Iterator<T>) : Iterator<IndexedValue<T>> {
    private var index = 0
    final override fun hasNext(): Boolean = iterator.hasNext()
    final override fun next(): IndexedValue<T> = IndexedValue(checkIndexOverflow(index++), iterator.next())
}

/*
 * Copyright 2010-2019 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
 * found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("ComparisonsKt")

package kotlin.comparisons

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*

/**
 * Returns the greater of two values.
 * If values are equal, returns the first one.
 */
@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> maxOf(a: T, b: T): T

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Byte, b: Byte): Byte

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Short, b: Short): Short

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Int, b: Int): Int

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Long, b: Long): Long

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> maxOf(a: T, b: T, c: T): T

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
public expect fun maxOf(a: Short, b: Short): Short

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
public expect fun maxOf(a: Int, b: Int): Int

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
public expect fun maxOf(a: Float, b: Float): Float

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
public expect fun maxOf(a: Double, b: Double): Double

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
public expect fun maxOf(a: T, b: T, c: T): T

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
public expect fun maxOf(a: Byte, b: Byte, c: Byte): Byte

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
public expect fun maxOf(a: Short, b: Short, c: Short): Short

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
public expect fun maxOf(a: Int, b: Int, c: Int): Int

*/

Open Source Used In DNAC Platform Antelope 3894
/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Long, b: Long, c: Long): Long

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Float, b: Float, c: Float): Float

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
public expect fun <T> maxOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {
    return maxOf(a, maxOf(b, c, comparator), comparator)
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> minOf(a: T, b: T): T {
    return minOf(a, minOf(b, comparator), comparator)
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Byte, b: Byte): Byte

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Short, b: Short): Short

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Int, b: Int): Int

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Long, b: Long): Long

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Float, b: Float): Float

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Double, b: Double): Double

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
public fun <T> minOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {
    return minOf(a, minOf(b, c, comparator), comparator)
}

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
public fun <T> minOf(a: T, b: T, comparator: Comparator<in T>): T {
    return if (comparator.compare(a, b) <= 0) a else b
}

/**
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("MapsKt")

package kotlin.collections

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
import kotlin.random.*
import kotlin.ranges.Contains
import kotlin.ranges.reversed
import kotlin.ranges.<caret>
Open Source Used In DNAC Platform Antelope 3896

/*
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each entry of original map.
 */
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Performs the given [action] on each entry.
 */
@kotlin.internal.HidesMembers
public inline fun <K, V, R> Map<out K, V>.forEach(action: (Map.Entry<K, V>) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Returns the first entry yielding the largest value of the given function or 'null' if there are no entries.
 */
    return entries.maxBy(selector)
}

/**
 * Returns the first entry having the smallest value according to the provided [comparator] or 'null' if there are no entries.
 */
    return entries.minBy(selector)
}

/**
 * Returns the first entry having the smallest value according to the provided [comparator] or 'null' if there are no entries.
 */
    return entries.minBy(selector)
}

/**
 * Returns the first entry having the smallest value according to the provided [comparator] or 'null' if there are no entries.
 */
public inline fun <K, V> Map<out K, V>.forEach(action: (Map.Entry<K, V>) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Returns `true` if all entries match the given [predicate].
 */
public inline fun <K, V> Map<out K, V>.all(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {
    return true
}

/**
 * Returns `true` if at least one entry matches the given [predicate].
 */
public inline fun <K, V> Map<out K, V>.any(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {
    return true
}

/**
 * Returns the number of entries in this map.
 */
@kotlin.internal.InlineOnly
public inline fun <K, V> Map<out K, V>.count(): Int {
    return size
}

/**
 * Returns the number of entries matching the given [predicate].
 */
public inline fun <K, V> Map<out K, V>.count(predicate: (Map.Entry<K, V>) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}
 return entries.minWith(comparator) ?: return null
}

/**
 * Returns true if the map has no entries.
 */
public fun <K, V> Map.Entry<K, V>.isEmpty(): Boolean = entries.isEmpty()

/**
 * Returns 'true' if no entries match the given [predicate].
 */
public fun <K, V> @sample samples.collections.Collections.Aggregates.none(map: Map.Entry<K, V>): Boolean = entries.minWith(comparator) ?: return false

/**
 * Performs the given [action] on each entry and returns the map itself Afterwards.
 */
public fun <K, V> Map.Entry<K, V>.forEach(action: (Map.Entry<K, V>) -> Unit): Unit = M.forEach { element

/**
 * Creates an [Iterable] instance that wraps the original map returning its entries when being iterated.
 */

/**
 * Creates a [Sequence] instance that wraps the original map returning its entries when being iterated.
 */
public fun <K, V> Map.Entry<K, V>.asSequence(): Sequence<Map.Entry<K, V>> = entries.asSequence()


Sequence<T>.first(): T { val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException("Sequence is empty.")
    return iterator.next()
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if the sequence is empty.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.first(predicate: (T) -> Boolean): T {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Sequence contains no element matching the predicate.")
}

/**
 * Returns first index of [element], or -1 if the sequence does not contain such element.
 * The operation is _terminal_.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (element == item)
            return index
        index++
    }
    return -1
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the sequence does not contain such element.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.indexOfFirst(predicate: (T) -> Boolean): Int {
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (predicate(item))
            return index
        index++
    }
    return -1
}

/**
 * Returns last index of [element], or -1 if the sequence does not contain such element.
 * The operation is _terminal_.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.lastIndexOf(element: T): Int {
    var lastIndex = -1
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (element == item)
            lastIndex = index
        index++
    }
    return lastIndex
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the sequence is empty.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.last(): T {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException("Sequence is empty.")
    var last = iterator.next()
    while (iterator.hasNext())
        last = iterator.next()
    return last
}

/**
 * Returns last index of [element], or -1 if the sequence does not contain such element.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.lastIndexOf(element: T): Int {
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (element == item)
            index++
        index++
    }
    return index
}

/**
 * Returns last element matching the given [predicate], or ‘null’ if no such element was found.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.lastOrNull(predicate: (T) -> Boolean): T? {
    var last: T? = null
    for (element in this) {
        if (predicate(element))
            last = element
    }
    return last
}

/**
 * Returns the single element, or throws an exception if the sequence is empty or has more than one element.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.single(): T { val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException("Sequence is empty.")
    val single = iterator.next()
    if (iterator.hasNext())
        throw IllegalArgumentException("Sequence has more than one element.")
    return single
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.single(predicate: (T) -> Boolean): T { val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException("Sequence is empty.")
    val single = iterator.next()
    if (iterator.hasNext())
        throw IllegalArgumentException("Sequence has more than one element.")
    return single
}
Boolean): T \{ \n    var single: T? = null
    var found = false
    for (element in this) {  
        if (predicate(element)) {  
            if (found) throw IllegalArgumentException("Sequence contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Sequence contains no element matching the predicate.")

    return single
}

/**
 * Returns single element, or `null` if the sequence is empty or has more than one element.
 */
public fun <T> Sequence<T>.singleOrNull(): T? {
    val iterator = iterator()
    if (!iterator.hasNext())
        return null
    val single = iterator.next()
    if (iterator.hasNext())
        return null
    return single
}

/**
 * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.
 */
public inline fun <T> Sequence<T>.singleOrNull(predicate: (T) -> Boolean): T? {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns a sequence containing all elements except first [n] elements.
 * The operation is _intermediate_ and _stateless_.
 */
public fun <T> Sequence<T>.drop(n: Int): Sequence<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return when {
        n == 0 -> this
        this is DropTakeSequence -> this.drop(n)
        else -> DropSequence(this, n)
    }
}

/**
 * Returns a sequence containing all elements except first elements that satisfy the given [predicate].
 */
public fun <T> Sequence<T>.dropWhile(predicate: (T) -> Boolean): Sequence<T> {
    return DropWhileSequence(this, predicate)
}

/**
 * Returns a sequence containing only elements matching the given [predicate].
 */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    return FilteringSequence(this, true, predicate)
}

/**
 * Returns a sequence containing only elements matching the given [predicate].
 */
public fun <T> Sequence<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): Sequence<T> {
    return TransformingSequence(FilteringSequence(IndexingSequence(this), true, { predicate(it.index, it.value) }), { it.value })
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {
    forEachIndexed {
        index, element ->
            if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Returns a sequence containing all elements that are instances of specified type parameter R.
 */
public inline fun <reified R> Sequence<*>.filterIsInstance(): Sequence<R> {
    return filter { it is R } as Sequence<R>
}

/**
 * Appends all elements that are instances of specified type parameter R to the given [destination].
 */
public fun <C : MutableCollection<in T>> Sequence<T>.filterIsInstanceTo(destination: C) {
    for (element in this) if (element is R)
        destination.add(element)
}

/**
 * Returns a sequence containing all elements not matching the given [predicate].
 */
public fun <T> Sequence<T>.filterNot(predicate: (T) -> Boolean): Sequence<T> {
    return FilteringSequence(this, false, predicate)
}

/**
 * Returns a sequence containing all elements that are not `null`.
 */
public fun <T : Any> Sequence<T?>.filterNotNull(): Sequence<T> {
    return filterNot { it == null } as Sequence<T>
}

/**
 * Returns a sequence containing all elements that are not null.
 */
public fun <C :
MutableCollection<in T>, T: Any> Sequence<T>.filterNotNullTo(destination: C): C {
    for (element in this) if (element != null) destination.add(element)
    return destination
}

/**
 * Appends all elements not matching
 * the given [predicate] to the given [destination].
 * 
 * The operation is _terminal_.
 */
public inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * 
 * The operation is _terminal_.
 */
public inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a sequence containing first [n] elements.
 * 
 * The operation is _intermediate_ and _stateless_.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.take(n: Int): Sequence<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return when {
        n == 0 -> emptySequence()
        this is DropTakeSequence -> this.take(n)
        else -> TakeSequence(this, n)
    }
}

/**
 * Returns a sequence containing elements matching the given [predicate].
 * 
 * The operation is _intermediate_ and _stateless_.
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)
}

/**
 * Returns a sequence that yields elements of this sequence sorted
 * according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * 
 * The returned sequence is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * 
 * The operation is _intermediate_ and _stateful_.
 */
public fun <T : Comparable<T>> Sequence<T>.sorted(): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sorted.toMutableList()
            sortedList.sort()
            return sortedList.iterator()
        }
    }
}

/**
 * Returns a sequence that yields elements of this sequence sorted
 * descending according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * 
 * The operation is _intermediate_ and _stateful_.
 */
public fun <T : Comparable<T>> Sequence<T>.sortedDescending(): Sequence<T> {
    return sortedWith(reverseOrder())
}

/**
 * Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given sequence.
 * 
 * The operation is _terminal_.
 */
public inline fun <T, K, V> Sequence<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    return associateTo(LinkedHashMap<K, V>(), transform)
}

/**
 * Returns a sequence that yields elements of this sequence sorted
 * descending according to the specified [comparator].
 * 
 * The operation is _intermediate_ and _stateful_.
 */
public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}
The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_.

```kotlin
public inline fun <T, K> Sequence<T>.associateBy(keySelector: (T) -> K): Map<K, T> { return associateByTo(LinkedHashMap<K, T>(), keySelector)
}
```

```kotlin
public inline fun <T, K, V> Sequence<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> { return associateByTo(LinkedHashMap<K, V>(), keySelector, valueTransform)
}
```

```kotlin
public inline fun <T, K, M : MutableMap<in K, in T>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K): M { return associateTo(destination.put(keySelector(element), element))
}
```

```kotlin
public inline fun <T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M { return associateTo(destination.put(keySelector(element), valueTransform(element)))
}
```

```kotlin
public inline fun <T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M { return destination += transform(element)
}
```

```kotlin
public inline fun <K, V> Sequence<K>.associateWith(valueSelector: (K) -> V): Map<K, V> { return associateWithTo(LinkedHashMap<K, V>(), valueSelector)
}
```

```kotlin
public inline fun <K, V, M : MutableMap<in K, in V>> Sequence<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M { return associateWithTo(destination.put(element, valueSelector(element)))
}
```

```kotlin
public fun <T, C : MutableCollection<in T>> Sequence<T>.toCollection(destination: C): C { return destination.addAll(element)
}
```

```kotlin
public fun <T> Sequence<T>.toHashSet(): HashSet<T> { return toCollection(HashSet<T>())
}
```

```kotlin
public fun <T> Sequence<T>.toList(): List<T> { return this.toMutableList().optimizeReadOnlyList()
}
```

```kotlin
public fun <T> Sequence<T>.toMutableList(): MutableList<T> { return toCollection(ArrayList<T>())
}
```

```kotlin
public fun <T> Sequence<T>.toSet(): Set<T> { return toCollection(LinkedHashSet<T>())
}
```
R> Sequence<T>.mapIndexed(transform: (index: Int, T) -> R): Sequence<R> { return TransformingIndexedSequence(this, transform) }\n
** Returns a sequence containing only the non-null results of applying the given [transform] function to each element and its index in the original sequence.\n
* @param [transform] function that takes the index of an element and the element itself \n* and returns the result of the transform applied to the element.\n
* The operation is _intermediate_ and _stateless_. \n* @public fun <T, R : Any, C : MutableCollection<R>> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C { \n var index = 0 \n for (item in this) { \n destination.add(transform(checkIndexOverflow(index++), item)) }\n return destination }\n
** Applies the given [transform] function to each element and its index in the original sequence \n* and appends only the non-null results to the given [destination].\n
* The operation is _terminal_. \n* @param [transform] function that takes the index of an element and the element itself \n* and returns the result of the transform applied to the element.\n
* The operation is _terminal_. \n* @public inline fun <T, R : Any, C : MutableCollection<R>> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C { \n var index = 0 \n for (item in this) { \n destination.add(transform(checkIndexOverflow(index++), item)) }\n return destination }\n
** Applies the given [transform] function to each element and its index in the original sequence \n* and appends only the non-null results to the given [destination].\n
* The operation is _terminal_. \n* @param [transform] function that takes the index of an element and the element itself \n* and returns the result of the transform applied to the element.\n
* The operation is _terminal_. \n* @public inline fun <T, R : Any, C : MutableCollection<R>> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C { \n var index = 0 \n for (item in this) { \n destination.add(transform(checkIndexOverflow(index++), item)) }\n return destination }\n
** Applies the given [transform] function to each element and its index in the original sequence \n* and appends only the non-null results to the given [destination].\n
* The operation is _terminal_. \n* @param [transform] function that takes the index of an element and the element itself \n* and returns the result of the transform applied to the element.\n
* The operation is _terminal_. \n* @public inline fun <T, R : Any, C : MutableCollection<R>> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C { \n var index = 0 \n for (item in this) { \n destination.add(transform(checkIndexOverflow(index++), item)) }\n return destination }\n
** Applies the given [transform] function to each element and its index in the original sequence \n* and appends only the non-null results to the given [destination].\n
* The operation is _terminal_. \n* @param [transform] function that takes the index of an element and the element itself \n* and returns the result of the transform applied to the element.

* The operation is _intermediate_ and _stateless_. \n* @public fun <T> Sequence<T>.withIndex(): Sequence<IndexedValue<T>> { return IndexingSequence(this) }\n
** Returns a sequence containing only distinct elements from the given sequence.\n
* The elements in the resulting sequence are in the same order as they were in the source sequence.\n
* The operation is _intermediate_ and _stateless_. \n* @public fun <T> Sequence<T>.distinct(): Sequence<T> { return this.distinctBy { it } }\n
** Returns a sequence containing only elements from the given sequence having distinct keys returned by the given [selector] function.\n
* The elements in the resulting sequence are in the same order as they were in the source sequence.\n
* The operation is _intermediate_ and _stateful_. \n* @public fun <T, K> Sequence<T>.distinctBy(selector: (T) -> K): Sequence<T> { return DistinctSequence(this, selector) }\n
** Returns a mutable set containing all distinct elements from the given sequence.\n
* The returned set preserves the element iteration order of the original sequence.\n
* The operation is _terminal_. \n* @public fun <T> Sequence<T>.toMutableSet(): MutableSet<T> { \n val set = LinkedHashSet<T>() \n for (item in this) \n set.add(item) \n return set }\n
** Returns `true` if all elements match the given [predicate].\n
* The operation is _terminal_. \n* @sample samples.collections.Collections.Aggregates.all\n* @public inline fun <T> Sequence<T>.all(predicate: (T) -> Boolean): Boolean { \n for (element in this) if (!predicate(element)) return false \n return true }\n
** Returns `true` if sequence has at least one element.\n
* The operation is _terminal_. \n* @sample samples.collections.Collections.Aggregates.any\n* @public fun <T> Sequence<T>.any(): Boolean { \n return iterator().hasNext() }\n
** Returns `true` if at least one element
matches the given \[\text{predicate}\]. \nThe operation is \_terminal\_ \* @sample

samples.collections.Collections.Aggregates.anyWithPredicate\n\*/

public inline fun <T> Sequence<T>.any(predicate: (T) -> Boolean): Boolean {
    return matches the given \[\text{predicate}\]. \nThe operation is \_terminal\_ \* @sample

samples.collections.Collections.Aggregates.anyWithPredicate\n\*/

public inline fun <T> Sequence<T>.any(predicate: (T) -> Boolean): Boolean {

    for (element in this) if (predicate(element)) return true
    return false\n\*/

public inline fun <T> Sequence<T>.count(): Int {
    var count = 0
    for (element in this) count++
    return count\n\*/

public inline fun <T> Sequence<T>.count(predicate: (T) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) count++
    return count\n\*/

public inline fun <T, R> Sequence<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator\n\*/

public inline fun <T, R> Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0
    var accumulator = initial
    for (item in this) accumulator = operation(checkIndexOverflow(index++), accumulator, item)
    return accumulator\n\*/

public inline fun <T> Sequence<T>.forEach(action: (T) -> Unit): Unit {
    for (element in this) action(element)
\*/

public inline fun <T> Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {
    var index = 0
    for (item in this) action(checkIndexOverflow(index++), item)
\*/

public fun Sequence<Double>.max(): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var max = iterator.next()
    if (max.isNaN()) return max
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (max < e) max = e
    }
    return max
\*/

public fun Sequence<Float>.max(): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var max = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (max < e) max = e
    }
    return max
\*/

public fun <T : Comparable<T>> Sequence<T>.max(): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var max = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (max < e) max = e
    }
    return max
\*/

public fun <T> Sequence<T>.maxWith(comparator: Comparator<in T>): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var max = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
\*/
The operation is _terminal_.

```kotlin
public fun Sequence<Double>.min(): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    if (min.isNaN()) return min
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * The operation is _terminal_.
 */
public fun Sequence<Float>.min(): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    if (min.isNaN()) return min
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * The operation is _terminal_.
 */
public fun <T : Comparable<T>> Sequence<T>.min(): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minElem = iterator.next()
    var minValue = minElem
    do {
        val e = iterator.next()
        val v = minValue
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 * The operation is _terminal_.
 */
public inline fun <T, R : Comparable<R>> Sequence<T>.minBy(selector: (T) -> R): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minElem = iterator.next()
    var minValue = selector(minElem)
    do {
        val e = iterator.next()
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minElem
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.minWith(comparator: Comparator<in T>): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns `true` if the sequence has no elements.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.none(): Boolean {
    return !iterator().hasNext()
}

/**
 * Returns `true` if no elements match the given [predicate].
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.none(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns a sequence which performs the given [action] on each element of the original sequence as they pass through it.
 * The operation is _intermediate_ and _stateless_.
 */
@SinceKotlin("1.1")
public fun Sequence<T>.onEach(action: (T) -> Unit): Sequence<T> {
    // Implementation...
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 * The operation is _terminal_.
 */
public inline fun <S, T : S> Sequence<T>.reduce(operation: (acc: S, T) -> S): S {
    // Implementation...
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original sequence.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself and calculates the next accumulator value.
 * The operation is _terminal_.
 */
public inline fun <S, T : S> Sequence<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {
    // Implementation...
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the sequence.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.sumBy(selector: (T) -> Int): Int {
    // Implementation...
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the sequence.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.sumByDouble(selector: (T) -> Double): Double {
    // Implementation...
}
```
(T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns an original collection containing all the non-`null` elements, throwing an
 * [IllegalStateException] if there are any `null` elements.
 * The operation is _intermediate_ and _stateless_.
 */
public fun <T : Any> Sequence<T?>? requireNoNulls(): Sequence<T> {
    return return windowed(size, size, partialWindows = true) { it ?: throw IllegalStateException("null element found in $this.") |} |] | | |
}

/**
 * Returns a sequence containing all elements of the original sequence except the elements contained in the given [elements].
 * The operation is _intermediate_ and _stateful_.
 * The elements collection may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 */
public operator fun <T> Sequence<T>.minus(elements: Iterable<T>): Sequence<T> {
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.convertToSetForSetOperation()
            if (other.isEmpty())
                return this
            else
                return this.filterNot { it in other }.iterator()
        }
    }
}

/**
 * Returns a sequence containing all elements of the original sequence except the elements contained in the given [elements] array.
 * The operation is _intermediate_ and _stateless_.
 */
public operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty())
        return this
    else
        return this.filterNot { it in other }.iterator()
}

/**
 * Returns a sequence containing all elements of the original sequence except the elements contained in the given [elements] array.
 * The operation is _intermediate_ and _stateless_.
 */
public operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty())
        return this
    else
        return this.filterNot { it in other }.iterator()
}

/**
 * Returns a sequence containing all elements of the original sequence except the elements contained in the given [elements] array.
 * The operation is _intermediate_ and _stateless_.
 */
public operator fun <T> Sequence<T>.minus(element: T): Sequence<T> {
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            var removed = false
            return this.filter { if (!removed && it == element) {
                removed = true; false
            } else true }.iterator()
        }
    }
}

/**
 * Returns a sequence containing all elements of the original sequence except the elements contained in the given [elements] array.
 * The operation is _intermediate_ and _stateful_.
 * The [elements] array may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 */
public operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty())
        return this
    else
        return this.filterNot { it in other }.iterator()
}

/**
 * Returns a sequence containing all elements of the original sequence except the elements contained in the given [elements] array.
 * The operation is _intermediate_ and _stateful_.
 * The elements collection may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 */
public operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty())
        return this
    else
        return this.filterNot { it in other }.iterator()
}

/**
 * Returns a sequence containing all elements of the original sequence except the elements contained in the given [elements] array.
 * The operation is _intermediate_ and _stateful_.
 * The elements collection may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 */
public operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty())
        return this
    else
        return this.filterNot { it in other }.iterator()
}

/**
 * Returns a sequence containing all elements of the original sequence except the elements contained in the given [elements] array.
 * The operation is _intermediate_ and _stateful_.
 * The elements collection may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 */
public operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty())
        return this
    else
        return this.filterNot { it in other }.iterator()
}
invocations.

 */
public operator fun <T> Sequence<T>.minus(elements: Sequence<T>): Sequence<T> {
    return this@minus.filterNot { it in other }
}

/**
 * Returns a sequence containing all elements of the original sequence without the first occurrence of the given [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minusElement(element: T): Sequence<T> {
    return minus(element)
}

/**
 * Splits the original sequence into pair of lists, where *first* list contains elements for which [predicate] yielded `true`, while *second* list contains elements for which [predicate] yielded `false`.
 */
public inline fun <T> Sequence<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {
    val first = ArrayList<T>()
    val second = ArrayList<T>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Returns a sequence containing all elements of the original sequence and then the given [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.plusElement(element: T): Sequence<T> {
    return plus(element)
}

/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] array.
 */
public operator fun <T> Sequence<T>.plus(elements: Array<out T>): Sequence<T> {
    return this.plus(elements.asList())
}

/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] collection.
 */
public operator fun <T> Sequence<T>.plus(elements: Iterable<T>): Sequence<T> {
    return sequenceOf(this, elements.asSequence()).flatten()
}

/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] sequence.
 */
public operator fun <T> Sequence<T>.plus(elements: Sequence<T>): Sequence<T> {
    return sequenceOf(this, elements).flatten()
}

/**
 * Returns a sequence of snapshots of the window of the given [size] sliding along this sequence with the given [step], where each snapshot is a list.
 */
@SinceKotlin("1.2")
    return windowedSequence(size, step, partialWindows, reuseBuffer = true)
}

/**
 * Returns a sequence of results of applying the given [transform] function to an each list representing a view over the window of the given [size] sliding along this sequence with the given [step], where each list is a snapshot of partial windows won't be preserved.
 */
    return windowedSequence(size, step, partialWindows, reuseBuffer = true)
}
step the number of elements to move the window forward by on each step, by default 1

partialWindows controls whether or not to keep partial windows in the end if any. * by default `false` which means partial windows won't be preserved

* @param

windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): Sequence<R> | \{\n  return returnWindowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform)\n\} \* @ sample

samples.collections.Sequences.Transformations.averageWindows

* @SinceKotlin("1.2")

public fun <T, R> Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): Sequence<R> | \{\n  return returnWindowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform)\n\} \* @ sample

samples.collections.Sequences.Transformations.averageWindows

* @SinceKotlin("1.2")

public fun <T, R> Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): Sequence<R> | \{\n  return returnWindowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform)\n\} \* @ sample

samples.collections.Sequences.Transformations.averageWindows

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip

* @SinceKotlin("1.2")

public function to each pair of elements. * The resulting sequence ends as soon as the shortest input sequence ends. * The operation is _intermediate_and_stateless_ * \* @ sample

samples.collections.Sequences.Transformations.zip
a [Sequence].
*/

@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.asSequence(): Sequence<T> {
    return this
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfByte")
public fun Sequence<Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfShort")
public fun Sequence<Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfInt")
public fun Sequence<Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Sequence<Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Sequence<Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfDouble")
public fun Sequence<Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Sequence<Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Sequence<Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Sequence<Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Sequence<Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Sequence<Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Sequence<Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SetsKt")

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns a set containing all elements of the original
 * set except the given [element].
 * The returned set preserves the element iteration order of the original set.
 */
public operator fun <T> Set<T>.minus(element: T): Set<T> { val result =
var removed = false
return this.filterTo(result) { if (!removed && & it == element) { removed = true; false } else true }\n
* Returns a set containing all elements of the original set except the elements contained in the given [elements] array.\n
* The returned set preserves the element iteration order of the original set.\n
* The [elements] array may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.minus(elements: Array<out T>): Set<T> {\n  val result = LinkedHashSet<T>(this)\n  result.removeAll(elements)\n  return result\n}

* Returns a set containing all elements of the original set except the elements contained in the given [elements]\n  collection.\n
* The returned set preserves the element iteration order of the original set.\n
* The [elements] collection may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.minus(elements: Iterable<T>): Set<T> {\n  val other = elements.convertToSetForSetOperationWith(this)\n  if (other.isEmpty())\n      return this\n  if (other is Set)\n      return this.filterNotTo(LinkedHashSet<T>()) { it in other }\n  var result = LinkedHashSet<T>(this)\n  result.removeAll(other)\n  return result\n}

* Returns a set containing all elements of the original set except the elements contained in the given [elements]\n  sequence.\n
* The returned set preserves the element iteration order of the original set.\n
* The [elements] sequence may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.minus(elements: Sequence<T>): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(size + 1))\n  result.addAll(this)\n  result.add(element)\n  return result\n}

* Returns a set containing all elements of the original set except the given [element].\n
* The returned set preserves the element iteration order of the original set.\n
* The [element] may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.minusElement(element: T): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(this.size + elements.size))\n  result.addAll(this)\n  result.addAll(elements)\n  return result\n}

* Returns a set containing all elements of the original set and the given [elements] collection.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The [elements] collection may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.plus(element: T): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(size + 1))\n  result.addAll(this)\n  result.add(element)\n  return result\n}

* Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The [element] may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.plus(element: T): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(this.size + elements.size))\n  result.addAll(this)\n  result.addAll(elements)\n  return result\n}

* Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The [element] may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.plus(elements: Array<out T>): Set<T> {\n  val result = LinkedHashSet<T>(this)\n  result.addAll(elements)\n  return result\n}

* Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The [element] may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.plus(elements: Sequence<T>): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(this.size + 2))\n  result.addAll(this)\n  result.addAll(elements)\n  return result\n}

* Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The [element] may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.plus(elements: Sequence<T>): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(this.size + 2))\n  result.addAll(this)\n  result.addAll(elements)\n  return result\n}

* Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The [element] may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.plus(elements: Sequence<T>): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(this.size + 2))\n  result.addAll(this)\n  result.addAll(elements)\n  return result\n}

* Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The returned set preserves the element iteration order of the original set.\n
* The [element] may be converted to a [HashSet] to speed up the operation, thus the elements are required to have\n  * a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.\n
*public operator fun <T> Set<T>.plus(elements: Sequence<T>): Set<T> {\n  val result = LinkedHashSet<T>(mapCapacity(this.size + 2))\n  result.addAll(this)\n  result.addAll(elements)\n  return result\n}
sequence.

```kotlin
/**
 * Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.
 */
public fun CharSequence.getOrNull(index: Int): Char? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}

/**
 * Returns index of the first character matching the given [predicate], or `-1` if no such character was found.
 */
public inline fun CharSequence.indexOfFirst(predicate: (Char) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

/**
 * Returns index of the last character matching the given [predicate], or `-1` if no such character was found.
 */
public inline fun CharSequence.indexOfLast(predicate: (Char) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}
```

```kotlin
/**
 * Returns a random character from this char sequence.
 * 
 * @throws NoSuchElementException if this char sequence is empty.
 */
val randomChar: Char = this.randomChar
```

```kotlin
open Source Used In DNAC Platform Antelope
```

3911

---

`Open Source Used In DNAC Platform Antelope 3911`
Since Kotlin("1.3")\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.random(): Char {
    return random(Random)
}

/**
 * Returns a random character from this char sequence using the specified source of randomness.
 * @returns a random character from this char sequence.
 * @throws NoSuchElementException if this char sequence is empty.
 */
@SinceKotlin("1.3")\n@kotlin.internal.InlineOnly
public inline fun CharSequence.random(random: Random): Char {
    if (isNotEmpty())
        throw NoSuchElementException("Sequence is empty.")
    return get(random.nextInt(length))
}

/**
 * Returns the single character or throws an exception if the sequence is empty or has more than one character.
 */
@SinceKotlin("1.3")
public fun CharSequence.single(): Char {
    return when (length) {
        0 -> throw
            NoSuchElementException("Sequence is empty.")
        1 -> this[0]
        else -> throw
            IllegalArgumentException("Sequence has more than one element.")
    }
}

/**
 * Returns the single character, or throws an exception if there is no or more than one matching character.
 */
@SinceKotlin("1.3")
public fun CharSequence.single(predicate: (Char) -> Boolean): Char {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Sequence contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw
        NoSuchElementException("Sequence contains no character matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single
}

/**
 * Returns a subsequence of this char sequence with the first [n] characters removed.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.text.Strings.drop
 */
public fun CharSequence.drop(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return subSequence(n.coerceAtMost(length), length)
}

/**
 * Returns a string with the first [n] characters removed.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.text.Strings.drop
 */
public fun String.drop(n: Int): String {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return substring(n.coerceAtMost(length))
}

/**
 * Returns a subsequence of this char sequence containing all characters except last characters that satisfy the given 
 * [predicate].
 */
public inline fun CharSequence.dropLastWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index])
            return subSequence(0, index + 1)
    return ""
}

/**
 * Returns a string containing all characters except last characters that satisfy the given 
 * [predicate].
 */
public inline fun String.dropLastWhile(predicate: (Char) -> Boolean): String {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index])
            return substring(0, index + 1)
    return ""
}

/**
 * Returns a subsequence of this char sequence containing all characters except first characters that satisfy the given 
 * [predicate].
 */
public inline fun CharSequence.dropWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in this.indices)
        if (!predicate(this[index])
            return subSequence(index, length)
    return ""
}

/**
 * Returns a string containing all characters except first characters that satisfy the given 
 * [predicate].
 */
public inline fun String.dropWhile(predicate: (Char) -> Boolean): String {
    for (index in this.indices)
        if (!predicate(this[index])
            return substring(index)
    return ""


```kotlin
fun filter(predicate: (Char) -> Boolean): CharSequence {
    return filterTo(StringBuilder(), predicate)
}

fun filter(predicate: (Char) -> Boolean): String {
    return filterTo(StringBuilder(), predicate).toString()
}

fun filterIndex(predicate: (Int, Char) -> Boolean): CharSequence {
    return filterIndexedTo(StringBuilder(), predicate)
}

fun filterIndex(predicate: (Int, Char) -> Boolean): String {
    return filterIndexedTo(StringBuilder(), predicate).toString()
}

fun filterIndexedTo(destination: Appendable, predicate: (Int, Char) -> Boolean): Appendable {
    forEachIndexed { index, element ->
        if (predicate(index, element))
            destination.append(element)
    }
    return destination
}

fun filterNot(predicate: (Char) -> Boolean): CharSequence {
    return filterNotTo(StringBuilder(), predicate)
}

fun filterNot(predicate: (Char) -> Boolean): String {
    return filterNotTo(StringBuilder(), predicate).toString()
}

fun filterNotTo(destination: Appendable, predicate: (Char) -> Boolean): Appendable {
    for (element in this) if (!predicate(element))
        destination.append(element)
    return destination
}

fun filterTo(destination: Appendable, predicate: (Char) -> Boolean): Appendable {
    for (index in 0 until length) {
        val element = get(index)
        if (predicate(element))
            destination.append(element)
    }
    return destination
}

fun slice(indices: IntRange): CharSequence {
    if (indices.isEmpty()) return"
    return subSequence(indices)
}

fun slice(indices: IntRange): String {
    if (indices.isEmpty()) return"
    return substring(indices)
}

fun slice(indices: Iterable<Int>): CharSequence {
    val size = indices.collectionSizeOrDefault(10)
    var result = StringBuilder(size)
    for (i in indices) {
        result.append(get(i))
    }
    return result
}

fun slice(indices: Iterable<Int>): String {
    return (this as charSequence).slice(indices).toString()
}

fun take(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return subSequence(0, n.coerceAtMost(length))
}

fun take(n: Int): String {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return substring(0, n.coerceAtMost(length))
}
```

---

**Open Source Used In DNAC Platform Antelope 3913**
characters from this char sequence, or the entire char sequence if this char sequence is shorter.

* @throws IllegalArgumentException if n is negative.

* @sample samples.text.Strings.take
*/
public fun CharSequence.takeLast(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return subSequence(length - n.coerceAtMost(length), length)
}

/**
 * Returns a string containing the last n characters from this string, or the entire string if this string is shorter.

* @throws IllegalArgumentException if n is negative.

* @sample samples.text.Strings.take
*/
public fun String.takeLast(n: Int): String {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return substring(length - n.coerceAtMost(length))
}

/**
 * Returns a subsequence of this char sequence containing last characters that satisfy the given [predicate].

* @sample samples.text.Strings.take
*/
public inline fun CharSequence.takeLastWhile(predicate: (Char) -> Boolean):
    CharSequence {
        for (index in lastIndex downTo 0) {
            if (!predicate(this[index])) {
                return subSequence(index + 1, length)
            }
        }
        return subSequence(0, length)
    }

/**
 * Returns a string containing last characters that satisfy the given [predicate].

* @sample samples.text.Strings.take
*/
public inline fun String.takeLastWhile(predicate: (Char) -> Boolean): String {
    for (index in lastIndex downTo 0) {
        if (!predicate(get(index))) {
            return substring(index + 1)
        }
    }
    return this
}

/**
 * Returns a subsequence of this char sequence containing first characters that satisfy the given [predicate].

* @sample samples.text.Strings.take
*/
public inline fun CharSequence.takeWhile(predicate: (Char) -> Boolean):
    CharSequence {
        for (index in 0 until length)
            if (!predicate(get(index))) {
                return subSequence(0, index)
            }
        return subSequence(0, length)
    }

/**
 * Returns a string containing first characters that satisfy the given [predicate].

* @sample samples.text.Strings.take
*/
public inline fun String.takeWhile(predicate: (Char) -> Boolean): String {
    for (index in 0 until length)
        if (!predicate(get(index))) {
            return substring(0, index)
        }
    return this
}

/**
 * Returns a char sequence with characters in reversed order.
 */
public fun CharSequence.reversed(): CharSequence {
    return StringBuilder(this).reverse()
}

/**
 * Returns a string with characters in reversed order.
 */
@kotlin.internal.InlineOnly
public inline fun String.reversed(): String {
    return (this as CharSequence).reversed().toString()
}

/**
 * Returns a [Map] containing key-value pairs provided by [transform] function applied to characters of the given char sequence.

* If any of two pairs would have the same key the last one gets added to the map.

* The returned map preserves the entry iteration order of the original char sequence.

* @sample samples.text.Strings.associate
*/
public inline fun <K, V> CharSequence.associate(transform: (Char) -> Pair<K, V>):
    Map<K, V> {
        val capacity = mapCapacity(length).coerceAtLeast(16)
        return associateTo(LinkedHashMap<K, V>(capacity), transform)
    }

/**
 * Returns a [Map] containing characters from the given char sequence indexed by the key returned from [keySelector] function applied to each character.

* If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

* The returned map preserves the entry iteration order of the original char sequence.

* @sample samples.text.Strings.associateBy
*/
public inline fun <K> CharSequence.associateBy(keySelector: (Char) -> K): Map<K, Char> {
    val capacity = mapCapacity(length).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to characters of the given char sequence.

* If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

* The returned map preserves the entry iteration order of the original char sequence.

* @sample samples.text.Strings.associateBy
*/
public inline fun <K, V> CharSequence.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V):
    Map<K, V> {
        val capacity = mapCapacity(length).coerceAtLeast(16)
        return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
    }

/**
 * Populates and returns the mutable map with key-value pairs.

* @throws IllegalStateException if [n] is negative.

* @sample samples.text.Strings.associateTo
*/
public inline fun <K, M : MutableMap<in K, in Char>>
    CharSequence.associateTo(destination: M, keySelector: (Char) -> K): M {
        for (element in this) {
            destination.put(keySelector(element), element)
        }
        return destination
    }

Open Source Used In DNAC Platform Antelope 3914
Open Source Used In DNAC Platform Antelope 3915

[destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to characters of the given char sequence. If any two characters would have the same key returned by [keySelector] the last one gets added to the map. 

If any two characters would have the same key the last one gets added to the map.

*/
public inline fun <K, V, M : MutableMap<in K, in V>> CharSequence.associateByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs provided by the [transform] function applied to each character of the given char sequence.
 * 
 * If any two characters are equal, the last one overwrites the former value in the map.
 * 
 * @sample samples.text.Strings.associateWith
 */
@SinceKotlin("1.3")
public inline fun <V> CharSequence.associateWith(valueSelector: (Char) -> V): Map<Char, V> {
    val result = LinkedHashMap<Char, V>(mapCapacity(length).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/**
 * Groups characters of the original char sequence by the key returned by the given [keySelector] function applied to each character and returns a map where each group key is associated with a list of corresponding characters.
 * 
 * The returned map preserves the entry iteration order of the keys produced from results of [transform] function being invoked on each character of original char sequence.
 * 
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K> CharSequence.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {
    return groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)
}

/**
 * Groups values returned by the [valueSelector] function applied to each character of the original char sequence by the key returned by the given [keySelector] function applied to each character and returns a map where each group key is associated with a list of corresponding characters.
 * 
 * The returned map preserves the entry iteration order of the keys produced from results of [transform] function being invoked on each character of original char sequence.
 * 
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K> CharSequence.groupBy(keySelector: (Char) -> K): Map<K, MutableList<Char>> {
    return groupByTo(ArrayList<Char>(length))
}
from the original char sequence.

@sample samples.collections.Collections.Transformations.groupByKeysAndValues

The method `groupByTo` takes a `Map<K, MutableList<V>>` as the destination map and returns it. It creates a `LinkedHashMap` as the destination map and iterates over each element in the `this` sequence. For each element, it extracts the key using the `keySelector` function and then uses the `valueTransform` function to transform the value. The transformed value is then added to the list associated with the key in the destination map.

```kotlin
public inline fun <K, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K): M {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

**Groups values returned by the [valueTransform] function applied to each character of the original char sequence by the key returned by the given [keySelector] function applied to each character and puts to the [destination] map each group key associated with a list of corresponding characters.**

@sample samples.collections.Collections.Transformations.groupByKeysAndValues

The method `groupByTo` takes a `Map<K, MutableList<V>>` as the destination map and returns it. It creates a `LinkedHashMap` as the destination map and iterates over each element in the `this` sequence. For each element, it extracts the key using the `keySelector` function and then uses the `valueTransform` function to transform the value. The transformed value is then added to the list associated with the key in the destination map.

```kotlin
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    var index = 0
    for (item in this)
        destination.add(valueTransform(item))
    return destination
}
```

**Groups values returned by the [valueTransform] function applied to each character and its index in the original char sequence by the key returned by the given [keySelector] function applied to each character and puts to the [destination] map each group key associated with a list of corresponding values.**

@sample samples.collections.Collections.Transformations.groupByKeysAndValues

The method `groupByTo` takes a `Map<K, MutableList<V>>` as the destination map and returns it. It creates a `LinkedHashMap` as the destination map and iterates over each element in the `this` sequence. For each element, it extracts the key using the `keySelector` function and then uses the `valueTransform` function to transform the value. The transformed value is then added to the list associated with the key in the destination map.

```kotlin
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    var index = 0
    for (item in this)
        destination.add(valueTransform(item))
    return destination
}
```

**Groups values returned by the [valueTransform] function applied to each character and its index in the original char sequence by the key returned by the given [keySelector] function applied to each character and puts to the [destination] map each group key associated with a list of corresponding values.**

@sample samples.collections.Collections.Transformations.groupByKeysAndValues

The method `groupByTo` takes a `Map<K, MutableList<V>>` as the destination map and returns it. It creates a `LinkedHashMap` as the destination map and iterates over each element in the `this` sequence. For each element, it extracts the key using the `keySelector` function and then uses the `valueTransform` function to transform the value. The transformed value is then added to the list associated with the key in the destination map.

```kotlin
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    var index = 0
    for (item in this)
        destination.add(valueTransform(item))
    return destination
}
```

**Groups values returned by the [valueTransform] function applied to each character and its index in the original char sequence by the key returned by the given [keySelector] function applied to each character and puts to the [destination] map each group key associated with a list of corresponding values.**

@sample samples.collections.Collections.Transformations.groupByKeysAndValues

The method `groupByTo` takes a `Map<K, MutableList<V>>` as the destination map and returns it. It creates a `LinkedHashMap` as the destination map and iterates over each element in the `this` sequence. For each element, it extracts the key using the `keySelector` function and then uses the `valueTransform` function to transform the value. The transformed value is then added to the list associated with the key in the destination map.

```kotlin
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    var index = 0
    for (item in this)
        destination.add(valueTransform(item))
    return destination
}
```
/**
 * Applies the given [transform] function to each character in the original char sequence
 * and appends only the non-null results to the given [destination].
 */
public inline fun <R : Any, C : MutableCollection<R>> CharSequence.mapNotNullTo(destination: C, transform: (Char) -> R?): C {
    forEach {
        transform(it)?.let { destination.add(it) }
    }
    return destination
}

/**
 * Applies the given [transform] function to each character in the original char sequence
 * and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<R>> CharSequence.mapTo(destination: C, transform: (Char) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each character of the original char sequence
 * into an [IndexedValue] containing the index of that character and the character itself.
 */
public fun CharSequence.withIndex(): Iterable<IndexedValue<Char>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns `true` if all characters match the given [predicate].
 * 
 * @sample
 * samples.collections.Collections.Aggregates.all
 */
public inline fun CharSequence.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if char sequence has at least one character.
 * 
 * @sample
 * samples.collections.Collections.Aggregates.any
 */
public fun CharSequence.any(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if at least one character matches the given [predicate].
 * 
 * @sample
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun CharSequence.any(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns the length of this char sequence.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.count(): Int {
    return length
}

/**
 * Returns the number of characters matching the given [predicate].
 */
public inline fun CharSequence.count(predicate: (Char) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to each character and current accumulator value.
 */
public inline fun <R> CharSequence.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each character and current accumulator value.
 */
public inline fun <R> CharSequence.foldRight(initial: R, operation: (Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Performs the given [action] on each character.
 */
public inline fun CharSequence.forEach(action: (Char) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each character, providing sequential index with the character.
 */
public inline fun CharSequence.forEachIndexed(action: (index: Int, Char) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

/**
 * Returns the largest character or `null` if there are no characters.
 */
public fun CharSequence.max(): Char? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}
the first character yielding the largest value of the given function or `null` if there are no characters.

```kotlin
CharSequence.maxBy(selector: (Char) -> R): Char? {
    var maxElem = this[0]
    var lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        if (selector(this[i]) > maxValue) {
            maxElem = this[i]
            maxValue = selector(this[i])
        }
    }
    return maxElem
}
```

```kotlin
CharSequence.maxWith(comparator: Comparator<in Char>): Char? {
    var min = this[0]
    for (i in 1..lastIndex) {
        if (comparator.compare(min, this[i]) > 0) min = this[i]
    }
    return min
}
```

```kotlin
CharSequence.min(): Char? {
    var min = this[0]
    for (i in 1..lastIndex) {
        if (min > this[i]) min = this[i]
    }
    return min
}
```

```kotlin
CharSequence.minBy(selector: (Char) -> R): Char? {
    var minElem = this[0]
    var lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        if (selector(this[i]) < minValue) {
            minElem = this[i]
            minValue = selector(this[i])
        }
    }
    return minElem
}
```

```kotlin
CharSequence.minWith(comparator: Comparator<in Char>): Char? {
    var min = this[0]
    for (i in 1..lastIndex) {
        if (comparator.compare(min, this[i]) < 0) min = this[i]
    }
    return min
}
```

```kotlin
CharSequence.none(): Boolean {
    return isEmpty()
}
```

```kotlin
CharSequence.none(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
CharSequence.reduce(operation: (Char, Char) -> Char): Char {
    if (isEmpty()) throw UnsupportedOperationException("Empty char sequence can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
CharSequence.reduceIndexed(operation: (Int, Char, Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")
    var accumulator = get(index - 1) while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}
```

```kotlin
CharSequence.reduceRight(operation: (Char, Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")
    var accumulator = get(index--) while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}
```
public inline fun CharSequence.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.
 */
public inline fun CharSequence.sumBy(selector: (Char) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.
 */
public inline fun CharSequence.sumByDouble(selector: (Char) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Splits this char sequence into a list of strings each not exceeding the given [size].
 * The last string in the resulting list may have less characters than the given [size].
 * @param size the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence.
 */
@SinceKotlin("1.2")
public fun CharSequence.chunked(size: Int) {
    return windowed(size, size, partialWindows = true)
}

/**
 * Splits this char sequence into several char sequences each not exceeding the given [size] and applies the given [transform] function to an each.
 * The last string in the resulting sequence may have less characters than the given [size].
 * @param size the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence.
 * @param transform the function applied to each string, must be positive and can be greater than the number of elements in this char sequence.
 */
@SinceKotlin("1.2")
public fun <R> CharSequence.chunked(size: Int, transform: (CharSequence) -> R): List<R> {
    return windowedSequence(size, size, partialWindows = true, transform = transform)
}

/**
 * Splits the original char sequence into pair of char sequences,
 * where *first* char sequence contains characters for which [predicate] yielded `true`,
 * while *second* char sequence contains characters for which [predicate] yielded `false`.
 */
public inline fun CharSequence.partition(predicate: (Char) -> Boolean): Pair<CharSequence, CharSequence> {
    val first = StringBuilder()
    val second = StringBuilder()
    for (element in this) {
        if (predicate(element)) {
            first.append(element)
        } else {
            second.append(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original string into pair of strings,
 * where *first* string contains characters for which [predicate] yielded `true`,
 * while *second* string contains characters for which [predicate] yielded `false`.
 */
public inline fun String.partition(predicate: (Char) -> Boolean): Pair<String, String> {
    val first = StringBuilder()
    val second = StringBuilder()
    for (element in this) {
        if (predicate(element)) {
            first.append(element)
        } else {
            second.append(element)
        }
    }
    return Pair(first.toString(), second.toString())
}

/**
 * Returns a list of elements
 */
of snapshots of the window of the given [size]* sliding along this char sequence with the given [step], where each
* snapshot is a string.* Several last strings may have less characters than the given [size].* Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.*
* @param size the number of elements to take in each window* @param step the number of elements to move the window forward by on each step, by default 1
* @param partialWindows controls whether or not to keep partial windows in the end if any,* by default 'false' which means partial windows won't be preserved.* @sample samples.collections.Sequences.Transformations.takeWindows
@SinceKotlin("1.2")
public fun
**CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false) {**
* Returns a list of results of applying the given [transform] function to each char sequence representing a view over the window of the given [size]* sliding along this char sequence with the given [step].* Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.* You should not store it or allow it to escape in some way, unless you made a snapshot of it.* Several last char sequences may have less characters than the given [size].* Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.*
* @param size the number of elements to take in each window* @param step the number of elements to move the window forward by on each step, by default 1
* @param partialWindows controls whether or not to keep partial windows in the end if any,* by default 'false' which means partial windows won't be preserved.* @sample samples.collections.Sequences.Transformations.averageWindows
@SinceKotlin("1.2")
public fun <R>CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): List<R> {
* Returns a sequence of results of applying the given [transform] function to each char sequence representing a view over the window of the given [size]* sliding along this char sequence with the given [step].* Several last char sequences may have less characters than the given [size].* Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.*
* @param size the number of elements to take in each window* @param step the number of elements to move the window forward by on each step, by default 1
* @param partialWindows controls whether or not to keep partial windows in the end if any,* by default 'false' which means partial windows won't be preserved.* @sample samples.collections.Sequences.Transformations.takeWindows
@SinceKotlin("1.2")
public fun
**CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<String>**
@SinceKotlin("1.2")
public fun <R>CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): Sequence<R> {
* Returns a sequence of results of applying the given [transform] function to each char sequence representing a view over the window of the given [size]* sliding along this char sequence with the given [step].* Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.* You should not store it or allow it to escape in some way, unless you made a snapshot of it.* Several last char sequences may have less characters than the given [size].* Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.*
* @param size the number of elements to take in each window* @param step the number of elements to move the window forward by on each step, by default 1
* @param partialWindows controls whether or not to keep partial windows in the end if any,* by default 'false' which means partial windows won't be preserved.* @sample samples.collections.Sequences.Transformations.averageWindows
@SinceKotlin("1.2")
public fun
**CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): Sequence<String>**

`this` and the [other] char sequences with the same index
* The returned list has length of the shortest char sequence
* `trim(predicate: (Char) -> Boolean): CharSequence`.
  * Returns a string having leading and trailing characters matching the [predicate] removed.
  * `trim(vararg chars: Char): CharSequence`.
  * Returns a sub sequence of this char sequence having trailing characters matching the [predicate] removed.

---

Open Source Used In DNAC Platform Antelope 3921
from the [chars] array removed.

```kotlin
public fun String.trim(vararg chars: Char): String = trim { it in chars }

public fun CharSequence.trimStart(vararg chars: Char): CharSequence = trimStart { it in chars }

public fun String.trimStart(vararg chars: Char): String = trimStart { it in chars }

public fun CharSequence.trimEnd(vararg chars: Char): CharSequence = trimEnd { it in chars }

public fun String.trimEnd(vararg chars: Char): String = trimEnd { it in chars }

public fun CharSequence.trim(): CharSequence = trim(Char::isWhitespace)

public fun String.trim(): String = (this as CharSequence).trim().toString()

public fun CharSequence.trimStart(): CharSequence = trimStart(Char::isWhitespace)

public fun String.trimStart(): String = (this as CharSequence).trimStart().toString()

public fun CharSequence.trimEnd(): CharSequence = trimEnd(Char::isWhitespace)

public fun String.trimEnd(): String = (this as CharSequence).trimEnd().toString()

public fun CharSequence.padStart(length: Int, padChar: Char = ' '): CharSequence {
    if (length < 0)
        throw IllegalArgumentException("Desired length $length is less than zero.")
    if (length <= this.length)
        return this.subSequence(0, this.length)

    val sb = StringBuilder(length)
    for (i in 1..(length - this.length))
        sb.append(padChar)
    sb.append(this)
    return sb
}

public fun String.padStart(length: Int, padChar: Char = ' '): String =
    (this as CharSequence).padStart(length, padChar).toString()

public fun CharSequence.padEnd(length: Int, padChar: Char = ' '): CharSequence {
    if (length < 0)
        throw IllegalArgumentException("Desired length $length is less than zero.")
    if (length <= this.length)
        return this.subSequence(0, this.length)

    val sb = StringBuilder(length)
    sb.append(this)
    for (i in 1..(length - this.length))
        sb.append(padChar)
    return sb
}

public fun String.padEnd(length: Int, padChar: Char = ' '): String =
    (this as CharSequence).padEnd(length, padChar).toString()
```

```kotlin
from the [chars] array removed.

```
public inline fun CharSequence?.isNullOrEmpty(): Boolean {
    contract {
      returns(false) implies (this@isNullOrEmpty != null)
    }

    return this == null || this.length == 0
}

/**
 * Returns `true` if this char sequence is empty (contains no characters).
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.isEmpty(): Boolean = length == 0

/**
 * Returns `true` if this char sequence is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.isNotEmpty(): Boolean = length > 0

/*
 * Returns `true` if this nullable char sequence is either
 * `null` or empty or consists solely of whitespace characters.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence?.isNullOrBlank(): Boolean {
    contract {
        returns(false) implies (this@isNullOrBlank != null)
    }

    return this == null || this.isBlank()
}

/**
 * Iterator for characters of the given char sequence.
 */
public operator fun CharSequence.iterator(): CharIterator = object : CharIterator() {
    private var index = 0

    public override fun nextChar(): Char = get(index++)

    public override fun hasNext(): Boolean = index < length
}

/** Returns the string if it is not `null`, or the empty string otherwise. */
@kotlin.internal.InlineOnly
public inline fun String?.orEmpty(): String = this ?: ""

/**
 * Returns this char sequence if it's not empty
 * or the result of calling [defaultValue] function if the char sequence is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : CharSequence, C : R =
    if (isEmpty()) defaultValue() else this

/**
 * Returns this char sequence if it is not empty and doesn't consist solely of whitespace characters,
 * or the result of calling [defaultValue] function otherwise.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <C, R> C.ifBlank(defaultValue: () -> R): R where C : CharSequence, C : R =
    if (isBlank()) defaultValue() else this

/**
 * Returns the range of valid character indices for this char sequence.
 */
public val CharSequence.indices: IntRange
    get() = 0..length - 1

/**
 * Returns the index of the last character in the char sequence or -1 if it is empty.
 */
public val CharSequence.lastIndex: Int
    get() = this.length - 1

/**
 * Returns `true` if this CharSequence has Unicode surrogate pair at the specified [index].
 */
public fun CharSequence.hasSurrogatePairAt(index: Int): Boolean {
    return index in 0..length - 2
            && this[index].isHighSurrogate()
            && this[index + 1].isLowSurrogate()
}

/**
 * Returns a substring specified by the given [range] of indices.
 */
public fun String.substring(range: IntRange): String = substring(range.start, range.endInclusive + 1)

/**
 * Returns a subsequence of this char sequence specified by the given [range] of indices.
 */
@kotlin.internal.InlineOnly
public fun String.substring(range: IntRange): String =
    subSequence(range.start, range.endInclusive + 1).toString()
occurrence of $[delimiter]$. If the string does not contain the delimiter, returns $[missingDelimiterValue] which
defaults to the original string.

*public fun String.substringBefore(delimiter: Char, missingDelimiterValue:
String) { String = this; String { val index = indexOf(delimiter) return if (index == -1) missingDelimiterValue else
substring(0, index) }}

* Returns a substring before the first occurrence of $[delimiter]$. If the string does not contain the delimiter, returns
$[missingDelimiterValue] which defaults to the original string.

*public fun String.substringBefore(delimiter: String, missingDelimiterValue: String = this) { String = this; String { val index = indexOf(delimiter) return if (index == -1) missingDelimiterValue else
substring(0, index) }}

* Returns a substring before the first occurrence of $[delimiter]$. If the string does not contain the delimiter, returns
$[missingDelimiterValue] which defaults to the original string.

*public fun String.substringAfter(delimiter: Char, missingDelimiterValue: String = this) { String = this; String { val index = indexAfter(delimiter) return if (index == -1) missingDelimiterValue else
substring(index + 1, length) }}

* Returns a substring after the last occurrence of $[delimiter]$.

*public fun String.substringAfter(delimiter: String, missingDelimiterValue: String = this) { String = this; String { val index = lastIndexOf(delimiter) return if (index == -1) missingDelimiterValue else
substring(index + delimiter.length, length) }}

* Returns a substring after the last occurrence of $[delimiter]$. If the string does not contain the delimiter, returns
$[missingDelimiterValue] which defaults to the original string.

*public fun String.substringBeforeLast(delimiter: Char, missingDelimiterValue: String = this) { String = this; String { val index = lastIndexOf(delimiter) return if (index == -1) missingDelimiterValue else
substring(0, index) }}

* Returns a substring before the last occurrence of $[delimiter]$. If the string does not contain the delimiter, returns
$[missingDelimiterValue] which defaults to the original string.

*public fun String.substringBeforeLast(delimiter: String, missingDelimiterValue: String = this) { String = this; String { val index = lastIndexOf(delimiter) return if (index == -1) missingDelimiterValue else
substring(0, index) }}

* Returns a substring before the last occurrence of $[delimiter]$. If the string does not contain the delimiter, returns
$[missingDelimiterValue] which defaults to the original string.

*public fun String.substringAfterLast(delimiter: Char, missingDelimiterValue: String = this) { String = this; String { val index = lastIndexOf(delimiter) return if (index == -1) missingDelimiterValue else
substring(index + 1, length) }}

* Returns a substring after the last occurrence of $[delimiter]$. If the string does not contain the delimiter, returns
$[missingDelimiterValue] which defaults to the original string.

*public fun String.substringAfterLast(delimiter: String, missingDelimiterValue: String = this) { String = this; String { val index = lastIndexOf(delimiter) return if (index == -1) missingDelimiterValue else
substring(index + delimiter.length, length) }}

* Returns a substring after the last occurrence of $[delimiter]$. If the string does not contain the delimiter, returns
$[missingDelimiterValue] which defaults to the original string.

*public fun CharSequence.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence): CharSequence {
    if (endIndex < startIndex)
        throw IndexOutOfBoundsException("End index ($endIndex) is less than start index ($startIndex).")
    val sb = StringBuilder()
    sb.append(this, 0, startIndex)
    sb.append(replacement)
    sb.append(this, endIndex, length)
    return sb
}

*public fun String.replaceRange(startIndex: Int, endIndex: Int, replacement: String): String {
    (this as CharSequence).replaceRange(startIndex, endIndex, replacement).toString()
}

*public fun CharSequence.replaceRange(range: IntRange, replacement: CharSequence): CharSequence =
    replaceRange(range.start, range.endInclusive + 1, replacement) 

*public fun String.replaceRange(range: IntRange, replacement: String): String {
    (this as CharSequence).replaceRange(range, replacement).toString()
}

*public fun CharSequence.replaceRange(range: IntRange, replacement: CharSequence): CharSequence =
    this.replace(range, replacement) 

*public fun String.replaceRange(range: IntRange, replacement: String): String {
    (this as CharSequence).replaceRange(range, replacement).toString()
}

*public fun String.replaceRange(range: IntRange, replacement: CharSequence): String {
    this.replace(range, replacement) 

*public fun String.replaceRange(range: IntRange, replacement: CharSequence): String {
    this.replace(range, replacement) 

*public fun String.replaceRange(range: IntRange, replacement: CharSequence): String {
    this.replace(range, replacement) 

*public fun String.replaceRange(range: IntRange, replacement: String): String {
    (this as CharSequence).replaceRange(range, replacement).toString()
}

*public fun String.replaceRange(range: IntRange, replacement: CharSequence): String {
    this.replace(range, replacement) 

*public fun String.replaceRange(range: IntRange, replacement: String): String {
    (this as CharSequence).replaceRange(range, replacement).toString()
}

*public fun String.replaceRange(range: IntRange, replacement: CharSequence): String {
    this.replace(range, replacement) 

*public fun String.replaceRange(range: IntRange, replacement: String): String {
    (this as CharSequence).replaceRange(range, replacement).toString()
}
given range is removed.

```kotlin
@public fun CharSequence.removeRange(startIndex: Int, endIndex: Int):
CharSequence {
    if (endIndex < startIndex)
        throw IndexOutOfBoundsException("End index ($endIndex) is less than start index ($startIndex).")

    if (endIndex == startIndex)
        return this.subSequence(0, length)
    val sb = StringBuilder(length - (endIndex - startIndex))
    sb.append(this, 0, startIndex)
    sb.append(this, endIndex, length)
    return sb.toString()
}
```

```kotlin
public fun String.removeRange(startIndex: Int, endIndex: Int): String =
    (this as CharSequence).removeRange(startIndex, endIndex).toString()
```

```kotlin
public fun CharSequence.removeRange(range: IntRange): CharSequence =
    removeRange(range.start, range.endInclusive + 1)
```

```kotlin
public fun String.removeRange(range: IntRange): String =
    removeRange(range.start, range.endInclusive + 1).toString()
```

```kotlin
public fun CharSequence.removePrefix(prefix: CharSequence): CharSequence {
    if (startsWith(prefix)) {
        return subSequence(prefix.length, length)
    }
    return subSequence(0, length)
}
```

```kotlin
public fun String.removePrefix(prefix: CharSequence): String {
    if (startsWith(prefix)) {
        return substring(prefix.length)
    }
    return this
}
```

```kotlin
public fun CharSequence.removeSuffix(suffix: CharSequence): CharSequence {
    if (endsWith(suffix)) {
        return subSequence(0, length - suffix.length)
    }
    return subSequence(0, length)
}
```

```kotlin
public fun String.removeSuffix(suffix: CharSequence): String {
    if (endsWith(suffix)) {
        return substring(0, length - suffix.length)
    }
    return this
}
```

```kotlin
public fun CharSequence.removeSurrounding(prefix: CharSequence, suffix: CharSequence): CharSequence {
    if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {
        return subSequence(prefix.length, length - suffix.length)
    }
    return subSequence(0, length)
}
```

```kotlin
public fun String.removeSurrounding(prefix: CharSequence, suffix: CharSequence): String {
    if ((length >= prefix.length + suffix.length) &&
        startsWith(prefix) &&
        endsWith(suffix)) {
        return substring(prefix.length, length - suffix.length)
    }
    return this
}
```

```kotlin
public fun CharSequence.removeSurrounding(delimiter: CharSequence): CharSequence {
    if (length >= delimiter.length + suffix.length) &&
        startsWith(delimiter) &&
        endsWith(delimiter) {
        return subSequence(delimiter.length, length - suffix.length)
    }
    return subSequence(0, length)
}
```

```kotlin
public fun String.removeSurrounding(delimiter: CharSequence): String {
    if (length >= delimiter.length + suffix.length) &&
        startsWith(delimiter) &&
        endsWith(delimiter) {
        return substring(delimiter.length, length - suffix.length)
    }
    return this
}
```

```kotlin
public fun String.replaceBefore(delimiter: Char,
replacement: String, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if
```
Invoked when it's already known that arguments are not Strings, so no additional type checks are performed.

```kotlin
internal fun CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean): Boolean =
    regionMatches(this, other, thisOffset, otherOffset, length, ignoreCase)
```

```kotlin
/**
 * Implementation of [regionMatches] for CharSequences.
 *
 * Invoked when it's already known that arguments are not Strings, so that no additional type checks are performed.
 */
internal fun CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean): Boolean {
    if (!this.isSurrogatePairAt(it - 1) || other.isSurrogatePairAt(it - 1)) {
        return false
    }

    var i = 0
    while (i < shortestLength && this[i].equals(other[i], ignoreCase)) {
        i++
    }
    if (this[i].hasSurrogatePairAt(i - 1) || other[i].hasSurrogatePairAt(i - 1)) {
        i--
    }

    return subSequence(i, i + length).toString()
}
```

```kotlin
/**
 * Returns true if this char sequence starts with the specified character.
 */
public fun CharSequence.startsWith(char: Char, ignoreCase: Boolean = false): Boolean =
    this.length > 0 &&
    this[0].equals(char, ignoreCase)
```

```kotlin
/**
 * Returns true if this char sequence starts with the specified prefix.
 */
public fun CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && prefix is String)
        return this.startsWith(prefix)
    else
        return regionMatchesImpl(0, prefix, 0, prefix.length, ignoreCase)
```

```kotlin
/**
 * Returns true if this char sequence ends with the specified suffix.
 */
public fun CharSequence.endsWith(suffix: CharSequence, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && suffix is String)
        return this.endsWith(suffix)
    else
        return regionMatchesImpl(length - suffix.length, 0, suffix.length, ignoreCase)
```

```kotlin
/**
 * Returns the longest string `prefix` such that this char sequence and [other] char sequence both start with this prefix,
 * taking care not to split surrogate pairs.
 * If this and [other] have no common prefix, returns the empty string.
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @sample samples.text.Strings.commonPrefixWith
 */
public fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {
    val shortestLength = minOf(this.length, other.length)
    var i = 0
    while (i < shortestLength && this[i].equals(other[i], ignoreCase)) {
        i++
    }
    if (this[i].hasSurrogatePairAt(i - 1) || other[i].hasSurrogatePairAt(i - 1)) {
        i--
    }
    return subSequence(0, i).toString()
}
```

```kotlin
/**
 * Returns the longest string `suffix` such that this char sequence and [other] char sequence both end with this suffix,
 * taking care not to split surrogate pairs.
 * If this and [other] have no common suffix, returns the empty string.
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @sample samples.text.Strings.commonSuffixWith
 */
public fun CharSequence.commonSuffixWith(other: CharSequence, ignoreCase: Boolean = false): String {
    val thisLength = this.length
    val otherLength = other.length
    val shortestLength = minOf(thisLength, otherLength)
    var i = 0
    while (i < shortestLength && this[thisLength - i - 1].equals(other[otherLength - i - 1], ignoreCase)) {
        i++
    }
    if (this[i].hasSurrogatePairAt(thisLength - i - 1) || other[i].hasSurrogatePairAt(otherLength - i - 1)) {
        i--
    }
    return subSequence(thisLength - i, thisLength).toString()
}
```

```kotlin
/**
 * Finds the index of the first occurrence of any of the specified [chars] in this char sequence,
 * starting from the specified [startIndex] and optionally ignoring the case.
 * @param ignoreCase `true` to ignore character case when matching a character.
 * @sample samples.text.Strings.indexOfAny
 */
public fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int = 0, ignoreCase: Boolean = false): Int {
    if (!ignoreCase && chars.size == 1 && this is String) {
        return nativeIndexOf(chars.single())
    }

    for (index in startIndex.coerceAtLeast(0)..<lastIndex) {
        val charAtIndex = get(index)
        if (chars.any { it.equals(charAtIndex, ignoreCase) })
            return index
    }

    return -1
}
```
ignoreCase 'true' to ignore character case when matching a character. By default 'false'

* @return An index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.

```kotlin
*public fun CharSequence.lastIndexOfAny(chars: CharArray, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {
    val chars = this.toCharArray()  // Convert string to char array
    @Suppress("unused")
    if (!ignoreCase && chars.size == 1) {
        return this.indexOfChar(chars[0], 0, ignoreCase)  // Compute index of single char
    }
    for (index in startIndex.coerceAtMost(lastIndex) downTo 0) {
        val matchingChar = chars.firstOrNull { it.regionMatches(index, this, 0, it.length, ignoreCase) }
        if (matchingChar != null) {
            return index to matchingChar
        }
    }
    return -1
}
```
backward toward the beginning of the string.

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".

*param ignoreCase "true" to ignore character case when matching a string. By default "false".
sendMessage(fileOutputStream, string);
var result = "" for (char in this) { result += char } return result

/**
 * Concatenates characters in this [CharArray] or its subrange into a String.
 * @param startIndex the beginning (inclusive) of the subrange of characters, 0 by default.
 * @param endIndex the end (exclusive) of the subrange of characters, size of this array by default.
 * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].
 */
@ExperimentalStdlibApi
public actual fun CharArray.concatToString(startIndex: Int = 0, endIndex: Int = this.size): String {
    AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)
    var result = ""
    for (index in startIndex until endIndex) {
        result += this[index]
    }
    return result
}

/**
 * Returns a [CharArray] containing characters of this string.
 */
@ExperimentalStdlibApi
public actual fun String.toCharArray(): CharArray {
    return CharArray(length) { get(it) }
}

/**
 * Returns a [CharArray] containing characters of this string or its substring.
 * @param startIndex the beginning (inclusive) of the substring, 0 by default.
 * @param endIndex the end (exclusive) of the substring, length of this string by default.
 * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the length of this string.
 * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].
 */
@ExperimentalStdlibApi
public actual fun String.toCharArray(startIndex: Int = 0, endIndex: Int = this.length): CharArray {
    AbstractList.checkBoundsIndexes(startIndex, endIndex, this.length)
    return CharArray(endIndex - startIndex) { get(startIndex + it) }
}

/**
 * Decodes a string from the bytes in UTF-8 encoding in this array.
 */
@ExperimentalStdlibApi
public actual fun ByteArray.decodeToString(): String {
    return decodeUtf8(this, 0, size, false)
}

/**
 * Decodes a string from the bytes in UTF-8 encoding in this array or its subrange.
 * @param startIndex the beginning (inclusive) of the subrange to decode, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to decode, size of this array by default.
 * @param throwOnInvalidSequence specifies whether to throw an exception on malformed byte sequence or replace it by the replacement char `\uFFFD`.
 * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].
 * @throws CharacterCodingException if this string contains malformed UTF-8 byte sequence and [throwOnInvalidSequence] is true.
 */
@ExperimentalStdlibApi
public actual fun ByteArray.decodeToString(
    startIndex: Int = 0,
    endIndex: Int = this.size,
    throwOnInvalidSequence: Boolean = false
): String {
    AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)
    return decodeUtf8(this, startIndex, endIndex, throwOnInvalidSequence)
}

/**
 * Encodes this string to an array of bytes in UTF-8 encoding.
 */
@ExperimentalStdlibApi
public actual fun String.encodeToByteArray(): ByteArray {
    return encodeUtf8(this, 0, length, false)
}

/**
 * Encodes this string or its substring to an array of bytes in UTF-8 encoding.
 * @param startIndex the beginning (inclusive) of the substring to encode, 0 by default.
 * @param endIndex the end (exclusive) of the substring to encode, length of this string by default.
 * @param throwOnInvalidSequence specifies whether to throw an exception on malformed char sequence or replace it by the replacement byte sequence `\uFFFD`.
 * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.
 * @throws CharacterCodingException if this string contains malformed UTF-8 byte sequence and [throwOnInvalidSequence] is true.
 */
@ExperimentalStdlibApi
public actual fun String.encodeToByteArray(
    startIndex: Int = 0,
    endIndex: Int = this.length,
    throwOnInvalidSequence: Boolean = false
): ByteArray {
    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)
    return encodeUtf8(this, startIndex, endIndex, throwOnInvalidSequence)
}
throwOnInvalidSequence)

/**
 * Returns a copy of this string converted to upper case using the rules of the
default locale.
 *
 * @sample samples.text.Strings.toUpperCase
 */
@kotlin.internal.InlineOnly
public actual inline fun String.toUpperCase(): String = asDynamic().toUpperCase()

/**
 * Returns a copy of this string
converted to lower case using the rules of the default locale.
 *
 * @sample samples.text.Strings.toLowerCase
 */
@kotlin.internal.InlineOnly
public actual inline fun String.toLowerCase(): String = asDynamic().toLowerCase()

@kotlin.internal.InlineOnly
internal actual inline fun String.nativeIndexOf(str: String, fromIndex: Int): Int = asDynamic().indexOf(str, fromIndex)

@kotlin.internal.InlineOnly
internal actual inline fun String.nativeLastIndexOf(str: String, fromIndex: Int): Int = asDynamic().lastIndexOf(str, fromIndex)

@kotlin.internal.InlineOnly
internal inline fun String.nativeStartsWith(s: String, position: Int): Boolean = asDynamic().startsWith(s, position)

@kotlin.internal.InlineOnly
internal inline fun String.nativeEndsWith(s: String): Boolean = asDynamic().endsWith(s)

@kotlin.internal.InlineOnly
public actual inline fun String.substring(startIndex: Int): String = asDynamic().substring(startIndex)

@kotlin.internal.InlineOnly
public actual inline fun String.substring(startIndex: Int, endIndex: Int): String = asDynamic().substring(startIndex, endIndex)

@kotlin.internal.InlineOnly
public inline fun String.concat(str: String): String = asDynamic().concat(str)

@kotlin.internal.InlineOnly
public inline fun String.match(regex: String): Array<String>? = asDynamic().match(regex)

@SinceKotlin("1.2")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.compareTo(other: String, ignoreCase: Boolean = false): Int {
    if (ignoreCase) {
        val n1 = this.length
        val n2 = other.length
        val min = minOf(n1, n2)
        if (min == 0) return n1 - n2
        var start = 0
        while (true) {
            val end = minOf(start + 16, min)
            var s1 = this.substring(start, end)
            var s2 = other.substring(start, end)
            if (s1 != s2) {
                s1 = s1.toUpperCase()
                s2 = s2.toUpperCase()
                if (s1 != s2) {
                    s1 = s1.toLowerCase()
                    s2 = s2.toLowerCase()
                    if (s1 != s2) {
                        return s1.compareTo(s2)
                    }
                }
            }
            if (end == min) break
            start = end
        }
        return n1 - n2
    } else {
        return compareTo(other)
    }
}

private val STRING_CASE_INSENSITIVE_ORDER = Comparator<String> { a, b -> a.compareTo(b, ignoreCase = true) }

@.SinceKotlin("1.2")
@ExperimentalUnsignedTypes
public inline class UByteArray
@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@PublishedApi
internal constructor(@PublishedApi internal val storage: ByteArray) : Collection<UByte> {
    /** Creates
     * a new array of the specified 
     * [size], with all elements initialized to zero. */
    public constructor(size: Int) : this(ByteArray(size))

    /**
     * Returns the array element at the given 
     * [index]. This method can be called
     * using the index operator.[n]
     */
    public operator fun get(index: Int): UByte = storage[index].toUByte()

    /**
     * Sets the element at the given 
     * [index] to the given 
     * [value]. This method can be called using the index operator.[n]
     */
    public operator fun set(index: Int, value: UByte) {
        storage[index] = value.toByte()
    }

    /** Returns the number of elements in the array. */
    public override val size: Int get() = storage.size

    /** Creates an iterator over the elements of the array. */
    public override operator fun iterator(): UByteIterator = Iterator(storage)

    private class Iterator(private val array: ByteArray) : UByteIterator() {
        private var index = 0
        override fun hasNext() = index < array.size
        override fun nextUByte() = if (index < array.size)
array[index++].toUByte() else throw NoSuchElementException(index.toString())
}]

override fun contains(element: UByte): Boolean {
    return storage.contains(element.toByte())
}

override fun containsAll(elements: Collection<UByte>): Boolean {
    return (elements as Collection<*>).all { it is UByte && storage.contains(it.toByte()) }
}

override fun isEmpty(): Boolean = this.storage.size == 0
}

/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly
public inline fun UByteArray(size: Int, init: (Int) -> UByte): UByteArray {
    return UByteArray(ByteArray(size) { index ->
      init(index).toByte()
    })
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly
public inline fun ubyteArrayOf(vararg elements: UByte): UByteArray = elements

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that
 * can be found in the license/LICENSE.txt file.
 */

// Auto-generated file. DO NOT EDIT!

package kotlin

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UIntArray
@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@PublishedApi
internal constructor(@PublishedApi internal val storage: IntArray) : Collection<UInt> {
    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    constructor(size: Int) : this(IntArray(size))

    override operator fun get(index: Int): UInt = storage[index].toUInt()

    override operator fun set(index: Int, value: UInt) {
        storage[index] = value.toInt()
    }

    override val size: Int get() = storage.size

    override operator fun iterator(): UIntIterator = Iterator(storage)

    private class Iterator(private val array: IntArray) : UIntIterator() {
        private var index = 0
        override fun hasNext() = index < array.size
        override fun nextUInt() = if (index < array.size)
          array[index++].toUInt() else throw NoSuchElementException(index.toString())
    }

    override fun contains(element: UInt): Boolean {
        return storage.contains(element.toInt())
    }

    override fun containsAll(elements: Collection<UInt>): Boolean {
        return (elements as Collection<*>).all { it is UInt &&
          storage.contains(it.toInt()) }
    }

    override fun isEmpty(): Boolean = this.storage.size == 0
}

/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
      init(index).toInt()
    })
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly
public inline fun uintArrayOf(vararg elements: UInt): UIntArray = elements
private constructor(val storage: LongArray) : Collection<ULong> {

    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    constructor(size: Int) : this(LongArray(size)) /* * Returns the array element at the given [index]. This method can be called using the index operator. */

    operator fun get(index: Int): ULong = storage[index].toULong() /* * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS. */

    operator fun set(index: Int, value: ULong) {
        storage[index] = value.toLong()
    }

    /** Returns the number of elements in the array. */
    override val size: Int get() = storage.size

    /** Creates an iterator over the elements of the array. */
    override operator fun iterator(): ULongIterator = Iterator(storage)

    private class Iterator(private val array: LongArray) : ULongIterator() {
        private var index = 0
        override fun hasNext() = index < array.size
        override fun nextULong() = if (index < array.size) array[index++].toULong() else throw NoSuchElementException(index.toString())
    }

    override fun contains(element: ULong): Boolean {
        // TODO: Eliminate this check after KT-30016 gets fixed. Currently JS BE does not generate special bridge method for this method. If (element as Any?) !is ULong return false
        return storage.contains(element.toLong())
    }

    override fun containsAll(elements: Collection<ULong>): Boolean {
        return (elements as Collection<*>).all { it is ULong && storage.contains(it.toLong()) }
    }

    override fun isEmpty(): Boolean = this.storage.size == 0
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray(size: Int, init: (Int) -> ULong): ULongArray {
    return ULongArray(LongArray(size) { index -> init(index).toLong() })
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ulongArrayOf(vararg elements: ULong): ULongArray = elements

positive number if it's greater than other. */

@kotlin.internal.InlineOnly
public inline operator fun compareTo(other: UInt): Int = this.compareTo(other.toULong())
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UByte): ULong = this.plus(other.toULong())
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UShort): ULong = this.plus(other.toULong())
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UInt): ULong = this.plus(other.toULong())
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: ULong): ULong = ULong(this.data.plus(other.data))
/** Subtracts the other value from this value. */
@kotlin.internal.InlineOnly
public inline operator fun minus(other: UByte): ULong = this.minus(other.toULong())
/** Subtracts the other value from this value. */
@kotlin.internal.InlineOnly
public inline operator fun minus(other: UShort): ULong = this.minus(other.toULong())
/** Subtracts the other value from this value. */
@kotlin.internal.InlineOnly
public inline operator fun minus(other: UInt): ULong = this.minus(other.toULong())
/** Subtracts the other value from this value. */
@kotlin.internal.InlineOnly
public inline operator fun minus(other: ULong): ULong = ULong(this.data.minus(other.data))
/** Multiplies this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: UByte): ULong = this.times(other.toULong())
/** Multiplies this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: UShort): ULong = this.times(other.toULong())
/** Multiplies this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: UInt): ULong = this.times(other.toULong())
/** Multiplies this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))
/** Divides this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): ULong = this.div(other.toULong())
/** Divides this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): ULong = this.div(other.toULong())
/** Divides this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): ULong = this.div(other.toULong())
/** Divides this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): ULong = ulongDivide(this, other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UByte): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = ulongRemainder(this, other)
/** Increments this value. */
@kotlin.internal.InlineOnly
public inline operator fun inc(): ULong = ULong(data.inc())
/** Decrements this value. */
@kotlin.internal.InlineOnly
public inline operator fun dec(): ULong = ULong(data.dec())
/** Creates a range from this value to the specified [other] value. */
@kotlin.internal.InlineOnly
public inline operator fun rangeTo(other: ULong): ULongRange = ULongRange(this, other)
/** Shifts this value left by the [bitCount] number of bits. */
@kotlin.internal.InlineOnly
public inline operator fun shl(bitCount: Int): ULong = ULong(data shl bitCount)
/** Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros. */
@kotlin.internal.InlineOnly
public inline operator fun shr(bitCount: Int): ULong = ULong(data ushr bitCount)
/** Performs a bitwise AND operation between the two values. */
@kotlin.internal.InlineOnly
public inline operator fun and(other: ULong): ULong = ULong(this.data and other.data)
/** Performs a bitwise OR operation between the two values. */
@kotlin.internal.InlineOnly
public inline operator fun or(other: ULong): ULong = ULong(this.data or other.data)
public inline infix fun or(other: ULong): ULong = ULong(this.data or other.data)

/** Performs a bitwise XOR operation between the two values. */
@kotlin.internal.InlineOnly
public inline infix fun xor(other: ULong): ULong = ULong(this.data xor other.data)

/** Inverts the bits in this value. */
@kotlin.internal.InlineOnly
public inline fun inv(): ULong = ULong(data.inv())

/** Converts this ULong value to [Byte]. */
@kotlin.internal.InlineOnly
public inline fun toByte(): Byte = data.toByte()

/** Converts this [ULong] value to [Short]. */
@kotlin.internal.InlineOnly
public inline fun toShort(): Short = data.toShort()

/** Converts this [ULong] value to [Int]. */
@kotlin.internal.InlineOnly
public inline fun toInt(): Int = data.toInt()

/** Converts this [ULong] value to [Long]. */
@kotlin.internal.InlineOnly
public inline fun toLong(): Long = data.toLong()

/** Converts this [ULong] value to [Float]. */
@kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toDouble().toFloat()

/** Converts this [ULong] value to [Double]. */
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double = ulongToDouble(data)

/** Returns this value. */
@kotlin.internal.InlineOnly
public inline fun toUByte(): UByte = data.toUByte()

/** Converts this [ULong] value to [UByte]. */
@kotlin.internal.InlineOnly
public inline fun toUByte(): UByte = data.toUByte()

/** Converts this [ULong] value to [UShort]. */
@kotlin.internal.InlineOnly
public inline fun toUShort(): UShort = data.toUShort()

/** Converts this [ULong] value to [UInt]. */
@kotlin.internal.InlineOnly
public inline fun toUInt(): UInt = data.toUInt()

/** Converts this [ULong] value to [Long]. */
@kotlin.internal.InlineOnly
public inline fun toULong(): ULong = this

/** Converts this [ULong] value to [Int]. */
@kotlin.internal.InlineOnly
public inline fun toInt(): Int = data.toInt()

/** Converts this [ULong] value to [Long]. */
@kotlin.internal.InlineOnly
public inline fun toLong(): Long = data.toLong()

/** Converts this [ULong] value to [Float]. */
@kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toDouble().toFloat()

/** Converts this [ULong] value to [Double]. */
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double = ulongToDouble(data)

/** Converts this [ULong] value to [Short]. */
@kotlin.internal.InlineOnly
public inline fun toShort(): Short = data.toShort()

/** Converts this [ULong] value to [Int]. */
@kotlin.internal.InlineOnly
public inline fun toInt(): Int = data.toInt()

/** Converts this [ULong] value to [Long]. */
@kotlin.internal.InlineOnly
public inline fun toLong(): Long = data.toLong()

/** Converts this [ULong] value to [Float]. */
@kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toDouble().toFloat()

/** Converts this [ULong] value to [Double]. */
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double = ulongToDouble(data)

/** Converts this [ULong] value to [Short]. */
@kotlin.internal.InlineOnly
public inline fun toShort(): Short = data.toShort()

/** Converts this [ULong] value to [Int]. */
@kotlin.internal.InlineOnly
public inline fun toInt(): Int = data.toInt()

/** Converts this [ULong] value to [Long]. */
@kotlin.internal.InlineOnly
public inline fun toLong(): Long = data.toLong()

/** Converts this [ULong] value to [Float]. */
@kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toDouble().toFloat()

/** Converts this [ULong] value to [Double]. */
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double = ulongToDouble(data)

/** Converts this [ULong] value to [Short]. */
@kotlin.internal.InlineOnly
public inline fun toShort(): Short = data.toShort()

/** Converts this [ULong] value to [Int]. */
@kotlin.internal.InlineOnly
public inline fun toInt(): Int = data.toInt()

/** Converts this [ULong] value to [Long]. */
@kotlin.internal.InlineOnly
public inline fun toLong(): Long = data.toLong()
Short.toULong(): ULong = ULong(this.toLong())

* Converts this [Int] value to [ULong].
* If this value is positive, the resulting `ULong` value represents the same numerical value as this `Int`.
* The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `Int` value, whereas the most significant 32 bits are filled with the sign bit of this value.

* @SinceKotlin("1.3")
* @ExperimentalUnsignedTypes
* @kotlin.internal.InlineOnly
public inline fun Int.toULong(): ULong

* Converts this [Long] value to [ULong].
* If this value is positive, the resulting `ULong` value represents the same numerical value as this `Long`.
* The resulting `ULong` value has the same binary representation as this `Long` value.

* @SinceKotlin("1.3")
* @ExperimentalUnsignedTypes
* @kotlin.internal.InlineOnly
public inline fun Long.toULong(): ULong

* @SinceKotlin("1.3")
* @ExperimentalUnsignedTypes
* @kotlin.internal.InlineOnly
public inline fun Float.toULong(): ULong

* The fractional part, if any, is rounded down towards zero. Returns zero if this `Float` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`.

* @SinceKotlin("1.3")
* @ExperimentalUnsignedTypes
* @kotlin.internal.InlineOnly
public inline class UShortArray

* @PublishedApi
* internal constructor(@PublishedApi internal val storage: ShortArray) : Collection<UShort> {

* Creates a new array of the specified [size], with all elements initialized to zero.
* 
* Sets the element at the given [index] to the given [value]. This method can be called using the index operator.

* Returns the number of elements in the array.

* Returns an iterator over the elements of the array.

* Returns whether the given value is contained in the array.

* Creates a new array of the specified [size], where each element is calculated by calling the specified [init] function.

* The function [init] is called for each array element sequentially starting from the first one.

* It should return the value for an array element given its index.

* @SinceKotlin("1.3")
* @ExperimentalUnsignedTypes
* @kotlin.internal.InlineOnly
public inline fun UShortArray(size: Int, init: (Int) -> UShort): UShortArray {
**Open Source Used In DNAC Platform Antelope**

Antelope

ushortArrayOf(vararg elements: UShort): UShortArray = elements

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("UArraysKt")
@package:kotlin.collections.unsigned

package kotlin.collections

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns 1st *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UIntArray.component1(): UInt {
    return get(0)
}

/**
 * Returns 1st *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun ULongArray.component1(): ULong {
    return get(0)
}

/**
 * Returns 1st *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UByteArray.component1(): UByte {
    return get(0)
}

/**
 * Returns 1st *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UShortArray.component1(): UShort {
    return get(0)
}

/**
 * Returns 2nd *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UIntArray.component2(): UInt {
    return get(1)
}

/**
 * Returns 2nd *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun ULongArray.component2(): ULong {
    return get(1)
}

/**
 * Returns 2nd *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UByteArray.component2(): UByte {
    return get(1)
}

/**
 * Returns 2nd *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UShortArray.component2(): UShort {
    return get(1)
}

/**
 * Returns 3rd *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UIntArray.component3(): UInt {
    return get(2)
}

/**
 * Returns 3rd *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun ULongArray.component3(): ULong {
    return get(2)
}

/**
 * Returns 3rd *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UByteArray.component3(): UByte {
    return get(2)
}

/**
 * Returns 3rd *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UShortArray.component3(): UShort {
    return get(2)
}

/**
 * Returns 4th *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UIntArray.component4(): UInt {
    return get(3)
}

/**
 * Returns 4th *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun ULongArray.component4(): ULong {
    return get(3)
}

/**
 * Returns 4th *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UByteArray.component4(): UByte {
    return get(3)
}

/**
 * Returns 4th *element* from the collection.
 */
@SinceKotlin("1.3")
public inline operator fun UShortArray.component4(): UShort {
    return get(3)
}

/**
 * Returns an element at the given [index] or
throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.

* * *

`@sample samples.collections.Collections.Elements.elementAt`

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UIntArray.elementAt(index: Int): UInt
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun ULongArray.elementAt(index: Int): ULong
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UByteArray.elementAt(index: Int): UByte
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UShortArray.elementAt(index: Int): UShort
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UIntArray.elementAtOrNull(index: Int): UInt? {
    return this.getOrNull(index)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun ULongArray.elementAtOrNull(index: Int): ULong? {
    return this.getOrNull(index)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UByteArray.elementAtOrNull(index: Int): UByte? {
    return this.getOrNull(index)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UShortArray.elementAtOrNull(index: Int): UShort? {
    return this.getOrNull(index)
}
```
UShortArray.elementAtOrNull(index: Int): UShort? {
    return this.getOrNull(index)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.find(predicate: (UInt) -> Boolean): UInt? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.find(predicate: (ULong) -> Boolean): ULong? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.find(predicate: (UByte) -> Boolean): UByte? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.find(predicate: (UShort) -> Boolean): UShort? {
    return firstOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.findLast(predicate: (UInt) -> Boolean): UInt? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.findLast(predicate: (ULong) -> Boolean): ULong? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.findLast(predicate: (UByte) -> Boolean): UByte? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {
    return lastOrNull(predicate)
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.first(): UInt {
    return storage.first().toUInt()
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.first(): ULong {
    return storage.first().toULong()
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.first(): UByte {
    return storage.first().toUByte()
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.first(): UShort {
    return storage.first().toUShort()
}

/**
 * Returns the first element matching the given [predicate],
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.first(predicate: (UInt) -> Boolean): UInt {
    for (element in this) if (predicate(element)) return
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate],
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.first(predicate: (ULong) -> Boolean): ULong {
    for (element in this) if (predicate(element)) return
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate],
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.first(predicate: (UByte) -> Boolean): UByte {
    for (element in this) if (predicate(element)) return
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate],
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.first(predicate: (UShort) -> Boolean): UShort {
    for (element in this) if (predicate(element)) return
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.first(predicate: (UByte) -> Boolean): UByte {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.first(predicate: (UShort) -> Boolean): UShort {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UIntArray.firstOrNull(): UInt? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun ULongArray.firstOrNull(): ULong? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.getOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.getOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.getOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.getOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
public fun UIntArray.getOrNull(index: Int): UInt? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.getOrNull(index: Int): ULong? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.getOrNull(index: Int): UByte? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.getOrNull(index: Int): UShort? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}

/**
 * Returns the first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.indexOf(element: UInt): Int {
    return storage.indexOf(element.toInt())
}

/**
 * Returns the first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.indexOf(element: ULong): Int {
    return storage.indexOf(element.toLong())
}

/**
 * Returns the first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.indexOf(element: UByte): Int {
    return storage.indexOf(element.toByte())
}

/**
 * Returns the first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.indexOf(element: UShort): Int {
    return storage.indexOf(element.toShort())
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUInt()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toULong()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUByte()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.indexOfFirst(predicate: (UShort) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUShort()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUInt()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toULong()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUByte()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUShort()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUInt()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toULong()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUByte()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUShort()) }
}
predicate(it.toUByte())

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {
    @throws [NoSuchElementException] if the array is empty.
    return storage.indexOfLast {
        predicate(it.toUShort())
    }
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.last(): UInt {
    return storage.last().toUInt()
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.last(): ULong {
    return storage.last().toULong()
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.last(): UByte {
    return storage.last().toUByte()
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.last(): UShort {
    return storage.last().toUShort()
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.lastIndexOf(element: UInt): Int {
    return storage.lastIndexOf(element.toInt())
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.lastIndexOf(element: ULong): Int {
    return storage.lastIndexOf(element.toLong())
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.lastIndexOf(element: UByte): Int {
    return storage.lastIndexOf(element.toByte())
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.lastIndexOf(element: UShort): Int {
    return storage.lastIndexOf(element.toShort())
}
Returns the last element, or `null` if the array is empty.

```kotlin
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UIntArray.lastOrNull(): UInt? {
    return if (isEmpty()) null else this[size - 1]
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun ULongArray.lastOrNull(): ULong? {
    return if (isEmpty()) null else this[size - 1]
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UByteArray.lastOrNull(): UByte? {
    return if (isEmpty()) null else this[size - 1]
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UShortArray.lastOrNull(): UShort? {
    return if (isEmpty()) null else this[size - 1]
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UIntArray.lastOrNull(predicate: (UInt) -> Boolean): UInt? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun ULongArray.lastOrNull(predicate: (ULong) -> Boolean): ULong? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UByteArray.lastOrNull(predicate: (UByte) -> Boolean): UByte? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UShortArray.lastOrNull(predicate: (UShort) -> Boolean): UShort? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UIntArray.random(): UInt {
    return random(Random)
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun ULongArray.random(): ULong {
    return random(Random)
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UByteArray.random(): UByte {
    return random(Random)
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UShortArray.random(): UShort {
    return random(Random)
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UIntArray.random(random: Random): UInt {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun ULongArray.random(random: Random): ULong {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UByteArray.random(random: Random): UByte {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public fun UShortArray.random(random: Random): UShort {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
get(random.nextInt(size)) * Returns a random element from this array using the specified source of randomness.

*Throws NoSuchElementException if this array is empty.

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@public fun UShortArray.random(random: Random): UShort {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size)) * Returns the single element, or throws an exception if the array is empty or has more than one element.

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun UIntArray.single(): UInt {
    return storage.single().toUInt() * Returns the single element, or throws an exception if the array is empty or has more than one element.

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun UShortArray.single(): UShort {
    return storage.single().toUShort() * Returns the single element, or throws an exception if the array is empty or has more than one element.

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun ULongArray.single(): ULong {
    return storage.single().toULong() * Returns the single element, or throws an exception if the array is empty or has more than one element.

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun UByteArray.single(): UByte {
    return storage.single().toUByte() * Returns the single element, or throws an exception if the array is empty or has more than one element.

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun UShortArray.single(): UShort {
    return storage.single().toUShort() * Returns the single element, or throws an exception if the array is empty or has more than one element.

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun UIntArray.single(predicate: (UInt) -> Boolean): UInt {
    var single: UInt? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as UInt
}

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun ULongArray.single(predicate: (ULong) -> Boolean): ULong {
    var single: ULong? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as ULong
}

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun UByteArray.single(predicate: (UByte) -> Boolean): UByte {
    var single: UByte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as UByte
}

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@inline fun UShortArray.single(predicate: (UShort) -> Boolean): UShort {
    var single: UShort? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as UShort
}

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@public fun UIntArray.singleOrNull(): UInt? {
    return if (size == 1) this[0] else null
}

*Úan@SinceKotlin("1.3")@ExperimentalUnsignedTypes@public fun UShortArray.randomOrNull(): UShort? {
    return if (size == 1) this[0] else null
}
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.singleOrNull(): ULong? {\n    return if (size == 1) this[0] else null\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.singleOrNull(): UByte? {\n    return if (size == 1) this[0] else null\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.singleOrNull(): UShort? {\n    return if (size == 1) this[0] else null\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {\n    var single: UInt? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {\n    var single: ULong? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {\n    var single: UByte? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.singleOrNull(predicate: (UShort) -> Boolean): UShort? {\n    var single: UShort? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.drop(n: Int): List<UInt> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.drop(n: Int): List<ULong> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.drop(n: Int): List<UByte> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.drop(n: Int): List<UShort> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.singleOrNull(): UInt? {
    var single: UInt? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
}

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.singleOrNull: ULong? {
    var single: ULong? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
}

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.singleOrNull: UByte? {
    var single: UByte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
}

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.singleOrNull: UShort? {
    var single: UShort? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
}

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.drop(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.drop(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.drop(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.drop(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count $n is less than zero." }\n    return takeLast((size - n).coerceAtLeast(0))\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.singleOrNull: UInt? {
    var single: UInt? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
}

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.singleOrNull: ULong? {
    var single: ULong? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
}

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.singleOrNull: UByte? {
    var single: UByte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
}

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.singleOrNull: UShort? {
    var single: UShort? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
}
samples.collections.Collections.Transformations.drop

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.dropLast(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.dropLast(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.dropLast(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.dropLast(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> {
    var yielding = false
    val list = ArrayList<UInt>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.dropWhile(predicate: (ULong) -> Boolean): List<ULong> {
    var yielding = false
    val list = ArrayList<ULong>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {
    var yielding = false
    val list = ArrayList<UByte>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {
    var yielding = false
    val list = ArrayList<UShort>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}
ArrayList<ULong>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item))
            list.add(item)
            yielding = true
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {
    var yielding = false
    val list = ArrayList<UByte>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item))
            list.add(item)
            yielding = true
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {
    var yielding = false
    val list = ArrayList<UShort>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item))
            list.add(item)
            yielding = true
    return list
}

/**
 * Returns a list containing only elements matching the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.filter(predicate: (UInt) -> Boolean): List<UInt> {
    return filterTo(ArrayList<UInt>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {
    return filterTo(ArrayList<ULong>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {
    return filterTo(ArrayList<UByte>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {
    return filterTo(ArrayList<UShort>(), predicate)
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.filterIndexed(predicate: (index: Int, UByte) -> Boolean): List<UByte> {
    return filterIndexedTo(ArrayList<UByte>(), predicate)
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.filterIndexed(predicate: (index: Int, UShort) -> Boolean): List<UShort> {
    return filterIndexedTo(ArrayList<UShort>(), predicate)
}
itself * and returns the result of predicate evaluation on the element.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun <C : 
MutableCollection<in UInt>> UIntArray.filterIndexedTo(destination: C, predicate: (index: Int, UInt) - Boolean): C {
    \n    forEachIndexed { index, element -\n    if (predicate(index, element)) destination.add(element)\n    }\n\n    return destination\n}* @\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun <C : 
MutableCollection<in ULong>> ULongArray.filterIndexedTo(destination: C, predicate: (index: Int, ULong) - Boolean): C {
    \n    forEachIndexed { index, element -\n    if (predicate(index, element)) destination.add(element)\n    }\n\n    return destination\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun <C : 
MutableCollection<in UByte>> UByteArray.filterIndexedTo(destination: C, predicate: (index: Int, UByte) - Boolean): C {
    \n    forEachIndexed { index, element -\n    if (predicate(index, element)) destination.add(element)\n    }\n\n    return destination\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun <C : 
MutableCollection<in UShort>> UShortArray.filterIndexedTo(destination: C, predicate: (index: Int, UShort) - Boolean): C {
    \n    forEachIndexed { index, element -\n    if (predicate(index, element)) destination.add(element)\n    }\n\n    return destination\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun <C : 
MutableCollection<in UInt>> UIntArray.filterNotTo(destination: C, predicate: (UInt) - Boolean): C {
    \n    for (element in this) if (!predicate(element)) destination.add(element)\n    }\n\n    return destination\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun <C : 
MutableCollection<in ULong>> ULongArray.filterNotTo(destination: C, predicate: (ULong) - Boolean): C {
    \n    for (element in this) if (!predicate(element)) destination.add(element)\n    }\n\n    return destination\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun <C : 
MutableCollection<in UByte>> UByteArray.filterNotTo(destination: C, predicate: (UByte) - Boolean): C {
    \n    for (element in this) if (!predicate(element)) destination.add(element)\n    }\n\n    return destination\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun <C : 
MutableCollection<in UShort>> UShortArray.filterNotTo(destination: C, predicate: (UShort) - Boolean): C {
    \n    for (element in this) if (!predicate(element)) destination.add(element)\n    }\n\n    return destination\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun UIntArray.filterNot(predicate: (UInt) - Boolean): List<UInt> {
    \n    return filterNotTo(ArrayList<UInt>(), predicate)\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun ULongArray.filterNot(predicate: (ULong) - Boolean): List<ULong> {
    \n    return filterNotTo(ArrayList<ULong>(), predicate)\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun UByteArray.filterNot(predicate: (UByte) - Boolean): List<UByte> {
    \n    return filterNotTo(ArrayList<UByte>(), predicate)\n}* @

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\public inline fun UShortArray.filterNot(predicate: (UShort) - Boolean): List<UShort> {
    \n    return filterNotTo(ArrayList<UShort>(), predicate)\n}* @

Open Source Used In DNAC Platform Antelope 3950
elements not matching the given [predicate] to the given [destination].

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UShort>> UShortArray.filterNotTo(destination: C, predicate: (UShort) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UInt>> UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in ULong>> ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.slice(indices: IntRange): List<UInt> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.slice(indices: IntRange): List<ULong> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.slice(indices: IntRange): List<UByte> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.slice(indices: IntRange): List<UShort> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}
```

*Open Source Used In DNAC Platform Antelope 3951*
Returns an array containing elements of this array at specified [indices].

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: Collection<Int>): UIntArray {
    return UIntArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: Collection<Int>): ULongArray {
    return ULongArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: Collection<Int>): UByteArray {
    return UByteArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: Collection<Int>): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: IntRange): UIntArray {
    return UIntArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: IntRange): ULongArray {
    return ULongArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: IntRange): UByteArray {
    return UByteArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: IntRange): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.take(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UInt>(n)
    for (item in this) {
        list.add(item)
        if (++count == n)
            break
    }
    return list
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.take(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<ULong>(n)
    for (item in this) {
        list.add(item)
        if (++count == n)
            break
    }
    return list
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.take(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UByte>(n)
    for (item in this) {
        list.add(item)
        if (++count == n)
            break
    }
    return list
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.take(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UShort>(n)
    for (item in this) {
        list.add(item)
        if (++count == n)
            break
    }
    return list
}
```

Open Source Used in DNAC Platform Antelope 3962
a list containing last \([n]\) elements.\n * \n * @throws IllegalArgumentException if \([n]\) is negative.\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.takeLast(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count \(n\) is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<UInt>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last \([n]\) elements.\n * \n * @throws IllegalArgumentException if \([n]\) is negative.\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.takeLast(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count \(n\) is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<ULong>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last \([n]\) elements.\n * \n * @throws IllegalArgumentException if \([n]\) is negative.\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.takeLast(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count \(n\) is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<UByte>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last \([n]\) elements.\n * \n * @throws IllegalArgumentException if \([n]\) is negative.\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.takeLast(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count \(n\) is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<UShort>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last elements satisfying the given \([predicate]\).\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given \([predicate]\).\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given \([predicate]\).\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given \([predicate]\).\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing first elements satisfying the given \([predicate]\).\n * \n * @sample samples.collections.Collections.Transformations.take

 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.takeWhile(predicate: (UInt) -> Boolean): List<UInt> {
    val list = ArrayList<UInt>()
    for (item in
this) { if (!predicate(item)) break list.add(item) return list }/**
 * Returns a list containing first elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.takeWhile(predicate: (ULong) -> Boolean): List<ULong> {
    val list = ArrayList<ULong>()
    for (item in this) {
        if (!predicate(item)) break list.add(item) return list
    }
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.takeWhile(predicate: (UByte) -> Boolean): List<UByte> {
    val list = ArrayList<UByte>()
    for (item in this) {
        if (!predicate(item)) break list.add(item) return list
    }
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {
    val list = ArrayList<UShort>()
    for (item in this) {
        if (!predicate(item)) break list.add(item) return list
    }
}

/**
 * Reverses elements in the array in-place.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reverse(): Unit {
    storage.reverse()
}

/**
 * Reverses elements in the array in-place.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reverse(): Unit {
    storage.reverse()
}

/**
 * Reverses elements in the array in-place.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reverse(): Unit {
    storage.reverse()
}

/**
 * Reverses elements in the array in-place.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reverse(): Unit {
    storage.reverse()
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.reversed(): List<UInt> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.reversed(): List<ULong> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.reversed(): List<UByte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.reversed(): List<UShort> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reversedArray(): UIntArray {
    return UIntArray(storage.reversedArray())
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reversedArray(): ULongArray {
    return ULongArray(storage.reversedArray())
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reversedArray(): UByteArray {
    return UByteArray(storage.reversedArray())
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reversedArray(): UShortArray {
    return UShortArray(storage.reversedArray())
}

/**
 * Sorts elements in the array in-place descending according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sortDescending(): Unit {
    if
(size > 1) { 
    sort() 
    reverse() 
} * * * * * * *{ 
Sorts elements in the array in-place descending according to their natural sort order.
}

public fun ULongArray.sortDescending(): Unit

if (size > 1) { 
    sort() 
    reverse() 
} * * * * * * *{ 
Sorts elements in the array in-place descending according to their natural sort order.
}

public fun UByteArray.sortDescending(): Unit

if (size > 1) { 
    sort() 
    reverse() 
} * * * * * * *{ 
Sorts elements in the array in-place descending according to their natural sort order.
}

public fun UShortArray.sortDescending(): Unit

if (size > 1) { 
    sort() 
    reverse() 
} * * * * * * *{ 
Returns a list of all elements sorted according to their natural sort order.
}

/**
 * Sorts elements in the array in-place descending according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sortDescending(): List<ULong>

{ 
    return copyOf().apply { sort() } 
}.asList() * * * * * * *{ 
Returns a list of all elements sorted according to their natural sort order.
}

/**
 * Sorts elements in the array in-place descending according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sortDescending(): List<UByte>

{ 
    return copyOf().apply { sort() } 
}.asList() * * * * * * *{ 
Returns a list of all elements sorted according to their natural sort order.
}

/**
 * Sorts elements in the array in-place descending according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sortDescending(): List<UShort>

{ 
    return copyOf().apply { sort() } 
}.asList() * * * * * * *{ 
Returns a list of all elements sorted according to their natural sort order.
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sortedDescending(): List<UInt>

{ 
    return copyOf().apply { sort() } 
}.reversed() * * * * * * *{ 
Returns a list of all elements sorted according to their natural sort order.
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sortedDescending(): List<ULong>

{ 
    return copyOf().apply { sort() } 
}.reversed() * * * * * * *{ 
Returns a list of all elements sorted according to their natural sort order.
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sortedDescending(): List<UByte>

{ 
    return copyOf().apply { sort() } 
}.reversed() * * * * * * *{ 
Returns a list of all elements sorted according to their natural sort order.
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sortedDescending(): List<UShort>

{ 
    return copyOf().apply { sort() } 
}.reversed() * * * * * * *{ 
Returns a list of all elements sorted according to their natural sort order.
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sortedArray(): UIntArray

{ 
    if (isEmpty()) return this
    return this.copyOf().apply { sort() } 
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sortedArray(): ULongArray

{ 
    if (isEmpty()) return this
    return this.copyOf().apply { sort() } 
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sortedArray(): UByteArray

{ 
    if (isEmpty()) return this
    return this.copyOf().apply { sort() } 
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sortedArray(): UShortArray

{ 
    if (isEmpty()) return this
    return this.copyOf().apply { sort() } 
}
Returns a list of all elements sorted descending according to their natural sort order. The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sortedDescending(): List<UByte> {
    return copyOf().apply { sort() }.reversed().toList() /* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting. */
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sortedDescending(): List<UShort> {
    return copyOf().apply { sort() }.reversed().toList()
}
```

Returns an array of type [ByteArray], which is a view of this array where each element is a signed reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.asByteArray(): ByteArray {
    return storage
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.asIntArray(): IntArray {
    return storage
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UIntArray.asList(): List<UInt>

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun ULongArray.asList(): List<ULong>

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UByteArray.asList(): List<UByte>

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UShortArray.asList(): List<UShort>
```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public infix fun UIntArray.contentEquals(other: UIntArray): Boolean {
    return storage.contentEquals(other.storage)
}
```

Returns an array of type [IntArray], which is a view of this array where each element is a signed reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ByteArray.asUByteArray(): UByteArray {
    return UByteArray(this)
}
```

Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.asUIntArray(): UIntArray {
    return this
}
```

Returns an array of type [LongArray], which is a view of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.asUShortArray(): UShortArray {
    return this
}
```

Returns an array of type [ShortArray], which is a view of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun LongArray.asUByteArray(): UByteArray {
    return this
}
```

Returns an array of type [ULongArray], which is a view of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ShortArray.asULongArray(): ULongArray {
    return this
}
```

Returns an array of type [ByteArrayOf], which is a view of this array where each element is a signed reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.asLongArray(): LongArray {
    return this
}
```

Returns an array of type [ShortArray], which is a view of this array where each element is a signed reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.asShortArray(): ShortArray {
    return this
}
```

Returns an array of type [UByte], which is a view of this array where each element is a signed reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.asUByteArray(): UByteArray {
    return this
}
```

Returns an array of type [ULongArray], which is a view of this array where each element is a signed reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.asUByteArray(): UByteArray {
    return this
}
```

Returns an array of type [UIntArray], which is a view of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.asUByteArrayOf(): UByteArrayOf {
    return this
}
```

Returns an array of type [ByteArray], which is a view of this array where each element is a signed reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ShortArray.asULongArray(): ULongArray {
    return this
}
```

Returns an array of type [UShortArray], which is a view of this array where each element is an unsigned reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ByteArray.asUByteArrayOf(): UByteArrayOf {
    return this
}
```
 ULonArray.contentEquals(other: ULonArray): Boolean {
    return storage.contentEquals(other.storage) * Returns `true` if the two specified arrays are structurally equal to one another, *i.e. contain the same number of the same elements in the same order.
}
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.copyInto(destination: ULongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ULongArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy,
 * 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.copyInto(destination: UByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UByteArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy,
 * 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.copyInto(destination: UShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UShortArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.copyOf(): UIntArray {
    return UIntArray(storage.copyOf())
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.copyOf(): ULongArray {
    return ULongArray(storage.copyOf())
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.copyOf(): UByteArray {
    return UByteArray(storage.copyOf())
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.copyOf(): UShortArray {
    return UShortArray(storage.copyOf())
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.copyOf(): UIntArray {
    return UIntArray()
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.copyOf(): ULongArray {
    return ULongArray()
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.copyOf(): UByteArray {
    return UByteArray()
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.copyOf(): UShortArray {
    return UShortArray()
}
`UIntArray.copyOf(newSize: Int): UIntArray` returns a new array which is a copy of the original array, resized to the given `newSize`. The copy is either truncated or padded at the end with zero values if necessary. If `newSize` is less than the size of the original array, the copy array is truncated to the `newSize`. If `newSize` is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

```kotlin
*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public inline fun UIntArray.copyOf(newSize: Int): UIntArray {
    return UIntArray(storage.copyOf(newSize))
}

/**
 * Returns new array which is a copy of the original array, resized to the given `newSize`. The copy is either truncated or padded at the end with zero values if necessary. If `newSize` is less than the size of the original array, the copy array is truncated to the `newSize`. If `newSize` is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public inline fun ULongArray.copyOf(newSize: Int): ULongArray {
    return ULongArray(storage.copyOf(newSize))
}

/**
 * Returns new array which is a copy of the original array, resized to the given `newSize`. The copy is either truncated or padded at the end with zero values if necessary. If `newSize` is less than the size of the original array, the copy array is truncated to the `newSize`. If `newSize` is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public inline fun UByteArray.copyOf(newSize: Int): UByteArray {
    return UByteArray(storage.copyOf(newSize))
}

/**
 * Returns new array which is a copy of the original array, resized to the given `newSize`. The copy is either truncated or padded at the end with zero values if necessary. If `newSize` is less than the size of the original array, the copy array is truncated to the `newSize`. If `newSize` is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public inline fun UShortArray.copyOf(newSize: Int): UShortArray {
    return UShortArray(storage.copyOf(newSize))
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public inline fun UIntArray.copyOfRange(fromIndex: Int, toIndex: Int): UIntArray {
    return UIntArray(storage.copyOfRange(fromIndex, toIndex))
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public inline fun ULongArray.copyOfRange(fromIndex: Int, toIndex: Int): ULongArray {
    return ULongArray(storage.copyOfRange(fromIndex, toIndex))
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public inline fun UByteArray.copyOfRange(fromIndex: Int, toIndex: Int): UByteArray {
    return UByteArray(storage.copyOfRange(fromIndex, toIndex))
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`

*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public inline fun UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {
    return UShortArray(storage.copyOfRange(fromIndex, toIndex))
}

/**
 * Fills this array or its subrange with the specified `element` value.
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

*\@SinceKotlin("1.3")*\@ExperimentalUnsignedTypes
public fun UIntArray.fill(element: UInt, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toInt(), fromIndex, toIndex)
}
```

**Open Source Used In DNAC Platform Antelope 3959**
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.fill(element: ULong, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toLong(), fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentExcepiton if [fromIndex] is greater than [toIndex].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.fill(element: UByte, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toByte(), fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentExcepiton if [fromIndex] is greater than [toIndex].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.fill(element: UShort, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toShort(), fromIndex, toIndex)
}

/**
 * Returns the range of valid indices for the array.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UIntArray.indices: IntRange
    get() = storage.indices

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val ULongArray.indices: IntRange
    get() = storage.indices

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UByteArray.indices: IntRange
    get() = storage.indices

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UShortArray.indices: IntRange
    get() = storage.indices

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.plus(element: UInt): UIntArray {
    return UIntArray(storage + element.toInt())
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.plus(element: ULong): ULongArray {
    return ULongArray(storage + element.toLong())
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.plus(element: UByte): UByteArray {
    return UByteArray(storage + element.toByte())
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return UShortArray(storage + element.toShort())
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.plus(elements: Collection<UInt>): UIntArray {
    var index = size
    val result = storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toInt()
    return
**Returns an array containing all elements of the original array and then all elements**

```kotlin
UIntArray(result).IllegalAccessException

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.plus(elements: UIntArray): UIntArray {
    return UIntArray(storage + elements.storage)
}

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.plus(elements: ULongArray): ULongArray {
    return ULongArray(storage + elements.storage)
}

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.plus(elements: UByteArray): UByteArray {
    return UByteArray(storage + elements.storage)
}

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UShortArray.plus(elements: UShortArray): UShortArray {
    return UShortArray(storage + elements.storage)
}

/**
 * Returns an array of type [ByteArray], which is a copy of this array where each element is a signed reinterpretation
 * of the corresponding element of this array.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.toByteArray(): ByteArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [IntArray], which is a copy of this array where each element is a signed reinterpretation
 * of the corresponding element of this array.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.toIntArray(): IntArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [LongArray], which is a copy of this array where each element is a signed reinterpretation
 * of the corresponding element of this array.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.toLongArray(): LongArray {
    return storage.copyOf()
}
```

```kotlin
UIntArray(result).IllegalAccessException

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator
```
element of this array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@kotlin.internal.InlineOnly@\@public inline fun
UShortArray.toShortArray(): ShortArray {
    return storage.copyOf()\n}  // **in* *Returns a *typed* object array containing all of the elements of this primitive array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@public fun UIntArray.toTypedArray(): Array<UInt>
    (index -> this[index]) \n}  // **in* *Returns a *typed* object array containing all of the elements of this primitive array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@public fun UByteArray.toTypedArray():
    Array<UByte> (index -> this[index]) \n}  // **in* *Returns a *typed* object array containing all of the elements of this primitive array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@public fun UShortArray.toTypedArray():
    Array<UShort> (index -> this[index]) \n}  // **in* *Returns a *typed* object array containing all of the elements of this primitive array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@public fun UByteArray.toUByteArray():
    UByteArray (this.copyOf()) \n}  // **in* *Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@public fun UIntArray.toUIntArray(): UIntArray
    (index -> this[index]) \n}  // **in* *Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@public fun ULongArray.toULongArray():
    ULongArray (index -> this[index]) \n}  // **in* *Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@public fun ShortArray.toUShortArray():
    UShortArray (index -> this[index]) \n}  // **in* *Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes@\@public fun ULongArray.toULongArray():
    ULongArray (index -> this[index]) \n}  // **in* *Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
Open Source Used In DNAC Platform Antelope

*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <R> UByteArray.flatMap(transform: (UByte) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n} \n*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <R> UShortArray.flatMap(transform: (UShort) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n} \n*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <R, C : MutableCollection<R>> ULongArray.flatMapTo(destination: C, transform: (ULong) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n} \n*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <R, C : MutableCollection<R>> UByteArray.flatMapTo(destination: C, transform: (UByte) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n} \n*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <R, C : MutableCollection<R>> UIntArray.flatMapTo(destination: C, transform: (UInt) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n} \n*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <R, C : MutableCollection<R>> UShortArray.flatMapTo(destination: C, transform: (UShort) -> Iterable<R>): C {\n    for (element in this) {\n        val list = transform(element)\n        destination.addAll(list)\n    }\n    return destination\n} \n*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <K> UIntArray.groupBy(keySelector: (UInt) -> K): Map<K, List<UInt>> {\n    return groupByTo(LinkedHashMap<K, MutableList<UInt>>(), keySelector)\n} \n*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <K> ULongArray.groupBy(keySelector: (ULong) -> K): Map<K, List<ULong>> {\n    return groupByTo(LinkedHashMap<K, MutableList<ULong>>(), keySelector)\n} \n*\@SinceKotlin("1.3")@\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun <K> UByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {\n    return groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)\n}
where each group key is associated with a list of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the original array.

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K>
UShortArray.groupBy(keySelector: (UShort) -> K): Map<K, List<UShort>> {
    return groupByTo(LinkedHashMap<K, MutableList<UShort>>(), keySelector)
}
```

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, V>
UIntArray.groupBy(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
```

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, V>
ULongArray.groupBy(keySelector: (ULong) -> K, valueTransform: (ULong) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
```

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, V>
UByteArray.groupBy(keySelector: (UByte) -> K, valueTransform: (UByte) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
```

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, M :
MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<UInt>()
        }
        list.add(element)
    }
    return destination
}
```

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, M :
MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrElse(key) {
            ArrayList<UInt>()
        }
        list.add(element)
    }
    return destination
}
```
MutableMap<in K, MutableList<ULong>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<ULong>() }
        list.add(element)
        if (element/\) return destination
    }
    return destination
}MutableMap<in K, MutableList<UInteger>> UShortArray.groupByTo(destination: M, keySelector: (UInteger) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<UInteger>() }
        list.add(element)
        if (element/\) return destination
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * * @return The [destination] map
 * * @sample samples.collections.Collections.Transformations.groupBy
 * */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
fun <K, M : MutableMap<in K, MutableList<UInteger>>> UByteArray.groupByTo(destination: M, keySelector: (UInteger) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<UInteger>() }
        list.add(element)
        if (element/\) return destination
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * * @return The [destination] map
 * * @sample samples.collections.Collections.Transformations.groupBy
 * */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
fun <K, M : MutableMap<in K, MutableList<UInteger>>> UShortArray.groupByTo(destination: M, keySelector: (UInteger) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<UInteger>() }
        list.add(element)
        if (element/\) return destination
    }
    return destination
}
destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @sample samples.collections.Collections.Transformations.map
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
    UIntArray.map(transform: (UInt) -> R): List<R> {
        return mapTo(ArrayList<R>(size), transform)
    }

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @sample samples.collections.Collections.Transformations.map
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
    ULongArray.map(transform: (ULong) -> R): List<R> {
        return mapTo(ArrayList<R>(size), transform)
    }

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @sample samples.collections.Collections.Transformations.map
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
    UByteArray.map(transform: (UByte) -> R): List<R> {
        return mapTo(ArrayList<R>(size), transform)
    }

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @sample samples.collections.Collections.Transformations.map
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
    UShortArray.map(transform: (UShort) -> R): List<R> {
        return mapTo(ArrayList<R>(size), transform)
    }

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C :
    MutableList<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {
        var index = 0
        for (item in this)
            destination.add(transform(index++, item))
        return destination
}

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C :
    MutableList<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {
        var index = 0
        for (item in this)
            destination.add(transform(index++, item))
        return destination
}

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
@SinceKotlin("1.3")
@FunctionalInterface
public inline fun <R, C :
    MutableList<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {
        var index = 0
        for (item in this)
            destination.add(transform(index++, item))
        return destination
}

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
@SinceKotlin("1.3")
@FunctionalInterface
public inline fun <R, C :
    MutableList<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
        var index = 0
        for (item in this)
            destination.add(transform(index++, item))
        return destination
}
MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and its index in the original array*
 * and appends the results to the given [destination].
 * @param [transform] function that takes the index of an
 * element and the element itself* and returns the result of the transform applied to the element.
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : 
MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and its index in the original array
 * and appends the results to the given [destination].
 * @param [transform] function that takes the index of an
 * element and the element itself* and returns the result of the transform applied to the element.
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : 
MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array* and
 * and appends the results to the given [destination].
 * @param [transform] function that takes the index of an
 * element and the element itself* and returns the result of the transform applied to the element.
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : 
MutableCollection<in R>> UIntArray.mapIndexedTo(destination: C, transform: (UInt) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : 
MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : 
MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R, C : 
MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an [IndexedValue] containing the index of that element and the element itself.*
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes
public fun UByteArray.withIndex():
    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.*
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes
public fun ULongArray.withIndex():
    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.*
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes
public fun UIntArray.withIndex():
    return IndexingIterable { iterator() }

/**
 * Returns `true` if all elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.all
 */
@SinceKotlin("1.3")@kotlin.experimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Any, C : Collection<R>> UByteArray.all(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

@OpenSourceUsedInDNACPlatformAntelope
/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.all(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
} /* Returns ‘true’ if all elements match the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.all(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
} /* Returns ‘true’ if all elements match the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.all(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
} /* Returns ‘true’ if all elements match the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.any(): Boolean {
    return storage.any()
} /* Returns ‘true’ if array has at least one element.
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.any(): Boolean {
    return storage.any()
} /* Returns ‘true’ if array has at least one element.
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.any(): Boolean {
    return storage.any()
} /* Returns ‘true’ if array has at least one element.
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.any(): Boolean {
    return storage.any()
} /* Returns ‘true’ if array has at least one element.
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.count(predicate: (UInt) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
} /* Returns the number of elements matching the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.count(predicate: (ULong) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
} /* Returns the number of elements matching the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.count(predicate: (UByte) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
} /* Returns the number of elements matching the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.count(predicate: (UShort) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
} /* Returns the number of elements matching the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.count(predicate: (UInt) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
} /* Returns the number of elements matching the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.count(predicate: (ULong) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
} /* Returns the number of elements matching the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.count(predicate: (UByte) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
} /* Returns the number of elements matching the given [predicate].
 */

/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.count(predicate: (UShort) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
} /* Returns the number of elements matching the given [predicate].
 */
*/
**Antelope 3969**

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public
inline fun UShortArray.count(predicate: (UShort) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Open Source Used In DNAC Platform Antelope 3969
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.foldRight(initial: R, operation: (UInt, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.foldRight(initial: R, operation: (ULong, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.foldRight(initial: R, operation: (UByte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.foldRight(initial: R, operation: (UShort, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.foldRightIndexed(initial: R, operation: (index: Int, UInt, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.foldRightIndexed(initial: R, operation: (index: Int, ULong, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.foldRightIndexed(initial: R, operation: (index: Int, UByte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
fun UIntArray.forEach(action: (UInt) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

ULongArray.forEach(action: (ULong) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

UByteArray.forEach(action: (UByte) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

UShortArray.forEach(action: (UShort) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

UIntArray.forEachIndexed(action: (index: Int, UInt) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

ULongArray.forEachIndexed(action: (index: Int, ULong) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

UByteArray.forEachIndexed(action: (index: Int, UByte) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

UShortArray.forEachIndexed(action: (index: Int, UShort) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.max(): UInt? {
    if (isEmpty())
        return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.max(): ULong? {
    if (isEmpty())
        return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.max(): UByte? {
    if (isEmpty())
        return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.max(): UShort? {
    if (isEmpty())
        return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UIntArray.maxBy(selector: (UInt) -> R): UInt? {
    if (isEmpty())
        return null
    var maxElem = this[0]
    var lastIndex = this.lastIndex
    if (lastIndex == 0)
        return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        if (selector(i) > maxValue)
            maxElem = i
            maxValue = selector(maxElem)
    }
    return maxElem
}
element yielding the largest value of the given function or ‘null’ if there are no elements.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R : Comparable<R>> ULongArray.maxBy(selector: (ULong) -> R): ULong? {
    if (isEmpty()) return null
    var maxElem = this[0]
    var lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte? {
    if (isEmpty()) return null
    var maxElem = this[0]
    var lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort? {
    if (isEmpty()) return null
    var maxElem = this[0]
    var lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.maxWith(comparator: Comparator<in UInt>): UInt? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.maxWith(comparator: Comparator<in ULong>): ULong? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.maxWith(comparator: Comparator<in UByte>): UByte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.min(): UInt? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.min(): ULong? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.min(): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.min(): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```
(isEmpty()) return null

var min = this[0]

for (i in 1..lastIndex) {
    val e = this[i]
    if (min > e) min = e
}

return min

Returns the first element yielding the smallest value of the given function or `null` if there are no elements.

@sample samples.collections.Collections.Aggregates.minBy

*\@SinceKotlin("1.3")*\n
// ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun <R : Comparable<R>> UIntArray.minBy(selector: (UInt) -> R): UInt? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem

    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }

    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 *
 * @sample samples.collections.Collections.Aggregates.minBy
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public inline fun <R : Comparable<R>> UByteArray.minBy(selector: (UByte) -> R): UByte? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem

    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }

    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 *
 * @sample samples.collections.Collections.Aggregates.minBy
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public inline fun <R : Comparable<R>> UShortArray.minBy(selector: (UShort) -> R): UShort? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem

    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }

    return minElem
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 *
 * @sample samples.collections.Collections.Aggregates.minWith
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public fun UIntArray.minWith(comparator: Comparator<in UInt>): UInt? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }

    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 *
 * @sample samples.collections.Collections.Aggregates.minWith
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public fun ULongArray.minWith(comparator: Comparator<in ULong>): ULong? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }

    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 *
 * @sample samples.collections.Collections.Aggregates.minWith
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public fun UByteArray.minWith(comparator: Comparator<in UByte>): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }

    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 *
 * @sample samples.collections.Collections.Aggregates.minWith
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public fun UShortArray.minWith(comparator: Comparator<in UShort>): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }

    return min
}

/**
 * Returns `true` if the array has no elements.
 *
 * @sample samples.collections.Collections.Aggregates.none
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public fun UIntArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 *
 * @sample samples.collections.Collections.Aggregates.none
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public fun ULongArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 *
 * @sample samples.collections.Collections.Aggregates.none
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public fun UByteArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 *
 * @sample samples.collections.Collections.Aggregates.none
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

public fun UShortArray.none(): Boolean {
    return isEmpty()
public inline fun ULongArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.none
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.none
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.none(predicate: (UInt) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.none(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduce(operation: (acc: UInt, UInt) -> UInt): UInt {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
Accumulates value starting with the first element and applying \([operation] \) from left to right\( ^n \) * to current accumulator value and each element with its index in the original array.\(^n \) * @param \([operation] \) function that takes the index of an element, current accumulator value\(^n \) * and the element itself and calculates the next accumulator value.\(^n \) *\n\n\n```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reduceIndexed(operation: (Int, acc: UInt, UInt) -> UInt): UInt {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

Accumulates value starting with the first element and applying \([operation] \) from left to right\( ^n \) * to current accumulator value and each element with its index in the original array.\(^n \) * @param \([operation] \) function that takes the index of an element, current accumulator value\(^n \) * and the element itself and calculates the next accumulator value.\(^n \) *\n\n\n```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.reduceIndexed(operation: (Int, acc: ULong, ULong) -> ULong): ULong {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

Accumulates value starting with the last element and applying \([operation] \) from right to left\( ^n \) to each element and current accumulator value.\(^n \) *\n\n\n```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with the last element and applying \([operation] \) from right to left\( ^n \) to each element and current accumulator value.\(^n \) *\n\n\n```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort): UShort {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with the last element and applying \([operation] \) from right to left\( ^n \) to each element and current accumulator value.\(^n \) *\n\n\n```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```
while (index >= 0) {\n    accumulator = operation(get(index--), accumulator)\n} \nreturn accumulator\n\n/*\naccumulate value starting with last element and applying [operation] from right to left to each element and current accumulator value.\n*/\n*\n\n@SinceKotlin(\"1.3\")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort) \n\n{\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(accumulator, get(index--))\n    }\n    return accumulator\n}\n\n/*\naccumulate value starting with last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.\n*/\n* @param [operation]\nfunction that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.\n*\n*\n@SinceKotlin(\"1.3\")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.reduceRightIndexed(operation: (Int, UInt, acc: UInt) -> UInt) \n\n{\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n    }\n    return accumulator\n}\n\n/*\naccumulate value starting with last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.\n*/\n* @param [operation]\nfunction that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.\n*\n*\n@SinceKotlin(\"1.3\")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.reduceRightIndexed(operation: (Int, ULong, acc: ULong) -> ULong) \n\n{\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n    }\n    return accumulator\n}\n\n/*\naccumulate value starting with last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.\n*/\n* @param [operation]\nfunction that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.\n*\n*\n@SinceKotlin(\"1.3\")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun\nUByteArray.reduceRightIndexed(operation: (Int, UByte, acc: UByte) -> UByte) \n\n{\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n    }\n    return accumulator\n}\n\n/*\naccumulate value starting with last element and applying [operation] from right to left to each element with its index in the original array and current accumulator value.\n*/\n* @param [operation]\nfunction that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.\n*\n*\n@SinceKotlin(\"1.3\")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun\nUShortArray.reduceRightIndexed(operation: (Int, UShort, acc: UShort) -> UShort) \n\n{\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n    }\n    return accumulator\n}\n\n/*\nreturns the sum of all values produced by [selector] function applied to each element in the array.\n*/\n*\n*\n@SinceKotlin(\"1.3\")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun\nUIntArray.sumBy(selector: (UInt) -> Int) \n\n{\n    var sum = 0u\n    for (element in this) \n    \n        sum += selector(element)\n    return sum\n}\n\n/*\nreturns the sum of all values produced by [selector] function applied to each element in the array.\n*/\n*\n*\n@SinceKotlin(\"1.3\")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun\nULongArray.sumBy(selector: (ULong) -> Int) \n\n{\n    var sum = 0u\n    for (element in this) \n    \n        sum += selector(element)\n    return sum\n}
function applied to each element in the array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.sumBy(selector: (UByte) -> UInt): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumBy(selector: (UShort) -> UInt): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.sumByDouble(selector: (UInt) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.sumByDouble(selector: (ULong) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.sumByDouble(selector: (UByte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumByDouble(selector: (UShort) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public infix fun <R> UIntArray.zip(other: Array<out R>): List<Pair<UInt, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public infix fun <R> ULongArray.zip(other: Array<out R>): List<Pair<ULong, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public infix fun <R> UByteArray.zip(other: Array<out R>): List<Pair<UByte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public infix fun <R> UShortArray.zip(other: Array<out R>): List<Pair<UShort, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array using the provided
 * [transform] function applied to each pair of elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, V> UIntArray.zipWithTransform(other: Array<out R>, transform: (a: UInt, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        val value = list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array using the provided
 * [transform] function applied to each pair of elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, V> ULongArray.zipWithTransform(other: Array<out R>, transform: (a: ULong, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        val value = list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array using the provided
 * [transform] function applied to each pair of elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, V> UByteArray.zipWithTransform(other: Array<out R>, transform: (a: UByte, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        val value = list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array using the provided
 * [transform] function applied to each pair of elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        val value = list.add(transform(this[i], other[i]))
    }
    return list
}
list has length of the shortest collection.

* @sample samples.collections.Operations.zipIterableWithTransform
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun <R, V> ULongArray.zip(\n*     other: Array<out R>, transform: (a: ULong, b: R) -> V): List<V> {\n*     val list = ArrayList<V>(size \n*         for (i in 0 until size) {\n*             list.add(transform(this[i], other[i]))\n*         }\n*     return list\n*     }\n*     \n*     * Returns a list of values built from the elements of ‘this’ array and the [other] array with the same index\n*     * using the provided [transform] function applied to each pair of elements.\n*     * The returned list has length of the shortest collection.\n*     * @sample samples.collections.Operations.zipIterableWithTransform\n*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun <R, V> UByteArray.zip(\n*     other: Array<out R>, transform: (a: UByte, b: R) -> V): List<V> {\n*     val list = ArrayList<V>(size \n*         for (i in 0 until size) {\n*             list.add(transform(this[i], other[i]))\n*         }\n*     return list\n*     }\n*     \n*     * Returns a list of values built from the elements of ‘this’ array and the [other] array with the same index\n*     * using the provided [transform] function applied to each pair of elements.\n*     * The returned list has length of the shortest collection.\n*     * @sample samples.collections.Operations.zipIterableWithTransform\n*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun <R, V> UShortArray.zip(\n*     other: Array<out R>, transform: (a: UShort, b: R) -> V): List<V> {\n*     val list = ArrayList<V>(size \n*         for (i in 0 until size) {\n*             list.add(transform(this[i], other[i]))\n*         }\n*     return list\n*     }\n*     \n*     * Returns a list of pairs built from the elements of ‘this’ array and [other] array with the same index.\n*     * The returned list has length of the shortest collection.\n*     * @sample samples.collections.Operations.zipIterable\n*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public infix fun <R> UIntArray.zip(\n*     other: Iterable<R>): List<Pair<UInt, R>> {\n*     return zip(other) { t1, t2 -> t1 to t2 }\n*     }\n*     \n*     * Returns a list of pairs built from the elements of ‘this’ collection and [other] array with the same index.\n*     * The returned list has length of the shortest collection.\n*     * @sample samples.collections.Operations.zipIterable\n*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public infix fun <R> ULongArray.zip(\n*     other: Iterable<R>): List<Pair<ULong, R>> {\n*     return zip(other) { t1, t2 -> t1 to t2 }\n*     }\n*     \n*     * Returns a list of pairs built from the elements of ‘this’ collection and [other] array with the same index.\n*     * The returned list has length of the shortest collection.\n*     * @sample samples.collections.Operations.zipIterable\n*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public infix fun <R> UByteArray.zip(\n*     other: Iterable<R>): List<Pair<UByte, R>> {\n*     return zip(other) { t1, t2 -> t1 to t2 }\n*     }\n*     \n*     * Returns a list of pairs built from the elements of ‘this’ collection and [other] array with the same index.\n*     * The returned list has length of the shortest collection.\n*     * @sample samples.collections.Operations.zipIterable\n*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public infix fun <R> UShortArray.zip(\n*     other: Iterable<R>): List<Pair<UShort, R>> {\n*     return zip(other) { t1, t2 -> t1 to t2 }\n*     }\n*     \n*     * Returns a list of pairs built from the elements of ‘this’ collection and [other] array with the same index.\n*     * The returned list has length of the shortest collection.\n*     * @sample samples.collections.Operations.zipIterable
list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <R, V> UByteBuffer.zip(other: Iterable<R>, transform: (a: UByte, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public infix fun UIntArray.zip(other: UIntArray): List<Pair<UInt, UInt>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public infix fun ULongArray.zip(other: ULongArray): List<Pair<ULong, ULong>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public infix fun UByteArray.zip(other: UByteArray): List<Pair<UByte, UByte>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public infix fun UShortArray.zip(other: UShortArray): List<Pair<UShort, UShort>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}
list has length of the shortest array.

* @sample
samples.collections.Iterables.Operations.zipIterableWithTransform

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public inline fun <V> UByteArray.zip(other: UByteArray, transform: (a: UByte, b: UByte) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size) for (i in 0 until size) { list.add(transform(this[i], other[i])) }
    return list
}

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size) for (i in 0 until size) { list.add(transform(this[i], other[i])) }
    return list
}

/**
 * Returns the sum of all elements in the array.
 */
@SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public fun Array<out UByte>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public fun Array<out UShort>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public fun UByteArray.zip(other: UByteArray, transform: (a: UByte, b: UByteArray) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size) for (i in 0 until size) { list.add(transform(this[i], other[i])) }
    return list
}

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public fun ULongArray.zip(other: ULongArray, transform: (a: ULong, b: ULong) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size) for (i in 0 until size) { list.add(transform(this[i], other[i])) }
    return list
}

/**
 * Returns the sum of all elements in the array.
 */
@SinceKotlin("1.3") * @ExperimentalUnsignedTypes
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size) for (i in 0 until size) { list.add(transform(this[i], other[i])) }
    return list
}

/**
 * Returns an array of UByte containing all of the elements of this collection.
 */
@SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public fun Collection<UByte>.toUByteArray(): UByteArray {
    val result = UByteArray(size)
    var index = 0
    for (element in this) { result[index++] = element
    return result
}

/**
 * Returns an array of UInt containing all of the elements of this collection.
 */
@SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public fun Collection<UInt>.toUIntArray(): UIntArray {
    val result = UIntArray(size)
    var index = 0
    for (element in this) { result[index++] = element
    return result
}

/**
 * Returns an array of ULong containing all of the elements of this collection.
 */
@SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public fun Collection<ULong>.toULongArray(): ULongArray {
    val result = ULongArray(size)
    var index = 0
    for (element in this) { result[index++] = element
    return result
}

/**
 * Returns an array of UShort containing all of the elements of this collection.
 */
@SinceKotlin("1.3") * @ExperimentalUnsignedTypes
public fun Collection<UShort>.toUShortArray(): UShortArray {
    val result = UShortArray(size)
    var index = 0
    for (element in this) { result[index++] = element
    return result
}

Open Source Used In DNAC Platform Antelope 3980
containing all of the elements of this collection.

```kotlin
fun Collection<UShort>.toUShortArray(): UShortArray {
    val result = UShortArray(size)
    var index = 0
    for (element in this) {
        result[index++] = element
    }
    return result
}

fun Iterable<UInt>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

fun Iterable<ULong>.sum(): ULong {
    var sum: ULong = 0uL
    for (element in this) {
        sum += element
    }
    return sum
}

fun Iterable<UByte>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

fun Iterable<UShort>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}
```

---

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("UComparisonsKt")

package kotlin.comparisons

import kotlin.random.*

fun maxOf(a: UInt, b: UInt): UInt {
    return if (a >= b) a else b
}

fun maxOf(a: ULong, b: ULong): ULong {
    return if (a >= b) a else b
}

fun maxOf(a: UByte, b: UByte): UByte {
    return if (a >= b) a else b
}

fun maxOf(a: UShort, b: UShort): UShort {
    return if (a >= b) a else b
}

    return maxOf(a, maxOf(b, c))
}

fun minOf(a: UInt, b: UInt): UInt {
    return if (a <= b) a else b
}

fun minOf(a: ULong, b: ULong): ULong {
    return if (a <= b) a else b
}

fun minOf(a: UByte, b: UByte): UByte {
    return if (a <= b) a else b
}

fun minOf(a: UShort, b: UShort): UShort {
    return if (a <= b) a else b
}

    return minOf(a, minOf(b, c))
}
```
fun minOf(a: ULong, b: ULong, c: ULong): ULong {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun minOf(a: Byte, b: Byte, c: Byte): Byte {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun minOf(a: Short, b: Short, c: Short): Short {
    return minOf(a, minOf(b, c))
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("URangesKt")

package kotlin.ranges

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt See:
// https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*

/**
 * Returns a random element from this range.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntRange.random(random: Random): UInt {
    try {
        return random.nextUInt(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the specified source of randomness.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongRange.random(random: Random): ULong {
    try {
        return random.nextULong(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntRange.contains(value: UByte): Boolean {
    return contains(value.toUInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: UByte): Boolean {
    return contains(value.toULong())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: UInt): Boolean {
    return contains(value.toULong())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntRange.contains(value: UShort): Boolean {
    return (value shr UInt.SIZE_BITS) == 0uL && contains(value.toUInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: UShort): Boolean {
    return contains(value.toULong())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntRange.contains(value: ULong): Boolean {
    return (value shr UInt.SIZE_BITS) == 0UL && contains(value.toUInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: ULong): Boolean {
    return (value shr ULong.SIZE_BITS) == 0UL && contains(value.toULong())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntRange.contains(value: UShort): Boolean {
    return (value shr UShort.SIZE_BITS) == 0UL && contains(value.toUInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: UShort): Boolean {
    return contains(value.toULong())
}

Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.

Open Source Used In DNAC Platform Antelope

3982
public infix fun UByte.downTo(to: UByte): UIntProgression { return UIntProgression.fromClosedRange(this.toUInt(), to.toUInt(), -1u) } * Returns a progression from this value down to the specified [to] value with the step -1. * The [to] value should be less than or equal to `this` value. * If the [to] value is greater than `this` value the returned progression is empty.

public infix fun UInt.downTo(to: UInt): UIntProgression { return UIntProgression.fromClosedRange(this, to, -1u) } * Returns a progression from this value down to the specified [to] value with the step -1. * The [to] value should be less than or equal to `this` value. * If the [to] value is greater than `this` value the returned progression is empty.

public infix fun ULong.downTo(to: ULong): ULongProgression { return ULongProgression.fromClosedRange(this, to, -1L) } * Returns a progression from this value down to the specified [to] value with the step -1. * The [to] value should be less than or equal to `this` value. * If the [to] value is greater than `this` value the returned progression is empty.

public infix fun UShort.downTo(to: UShort): UIntProgression { return UIntProgression.fromClosedRange(this.toUInt(), to.toUInt(), -1) } * Returns a progression that goes over the same range in the opposite direction with the same step.

public fun UIntProgression.reversed(): UIntProgression { return UIntProgression.fromClosedRange(last, first, -step) } * Returns a progression that goes over the same range in the opposite direction with the same step.

public fun ULongProgression.reversed(): ULongProgression { return ULongProgression.fromClosedRange(last, first, -step) } * Returns a progression that goes over the same range with the given step.

public infix fun UIntProgression.step(step: Int): UIntProgression { checkStepIsPositive(step > 0, step) return UIntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step) } * Returns a progression that goes over the same range with the given step.

public infix fun ULongProgression.step(step: Long): ULongProgression { checkStepIsPositive(step > 0, step) return ULongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step) } * Returns a range from this value up to but excluding the specified [to] value. * If the [to] value is less than or equal to `this` value, then the returned range is empty. * If the [to] value is greater than `this` value, then the returned range is from this value up to but excluding the specified [to] value. * If the [to] value is less than or equal to `this` value, then the returned range is empty. * If the [to] value is greater than `this` value, then the returned range is from this value up to but excluding the specified [to] value. * If the [to] value is less than or equal to `this` value, then the returned range is empty. * If the [to] value is greater than `this` value, then the returned range is from this value up to but excluding the specified [to] value. * If the [to] value is less than or equal to `this` value, then the returned range is empty. * If the [to] value is greater than `this` value, then the returned range is from this value up to but excluding the specified [to] value. * If the [to] value is less than or equal to `this` value, then the returned range is empty.
samples.comparisons.ComparableOps.coerceAtLeastUnsigned

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULong.coerceAtLeast(minimumValue: ULong): ULong {
    return if (this < minimumValue) minimumValue else this
}

* Ensures that this value is not less than the specified [minimumValue].
* \( \text{return this value if it's greater than or equal to the [minimumValue]} \)
* or the [minimumValue]. otherwise: \n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByte.coerceAtLeast(minimumValue: UByte): UByte {
    return if (this < minimumValue) minimumValue else this
}

* Ensures that this value is not less than the specified [minimumValue].
* \( \text{return this value if it's greater than or equal to the [minimumValue]} \)
* or the [minimumValue]. otherwise: \n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShort.coerceAtLeast(minimumValue: UShort): UShort {
    return if (this < minimumValue) minimumValue else this
}

* Ensures that this value is not less than the specified [minimumValue].
* \( \text{return this value if it's greater than or equal to the [minimumValue]} \)
* or the [minimumValue]. otherwise: \n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UInt.coerceAtMost(maximumValue: UInt): UInt {
    return if (this > maximumValue) maximumValue else this
}

* Ensures that this value is not greater than the specified [maximumValue].
* \( \text{return this value if it's less than or equal to the [maximumValue]} \)
* or the [maximumValue]. otherwise: \n * \n * @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULong.coerceAtMost(maximumValue: ULong): ULong {
    return if (this > maximumValue) maximumValue else this
}

* Ensures that this value is not greater than the specified [maximumValue].
* \( \text{return this value if it's less than or equal to the [maximumValue]} \)
* or the [maximumValue]. otherwise: \n * \n * @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByte.coerceAtMost(maximumValue: UByte): UByte {
    return if (this > maximumValue) maximumValue else this
}

* Ensures that this value is not greater than the specified [maximumValue].
* \( \text{return this value if it's less than or equal to the [maximumValue]} \)
* or the [maximumValue]. otherwise: \n * \n * @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShort.coerceAtMost(maximumValue: UShort): UShort {
    return if (this > maximumValue) maximumValue else this
}

* Ensures that this value is not greater than the specified [maximumValue].
* \( \text{return this value if it's less than or equal to the [maximumValue]} \)
* or the [maximumValue]. otherwise: \n * \n * @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UInt.coerceIn(minimumValue: UInt, maximumValue: UInt): UInt {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue)
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULong.coerceIn(minimumValue: ULong, maximumValue: ULong): ULong {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue)
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue)
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue)
```
returns maximumValue
    return this

/**
 * Ensures that this value lies in the specified range
 * [minimumValue..maximumValue].
 * 
 * @return this if it's in the range, or [minimumValue] if this value is
 * less than [minimumValue], or [maximumValue] if this value is greater than
 * [maximumValue].
 * 
 * @sample
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {
    return
    if (minimumValue > maximumValue) throw
    IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
    minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}

/**
 * Ensures that this value lies in the specified range
 * [minimumValue..maximumValue].
 * 
 * @return this if it's in the range, or [minimumValue] if this value is
 * less than [minimumValue], or [maximumValue] if this value is greater than
 * [maximumValue].
 * 
 * @sample
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {
    return
    if (minimumValue > maximumValue) throw
    IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
    minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}

/**
 * Ensures that this value lies in the specified [range].
 * 
 * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.
 * 
 * @sample
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UInt.coerceIn(range: ClosedRange<UInt>): UInt {
    return

/**
 * Ensures that this value lies in the specified [range].
 * 
 * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.
 * 
 * @sample
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULong.coerceIn(range: ClosedRange<ULong>): ULong {
    return

/**
 * Returns the sum of all elements in the sequence.
 * 
 * The operation is _terminal_.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<UInt>.sum(): UInt {
    return

/**
 * Returns the sum of all elements in the sequence.
 * 
 * The operation is _terminal_.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<ULong>.sum(): ULong {
    return

/**
 * Returns the sum of all elements in the sequence.
 * 
 * The operation is _terminal_.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<UByte>.sum(): UInt {
    return

Open Source Used In DNAC Platform Antelope 3895
public interface List<out E> : Collection<E> {
    override fun iterator(): Iterator<E>

    // Bulk Operations

    override val size: Int
    override fun isEmpty(): Boolean
    override fun contains(element: @UnsafeVariance E): Boolean
    override fun iterator(): Iterator<E>

    // Query Operations

    override val size: Int
    override fun isEmpty(): Boolean
    override fun contains(element: @UnsafeVariance E): Boolean
}

/**
 * A generic collection of elements. Methods in this interface support only read-only access
 * to the list;
 * read/write access is supported through the [MutableList] interface.
 * @param E the type of elements contained in the collection.
 */
public interface MutableList<out E> : List<E>, Collection<E>,
    MutableListInterface<out E> {
    // Query Operations
    override fun iterator(): MutableListIterator<E>

    // Modification Operations
    /**
     * Adds the specified element to the collection.
     * @param E the type of elements contained in the collection.
     */
    public fun add(element: E): Boolean

    /**
     * Removes a single instance of the specified element from this collection, if it is present.
     * @param E the type of elements contained in the collection.
     */
    public fun remove(element: E): Boolean

    /**
     * Removes all of the elements of the specified collection from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun removeAll(elements: Collection<E>): Boolean

    /**
     * Retains only the elements in this collection that are contained in the specified collection.
     * @param E the type of elements contained in the collection.
     */
    public fun retainAll(elements: Collection<E>): Boolean

    // Bulk Modification Operations
    /**
     * Adds all of the elements of the specified collection to this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun addAll(elements: Collection<E>): Boolean

    /**
     * Removes all elements from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun clear(): Unit
}

/**
 * A generic ordered collection of elements. Methods in this interface support only read-only access
 * to the collection.
 * read/write access is supported through the [MutableList] interface.
 * @param E the type of elements contained in the collection. The collection is
 * covariant on its element type.
 */
public interface Collection<out E> : Iterable<E> {
    // Query Operations

    override fun iterator(): Iterator<E>

    // Bulk Operations

    /**
     * Checks if the specified element is contained in this collection.
     */
    public operator fun contains(element: @UnsafeVariance E): Boolean

    override fun iterator(): Iterator<E>

    // Bulk Modification Operations
    /**
     * Adds all of the elements of the specified collection to this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun addAll(elements: Collection<E>): Boolean

    /**
     * Removes all elements from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun clear(): Unit
}

/**
 * A generic collection of elements. Methods in this interface support only read-only access to the collection;
 * read/write access is supported through the [MutableList] interface.
 * @param E the type of elements contained in the collection. The collection is
 * covariant on its element type.
 */
public interface MutableList<out E> : List<E>, Collection<E>,
    MutableListInterface<out E> {
    // Query Operations
    override fun iterator(): MutableListIterator<E>

    // Modification Operations
    /**
     * Adds the specified element to the collection.
     * @param E the type of elements contained in the collection.
     */
    public fun add(element: E): Boolean

    /**
     * Removes a single instance of the specified element from this collection, if it is present.
     * @param E the type of elements contained in the collection.
     */
    public fun remove(element: E): Boolean

    /**
     * Removes all of the elements of the specified collection from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun removeAll(elements: Collection<E>): Boolean

    // Bulk Modification Operations
    /**
     * Adds all of the elements of the specified collection to this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun addAll(elements: Collection<E>): Boolean

    /**
     * Removes all elements from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun clear(): Unit
}

/**
 * A generic collection of elements. Methods in this interface support only read-only access to the list;
 * read/write access is supported through the [MutableList] interface.
 * @param E the type of elements contained in the list. The list is covariant on its element type.
 */
public interface List<out E> : Collection<E>,
    ListInterface<out E> {
    // Query Operations

    override fun iterator(): Iterator<E>

    // Bulk Operations

    /**
     * Checks if all elements in the specified collection are contained in this collection.
     * @return `true` if any element was removed from the collection, `false` if the collection was not modified.
     */
    public fun retainAll(elements: Collection<E>): Boolean

    /**
     * Removes all of the elements of the specified collection from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun removeAll(elements: Collection<E>): Boolean

    // Bulk Modification Operations
    /**
     * Adds all of the elements of the specified collection to this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun addAll(elements: Collection<E>): Boolean

    /**
     * Removes all elements from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun clear(): Unit
}

/**
 * A generic ordered collection of elements. Methods in this interface support only read-only access
 * to the list; read/write access is supported through the [MutableList] interface.
 * @param E the type of elements contained in the list. The list is covariant on its element type.
 */
public interface List<out E> : Collection<E>,
    ListInterface<out E> {
    // Query Operations

    override fun iterator(): ListIterator<E>

    // Bulk Operations

    /**
     * Checks if all elements in the specified collection are contained in this collection.
     * @return `true` if any element was removed from the collection, `false` if the collection was not modified.
     */
    public fun retainAll(elements: Collection<E>): Boolean

    /**
     * Removes all of the elements of the specified collection from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun removeAll(elements: Collection<E>): Boolean

    // Bulk Modification Operations
    /**
     * Adds all of the elements of the specified collection to this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun addAll(elements: Collection<E>): Boolean

    /**
     * Removes all elements from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun clear(): Unit
}

/**
 * A generic collection of elements. Methods in this interface support only read-only access to the list;
 * read/write access is supported through the [MutableList] interface.
 * @param E the type of elements contained in the list. The list is covariant on its element type.
 */
public interface List<out E> : Collection<E>,
    ListInterface<out E> {
    // Query Operations

    override fun iterator(): ListIterator<E>

    // Bulk Operations

    /**
     * Checks if all elements in the specified collection are contained in this collection.
     * @return `true` if any element was removed from the collection, `false` if the collection was not modified.
     */
    public fun retainAll(elements: Collection<E>): Boolean

    /**
     * Removes all of the elements of the specified collection from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun removeAll(elements: Collection<E>): Boolean

    // Bulk Modification Operations
    /**
     * Adds all of the elements of the specified collection to this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun addAll(elements: Collection<E>): Boolean

    /**
     * Removes all elements from this collection.
     * @param E the type of elements contained in the collection.
     */
    public fun clear(): Unit
}
override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean // Positional Access Operations
    /**
     * Returns the element at the specified index in the list.
     */
    public operator fun get(index: Int): E
    // Search Operations
    /**
     * Returns the index of the first occurrence of the specified element in the list, or -1 if the specified element is not contained in the list.
     */
    public fun indexOf(element: @UnsafeVariance E): Int
    /**
     * Returns the index of the last occurrence of the specified element in the list, or -1 if the specified element is not contained in the list.
     */
    public fun lastIndexOf(element: @UnsafeVariance E): Int
    // Positional Access Operations
    get(index: Int: E) // Search Operations
    indexOf(element: E)
    lastIndexOf(element: E)

    /**
     * A generic unordered collection of elements that does not support adding and removing elements.
     * @param E the type of elements contained in the list. The mutable list is invariant on its element type.
     */
    public interface MutableList<E> : List<E>, MutableCollection<E> {
        // Modification Operations
        /**
         * Adds the specified element to the end of this list.
         */
        public override fun add(element: E): Boolean
        override fun addAll(elements: Collection<E>): Boolean
        override fun removeAll(elements: Collection<E>): Boolean
        override fun retainAll(elements: Collection<E>): Boolean
        /**
         * Removes an element at the specified position.
         */
        public operator fun set(index: Int, element: E): E
        /**
         * Inserts an element into the list at the specified position in this list with the specified element.
         */
        public fun add(index: Int, element: E): Boolean
        /**
         * Removes the element at the specified position.
         */
        public fun removeAt(index: Int): E
        /**
         * Inserts all of the elements of the specified collection [elements] into this list at the specified [index].
         */
        public fun addAll(index: Int, elements: Collection<E>): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun removeAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun retainAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun clear(): Unit
        override fun iterator(): MutableListIterator<E>
        override fun listIterator(): MutableListIterator<E>
        override fun subList(fromIndex: Int, toIndex: Int): List<E>
        override fun listIterator(index: Int): MutableListIterator<E>
        override fun subList(fromIndex: Int, toIndex: Int): List<E>
    }
    // Positional Access Operations
    /**
     * Replaces the element at the specified position in this list with the specified element.
     */
    public fun set(index: Int, element: E): Unit
    override fun clear(): Unit
    override fun iterator(): Iterator<E>
    override fun listIterator(): ListIterator<E>
    override fun subList(fromIndex: Int, toIndex: Int): List<E>
    /**
     * Returns a list iterator over the elements in this list (in proper sequence).
     */
    public fun listIterator(): ListIterator<E>
    /**
     * Returns a list iterator over the elements in this list (in proper sequence), starting at the specified [index].
     */
    public fun listIterator(index: Int): ListIterator<E>
    /**
     * Returns a view of the portion of this list between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).
     */
    public fun subList(fromIndex: Int, toIndex: Int): List<E>
    /**
     * Returns a view of the portion of this list between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).
     */
    public fun subList(fromIndex: Int, toIndex: Int): List<E>
    // Positional Access Operations
    /**
     * A generic unordered collection of elements that does not support duplicate elements.
     * @param E the type of elements contained in the list. The set is covariant on its element type.
     */
    public interface Set<out E> : Collection<E> {
        // Query Operations
        /**
         * Returns true if the specified element is contained in the set.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        /**
         * Returns true if the specified element is contained in the set.
         */
        override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean
        /**
         * Adds the specified element to the end of this list.
         */
        override fun add(element: E): Boolean
        override fun addAll(elements: Collection<E>): Boolean
        override fun removeAll(elements: Collection<E>): Boolean
        override fun retainAll(elements: Collection<E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: E): Boolean
        override fun add(element: E): Boolean
        override fun addAll(elements: Collection<E>): Boolean
        override fun removeAll(elements: Collection<E>): Boolean
        override fun retainAll(elements: Collection<E>): Boolean
        override fun clear(): Unit
        override fun iterator(): Iterator<E>
        override fun listIterator(): ListIterator<E>
        override fun subList(fromIndex: Int, toIndex: Int): List<E>
        /**
         * Returns the index of the first occurrence of the specified element in the list, or -1 if the specified element is not contained in the list.
         */
        override fun indexOf(element: E): Int
        /**
         * Returns the index of the last occurrence of the specified element in the list, or -1 if the specified element is not contained in the list.
         */
        override fun lastIndexOf(element: E): Int
        // Positional Access Operations
        /**
         * Determines whether the specified element is contained in this list.
         */
        public fun contains(element: E): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        public fun containsAll(elements: Collection<E>): Boolean
        /**
         * Adds the specified element to the end of this list.
         */
        public fun add(element: E): Boolean
        /**
         * Adds all of the elements of the specified collection to the end of this list.
         */
        public fun addAll(elements: Collection<E>): Boolean
        public fun addAll(index: Int, elements: Collection<E>): Boolean
        /**
         * Removes an element at the specified position.
         */
        public operator fun set(index: Int, element: E): E
        /**
         * Inserts an element into the list at the specified position.
         */
        public fun add(index: Int, element: E): Boolean
    }
    // Positional Access Operations
    /**
     * A generic unordered collection of elements that does not support duplicate elements.
     * @param E the type of elements contained in the list. The set is covariant on its element type.
     */
    public interface HashSet<E> : Collection<E>, Set<E> {
        // Query Operations
        /**
         * Returns true if the specified element is contained in the set.
         */
        override fun contains(element: E): Boolean
        /**
         * Returns true if the specified element is contained in the set.
         */
        override fun containsAll(elements: Collection<E>): Boolean
        /**
         * Adds the specified element to the end of this list.
         */
        override fun add(element: E): Boolean
        override fun addAll(elements: Collection<E>): Boolean
        override fun removeAll(elements: Collection<E>): Boolean
        override fun retainAll(elements: Collection<E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        override fun add(element: @UnsafeVariance E): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun removeAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun retainAll(elements: Collection<@UnsafeVariance E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        override fun add(element: @UnsafeVariance E): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun removeAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun retainAll(elements: Collection<@UnsafeVariance E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        override fun add(element: @UnsafeVariance E): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun removeAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun retainAll(elements: Collection<@UnsafeVariance E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        override fun add(element: @UnsafeVariance E): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun removeAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun retainAll(elements: Collection<@UnsafeVariance E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        override fun add(element: @UnsafeVariance E): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun removeAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun retainAll(elements: Collection<@UnsafeVariance E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        override fun add(element: @UnsafeVariance E): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun removeAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun retainAll(elements: Collection<@UnsafeVariance E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        override fun add(element: @UnsafeVariance E): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun removeAll(elements: Collection<@UnsafeVariance E>): Boolean
        override fun retainAll(elements: Collection<@UnsafeVariance E>): Boolean
        /**
         * Determines whether the specified element is contained in this list.
         */
        override fun contains(element: @UnsafeVariance E): Boolean
        override fun add(element: @UnsafeVariance E): Boolean
        override fun addAll(elements: Collection<@UnsafeVariance E>): Boolean
        override func
key. Map keys are unique; the map holds only one value for each key.

Methods in this interface support only read-only access to the map; read-write access is supported through the [MutableMap] interface. The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.

The map is invariant on its key type, as it can accept key as a parameter (of [containsKey] for example) and return it in [keys]. The map is covariant on its value type, and its generic type arguments are [K] and [V], where [K] is the type of keys and [V] is the type of values.
the next value in the sequence without boxing. */

// Auto-generated file. DO NOT EDIT!

package kotlin.ranges

/**
 * An iterator over a sequence of values of type `Char`. */
public abstract class CharProgressionIterator(first: Char, last: Char, val step: Int) :
  Iterator<Char> { /* */
    override final fun next() = nextChar() /* Returns the next value in the sequence without boxing. */

    override fun hasNext(): Boolean = hasNext

    override fun nextInt(): Int {
        val value = next
        if (value == finalElement) {
            if (!hasNext) throw kotlin.NoSuchElementException()
            hasNext = false
        } else {
            next += step
            hasNext = true
        }
        return value.toChar()
    }
}

/**
 * An iterator over a sequence of values of type `Boolean`. */
public abstract class BooleanIterator : Iterator<Boolean> { /* */
  override final fun next() = nextBoolean() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextDouble(): Double
}

/** An iterator over a sequence of values of type `Double`. */
public abstract class DoubleIterator : Iterator<Double> { /* */
    override final fun next() = nextDouble() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextFloat(): Float
}

/** An iterator over a sequence of values of type `Int`. */
public abstract class IntIterator : Iterator<Int> { /* */
    override final fun next() = nextInt() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextLong(): Long
}

/** An iterator over a sequence of values of type `Long`. */
public abstract class LongIterator : Iterator<Long> { /* */
    override final fun next() = nextLong() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextShort(): Short
}

/** An iterator over a sequence of values of type `Short`. */
public abstract class ShortIterator : Iterator<Short> { /* */
    override final fun next() = nextShort() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextCharacter(): Char
}

/** An iterator over a sequence of values of type `Char`. */
public abstract class CharIterator : Iterator<Char> { /* */
    override final fun next() = next() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextBoolean(): Boolean
}

/** An iterator over a sequence of values of type `Boolean`. */
public abstract class BooleanIterator : Iterator<Boolean> { /* */
    override final fun next() = nextBoolean() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextDouble(): Double
}

/** An iterator over a sequence of values of type `Double`. */
public abstract class DoubleIterator : Iterator<Double> { /* */
    override final fun next() = nextDouble() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextFloat(): Float
}

/** An iterator over a sequence of values of type `Int`. */
public abstract class IntIterator : Iterator<Int> { /* */
    override final fun next() = nextInt() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextLong(): Long
}

/** An iterator over a sequence of values of type `Long`. */
public abstract class LongIterator : Iterator<Long> { /* */
    override final fun next() = nextLong() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextShort(): Short
}

/** An iterator over a sequence of values of type `Short`. */
public abstract class ShortIterator : Iterator<Short> { /* */
    override final fun next() = nextShort() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextCharacter(): Char
}

/** An iterator over a sequence of values of type `Char`. */
public abstract class CharIterator : Iterator<Char> { /* */
    override final fun next() = nextCharacter() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextBoolean(): Boolean
}

/** An iterator over a sequence of values of type `Boolean`. */
public abstract class BooleanIterator : Iterator<Boolean> { /* */
    override final fun next() = nextBoolean() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextDouble(): Double
}

/** An iterator over a sequence of values of type `Double`. */
public abstract class DoubleIterator : Iterator<Double> { /* */
    override final fun next() = nextDouble() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextFloat(): Float
}

/** An iterator over a sequence of values of type `Int`. */
public abstract class IntIterator : Iterator<Int> { /* */
    override final fun next() = nextInt() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextLong(): Long
}

/** An iterator over a sequence of values of type `Long`. */
public abstract class LongIterator : Iterator<Long> { /* */
    override final fun next() = nextLong() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextShort(): Short
}

/** An iterator over a sequence of values of type `Short`. */
public abstract class ShortIterator : Iterator<Short> { /* */
    override final fun next() = nextShort() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextCharacter(): Char
}

/** An iterator over a sequence of values of type `Char`. */
public abstract class CharIterator : Iterator<Char> { /* */
    override final fun next() = nextCharacter() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextBoolean(): Boolean
}

/** An iterator over a sequence of values of type `Boolean`. */
public abstract class BooleanIterator : Iterator<Boolean> { /* */
    override final fun next() = nextBoolean() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextDouble(): Double
}

/** An iterator over a sequence of values of type `Double`. */
public abstract class DoubleIterator : Iterator<Double> { /* */
    override final fun next() = nextDouble() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextFloat(): Float
}

/** An iterator over a sequence of values of type `Int`. */
public abstract class IntIterator : Iterator<Int> { /* */
    override final fun next() = nextInt() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextLong(): Long
}

/** An iterator over a sequence of values of type `Long`. */
public abstract class LongIterator : Iterator<Long> { /* */
    override final fun next() = nextLong() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextShort(): Short
}

/** An iterator over a sequence of values of type `Short`. */
public abstract class ShortIterator : Iterator<Short> { /* */
    override final fun next() = nextShort() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextCharacter(): Char
}

/** An iterator over a sequence of values of type `Char`. */
public abstract class CharIterator : Iterator<Char> { /* */
    override final fun next() = nextCharacter() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextBoolean(): Boolean
}

/** An iterator over a sequence of values of type `Boolean`. */
public abstract class BooleanIterator : Iterator<Boolean> { /* */
    override final fun next() = nextBoolean() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextDouble(): Double
}

/** An iterator over a sequence of values of type `Double`. */
public abstract class DoubleIterator : Iterator<Double> { /* */
    override final fun next() = nextDouble() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextFloat(): Float
}

/** An iterator over a sequence of values of type `Int`. */
public abstract class IntIterator : Iterator<Int> { /* */
    override final fun next() = nextInt() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextLong(): Long
}

/** An iterator over a sequence of values of type `Long`. */
public abstract class LongIterator : Iterator<Long> { /* */
    override final fun next() = nextLong() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextShort(): Short
}

/** An iterator over a sequence of values of type `Short`. */
public abstract class ShortIterator : Iterator<Short> { /* */
    override final fun next() = nextShort() /* Returns the next value in the sequence without boxing. */

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextCharacter(): Char
}
**Open Source Used In DNAC Platform Antelope 3990**

JetBrains s.r.o. and Kotlin Programming Language contributors.

* Use of this source code is governed by the

```kotlin
LongProgression = LongProgression(rangeStart, rangeEnd, step)
```

```
/*
 * Copyright 2010-2019
         */
        public fun fromClosedRange(rangeStart: Long, rangeEnd: Long, step: Long):
```

```
* [step] must be greater than `Long.MIN_VALUE` and not equal to zero.

    companion object {
        /**
         * Creates LongProgression within the specified bounds of a closed range.
```

    /**
     * The step of the progression.
     */
    public val step: Long = step

    override fun iterator(): LongIterator =
```

```
/**
 * The first element in the progression.

    /**
     * The step
     */
    public val first: Long = start

    /**
     * The last element in the progression.
     */
    public val last: Long = getProgressionLastElement(start.toLong(), endInclusive.toLong(), step).toLong()

    /**
     * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].
```

    companion object {
        /**
         * Creates LongProgression within the specified bounds of a closed range.
```

    /**
     * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].
```

    companion object {
        /**
         * Creates CharProgression within the specified bounds of a closed range.
```

    /**
     * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].
```

    companion object {
        /**
         * Creates CharProgression within the specified bounds of a closed range.
```

    /**
     * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].
```

    companion object {
        /**
         * Creates IntProgression within the specified bounds of a closed range.
```

    /**
     * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].
```

    companion object {
        /**
         * Creates IntProgression within the specified bounds of a closed range.
```

    /**
     * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].
```

    companion object {
        /**
         * Creates IntProgression within the specified bounds of a closed range.
```
Apache 2.0 license that can be found in the license/LICENSE.txt file.

package kotlin.ranges

/**
 * Represents a range of values (for example, numbers or characters).
 * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/ranges.html) for more information.
 */

public interface ClosedRange<T: Comparable<T>> {
    /**
     * The minimum value in the range.
     */
    public val start: T

    /**
     * The maximum value in the range (inclusive).
     */
    public val endInclusive: T

    /**
     * Checks whether the specified [value] belongs to the range.
     */
    public operator fun contains(value: T): Boolean = value >= start && value <= endInclusive

    /**
     * Checks whether the range is empty.
     */
    public fun isEmpty(): Boolean = start > endInclusive
}

package kotlin

/**
 * The type with only one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override fun toString() = "kotlin.Unit"
}

package kotlin.annotation

import kotlin.annotation.AnnotationTarget.*

/**
 * Contains the list of code elements which are the possible annotation targets
 * Class, interface or object, annotation class is also included
 */
public enum class AnnotationTarget {
    CLASS,
    ANNOTATION_CLASS,
    TYPE_PARAMETER,
    PROPERTY,
    FIELD,
    LOCAL_VARIABLE,
    VALUE_PARAMETER,
    CONSTRUCTOR,
    FUNCTION,
    PROPERTY_GETTER,
    PROPERTY_SETTER,
    TYPE,
    EXPRESSION,
    FILE,
    TYPEALIAS
}

/**
 * Contains the list of possible annotation's retentions.
 * Determines how an annotation is stored in binary output.
 */
public enum class AnnotationRetention {
    SOURCE,
    BINARY,
    RUNTIME
}

/**
 * This meta-annotation indicates the kinds of code elements which are possible targets of an annotation.
 */
@Target(AnnotationTarget.ANNOTATION_CLASS)
@MustBeDocumented
public annotation class Target(vararg val allowedTargets: AnnotationTarget)

/**
 * This meta-annotation determines whether an annotation is stored in binary output and visible for reflection (default retention)
 */
@Target(AnnotationTarget.ANNOTATION_CLASS)
public annotation class Repeatable

/**
 * This
meta-annotation determines that an annotation is a part of public API and therefore should be included in the generated documentation for the element to which the annotation is applied.

```kotlin
@Target(AnnotationTarget.ANNOTATION_CLASS)
public annotation class MustBeDocumented
```

```kotlin
@Target(AnnotationTarget.ANNOTATION_CLASS)
public annotation class MustBeDocumented
```

```kotlin
/*
 * Copyright 2010-2016 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

```kotlin
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

```kotlin
package kotlin.internal
```

```kotlin
/**
 * Specifies that the corresponding type parameter is not used for unsafe operations such as casts or 'is' checks
 * That means it's completely safe to use generic types as argument for such parameter.
 */
@Target(AnnotationTarget.TYPE_PARAMETER)
@Retention(AnnotationRetention.BINARY)
internal annotation class PureReifiable
```

```kotlin
/**
 * Specifies that the corresponding built-in method exists depending on platform.
 */
@Target(AnnotationTarget.FUNCTION)
@Retention(AnnotationRetention.BINARY)
internal annotation class PlatformDependent
```

```kotlin
package kotlin.internal
```

```kotlin
private fun mod(a: Int, b: Int): Int {
    val mod = a % b
    return if (mod >= 0) mod else mod + b
}
```

```kotlin
private fun mod(a: Long, b: Long): Long {
    val mod = a % b
    return if (mod >= 0) mod else mod + b
}
```

```kotlin
/**
 * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative [step]. No validation on passed parameters is performed. The given parameters should satisfy the condition: [start] <= [end]
 * @param start first element of the progression
 * @param end ending bound for the progression
 * @param step increment, or difference of successive elements in the progression
 * @return the final element of the progression
 */
@PublishedApi
internal fun getProgressionLastElement(start: Int, end: Int, step: Int): Int = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)
    else -> throw kotlin.IllegalArgumentException("Step is zero.")
}
```

```kotlin
/**
 * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative [step]. No validation on passed parameters is performed. The given parameters should satisfy the condition: [start] <= [end]
 * @param start first element of the progression
 * @param end ending bound for the progression
 * @param step increment, or difference of successive elements in the progression
 * @return the final element of the progression
 */
@PublishedApi
internal fun getProgressionLastElement(start: Long, end: Long, step: Long): Long = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)
    else -> throw kotlin.IllegalArgumentException("Step is zero.")
}
```

```kotlin
/*
 * Copyright 2010-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

```kotlin
/*
 * Copyright 2010-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */
```
KType?\n) {\n    public companion object {\n        /**\n         * Star projection, denoted by the `*` character.\n         * For example, in the type `KClass<*>>`, `*` is the star projection.\n         * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#star-projections) for more information.\n         */\n        public val STAR: KTypeProjection = KTypeProjection(null, null)\n
        /**\n         * Creates an invariant projection of a given type. Invariant projection is just the type itself,\n         * without any use-site variance modifiers applied to it.\n         * For example, in the type `Set<String>` , `String` is an invariant projection of the type represented by the class `String`.\n         */\n        public fun invariant(type: KType): KTypeProjection = KTypeProjection(KVariance.INVARIANT, type)\n
        /**\n         * Creates a contravariant projection of a given type, denoted by the `in` modifier applied to a type.\n         * For example, in the type `MutableList<in Number>`, `in Number` is a contravariant projection of the type of class `Number`.\n         */\n        public fun contravariant(type: KType): KTypeProjection = KTypeProjection(KVariance.IN, type)\n
        /**\n         * Creates a covariant projection of a given type, denoted by the `out` modifier applied to a type.\n         * For example, in the type `Array<out Number>`, `out Number` is a covariant projection of the type of class `Number`.\n         */\n        public fun covariant(type: KType): KTypeProjection = KTypeProjection(KVariance.OUT, type)\n    }\n}\n
@SinceKotlin("1.1")\nenum class KVariance {
    /**\n     * The affected type parameter or type is *invariant*, which means it has no variance applied to it.\n     */\n    INVARIANT,

    /**\n     * The affected type parameter or type is *contravariant*. Denoted by the `in` modifier in the source code.\n     */\n    IN,

    /**\n     * The affected type parameter or type is *covariant*. Denoted by the `out` modifier in the source code.\n     */\n    OUT,
}

@SinceKotlin("1.1")\nenum class KVisibility {
    /**\n     * Visibility of declarations marked with the `public` modifier.\n     */\n    PUBLIC,

    /**\n     * Visibility of declarations marked with the `protected` modifier.\n     */\n    PROTECTED,

    /**\n     * Visibility of declarations marked with the `internal` modifier.\n     */\n    INTERNAL,

    /**\n     * Visibility of declarations marked with the `private` modifier.\n     */\n    PRIVATE,
}

@JsName("arrayIterator")\ninternal fun arrayIterator(array: dynamic, type: String?)\n    = when (type) {
        null -> {
            val arr: Array<dynamic> = array
            object : Iterator<dynamic> {
                var index = 0
                override fun hasNext() = index < arr.size
                override fun next() = if (index <
arr.size) arr[index++] else throw NoSuchElementException("$index")
}
}

@JsName("booleanArrayIterator")
internal fun booleanArrayIterator(array: BooleanArray) =
    object : BooleanIterator()
    {
        var index = 0
        override fun hasNext() = index < array.size
        override fun nextBoolean() = if (index < array.size) array[index++] else throw
    }

@JsName("byteArrayIterator")
internal fun byteArrayIterator(array: ByteArray) = object : ByteIterator()
    {
        var index = 0
        override fun hasNext() = index < array.size
        override fun nextByte() = if (index < array.size) array[index++] else throw
    }

@JsName("shortArrayIterator")
internal fun shortArrayIterator(array: ShortArray) = object : ShortIterator()
    {
        var index = 0
        override fun hasNext() = index < array.size
        override fun nextShort() = if (index < array.size) array[index++] else throw
    }

@JsName("charArrayIterator")
internal fun charArrayIterator(array: CharArray) = object : CharIterator()
    {
        var index = 0
        override fun hasNext() = index < array.size
        override fun nextChar() = if (index < array.size) array[index++] else throw
    }

@JsName("intArrayIterator")
internal fun intArrayIterator(array: IntArray) = object : IntIterator()
    {
        var index = 0
        override fun hasNext() = index < array.size
        override fun nextInt() = if (index < array.size) array[index++] else throw
    }

@JsName("floatArrayIterator")
internal fun floatArrayIterator(array: FloatArray) = object : FloatIterator()
    {
        var index = 0
        override fun hasNext() = index < array.size
        override fun nextFloat() = if (index < array.size) array[index++] else throw
    }

@JsName("doubleArrayIterator")
internal fun doubleArrayIterator(array: DoubleArray) = object : DoubleIterator()
    {
        var index = 0
        override fun hasNext() = index < array.size
        override fun nextDouble() = if (index < array.size) array[index++] else throw
    }

@JsName("longArrayIterator")
internal fun longArrayIterator(array: LongArray) = object : LongIterator()
    {
        var index = 0
        override fun hasNext() = index < array.size
        override fun nextLong() = if (index < array.size) array[index++] else throw
    }

@JsName("PropertyMetadata")
internal class PropertyMetadata(@JsName("callableName") val name: String)

@JsName("noWhenBranchMatched")
internal fun noWhenBranchMatched(): Nothing = throw NoWhenBranchMatchedException()

@JsName("subSequence")
internal fun subSequence(c: CharSequence, startIndex: Int, endIndex: Int): CharSequence {
    if (c is String) {
        return c.substring(startIndex, endIndex)
    } else {
        return c.asDynamic().`subSequence_vux9f0$`(startIndex, endIndex)
    }
}

@JsName("captureStack")
internal fun captureStack(baseClass: JsClass<in Throwable>, instance: Throwable) {
    if (js("Error").captureStackTrace) {
        js("Error").captureStackTrace(instance, instance::class.js)
    } else {
        instance.asDynamic().stack = js("new Error()").stack
    }
}

@JsName("newThrowable")
internal fun newThrowable(message: String?, cause: Throwable?): Throwable {
    val throwable = js("new Error()")
    if (message != null) throwable.message = message
    throwable.cause = cause
    throwable.name = "Throwable"
    return throwable
}

@JsName("BoxedChar")
internal class BoxedChar(val c: Int) : Comparable<Int> {
    override fun equals(other: Any?): Boolean {
        return other is BoxedChar && c == other.c
    }

    override fun hashCode(): Int {
        return c
    }

    override fun toString(): String {
        return js("this.c").unsafeCast<Char>().toString()
    }

    override fun compareTo(other: Int): Int {
        return js("this.c - other.c").unsafeCast<Int>().compareTo(other)
    }
}

@kotlin.internal.InlineOnly
internal inline fun <T> concat(args: Array<T>): T {
    val typed
fun arrayConcat(a: T, b: T): T {
    return (a as Array<*> && b as Array<*>)
        ? a.concat(b)
        : when (a is Array<*>)
            (TypedArray::class.java.isAssignableFrom(a.javaClass))
                .then { typedArrayConcat(a as Array<*>!, b as Array<*>) }
srcLen && index < dstLen) dst[index] = src[index++]
    return dst
}


internal fun arrayCopyResize(source: dynamic, newSize: Int, defaultValue: Any?): dynamic {
    val result = source.slice(0, newSize)
    return result
}

internal fun arrayCopy(array: dynamic, collection: Collection<T>): dynamic {
    val dst = array.newInstance(collection.size)
    copyArrayType(array, dst)
    var index = 0
    for (element in collection) dst[index++] = element
    return dst
}

internal fun arrayPlusCollection(array: dynamic, collection: Collection<T>): dynamic {
    val dst = array.slice()
    dst.length += collection.size
    copyArrayType(array, dst)
    var index = 0
    for (element in collection) dst[index++] = element
    return dst
}

internal fun fillFromCollection(dst: dynamic, startIndex: Int, collection: Collection<T>): dynamic {
    var index = startIndex
    for (element in collection) dst[index++] = element
    return dst
}

internal inline fun copyArrayType(from: dynamic, to: dynamic) {
    if (from.`$type$` !== undefined) {
        to.`$type$` = from.`$type$`
    }
}

internal inline fun jsIsType(obj: dynamic, jsClass: dynamic) = js("Kotlin").isType(obj, jsClass)
represents [success][Result.isSuccess] or the\n* defaultValue if it is [failure][Result.isFailure].\n* This
function is shorthand for `getOrElse { defaultValue }` (see [getOrElse]).\n
```kotlin
@InlineOnly
@SinceKotlin("1.3")
public inline fun <R, T : R> Result<T>.getOrElse(defaultValue: R): R {
    if (isFailure) return defaultValue
    return value as T
}
```

This function is shorthand for `getOrElse { defaultValue }` (see [getOrElse]).\n
```kotlin
@InlineOnly
@SinceKotlin("1.3")
public inline fun <R, T> Result<T>.fold(
    onSuccess: (value: T) -> R,
    onFailure: (exception: Throwable) -> R
): R {
    contract {
        callsInPlace(onSuccess, InvocationKind.AT_MOST_ONCE)
        callsInPlace(onFailure, InvocationKind.AT_MOST_ONCE)
    }
    return when (val exception = exceptionOrNull()) {
        null -> onSuccess(value as T)
        else -> onFailure(exception)
    }
}
```

This function is shorthand for `getOrElse { defaultValue }` (see [getOrElse]).\n
```kotlin
@InlineOnly
@SinceKotlin("1.3")
public inline fun <R, T> Result<T>.map(transform: (value: T) -> R): Result<R> {
    contract {
        callsInPlace(transform, InvocationKind.AT_MOST_ONCE)
    }
    return when {
        isSuccess -> Result.success(transform(value as T))
        else -> Result(value)
    }
}
```

This function is shorthand for `getOrElse { defaultValue }` (see [getOrElse]).\n
```kotlin
@InlineOnly
@SinceKotlin("1.3")
public inline fun <R, T : R> Result<T>.recover(transform: (exception: Throwable) -> R): Result<R> {
    contract {
        callsInPlace(transform, InvocationKind.AT_MOST_ONCE)
    }
    return when (val exception = exceptionOrNull()) {
        null -> this
        else -> Result.success(transform(exception))
    }
}
```

This function is shorthand for `getOrElse { defaultValue }` (see [getOrElse]).
found in the license/LICENSE.txt file.

```kotlin
package kotlin.coroutines

import kotlin.coroutines.*
import kotlin.coroutines.intrinsics.*
import kotlin.internal.InlineOnly

/**
 * Interface representing a continuation after a suspension point that returns a value of type `T`.
 */
public interface Continuation<in T> {
    /**
     * The context of the coroutine that corresponds to this continuation.
     */
    public val context: CoroutineContext

    /**
     * Resumes the execution of the corresponding coroutine passing a successful or failed
     * [result] as the return value of the last suspension point.
     */
    public fun resumeWith(result: Result<T> =@SinceKotlin("1.3")@Target(AnnotationTarget.CLASS)@Retention(AnnotationRetention.BINARY)@pub
     lic annotation class RestrictionsSuspension@n/**@n * Resumes the execution of the corresponding coroutine passing
     * [value] as the return value of the last suspension point.
     */
    public fun<T> Continuation<in T>.resume(value: T) =@SinceKotlin("1.3")@InlineOnly@npublic inline fun <T> Continuation<T>.resume(value: T): Unit =
    resumeWith(Result.success(value))

    /**
     * Resumes the execution of the corresponding coroutine so that the [exception] is re-thrown right after the
     * last suspension point.
     */
    public fun<T> Continuation<in T>.resumeWithException(exception: Throwable) =
    resumeWith(Result.failure(exception))

    /**
     * Creates a [Continuation] instance with the given [context] and
     * implementation of [resumeWith] method.
     */
    public fun<T> Continuation<in T>(context: CoroutineContext, crossinline resumeWith: (Result<T>) -> Unit): Continuation<T> =
        object : Continuation<T> {
            override val context: CoroutineContext
                get() = context

            override fun resumeWith(result: Result<T>) =
                resumeWith(result)
        }

    /**
     * Creates a coroutine without a receiver and with result type [T].
     * This function creates a new, fresh instance of suspendable computation every time it is invoked.
     * To start executing the created coroutine, invoke `resume(Unit)` on the returned
     * [continuation] instance.
     */
    public fun<T> (suspend () -> T).createCoroutine(completion: Continuation<T>) =
        createCoroutineUnintercepted(completion).intercepted().resume(Unit)

    /**
     * Creates a coroutine with receiver type [R] and result type [T].
     * This function creates a new, fresh instance of suspendable computation every time it is invoked.
     * To start executing the created coroutine, invoke `resume(Unit)` on the returned
     * [continuation] instance.
     */
    public fun<T, R>(R.() -> T).createCoroutine(receiver: R, completion: Continuation<T>) =
        createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)

    /**
     * Starts a coroutine without a receiver and with result type [T].
     * This function creates and starts a new, fresh instance of suspendable computation every time it is invoked.
     * The [completion] continuation is invoked when the coroutine completes with a result or an exception.
     */
    public fun<T> (suspend () -> T).startCoroutine(completion: Continuation<T>) =
        createCoroutineUnintercepted(completion).intercepted().resume(Unit)

    /**
     * Starts a coroutine with receiver type [R] and result type [T].
     * This function creates and starts a new, fresh instance of suspendable computation every time it is invoked.
     * The [completion] continuation is invoked when the coroutine completes with a result or an exception.
     */
    public fun<T, R>(R.() -> T).startCoroutine(receiver: R, completion: Continuation<T>) =
        createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)

    /**
     * Obtains the current continuation instance inside suspend functions and suspends
     * the currently running coroutine.
     */
    public fun<T> Continuation<in T>.suspended(): Continuation<T> =
        suspendedContinuation
```

---

Open Source Used In DNAC Platform Antelope 4000
function both [Continuation.resume] and [Continuation.resumeWithException] can be used either synchronously or
asynchronously later in the same thread or from a different thread of execution. Subsequent invocation of any resume function will produce an
[IllegalStateException].

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* An @SinceKotlin("1.3") @Suppress("WRONG_MODIFIER_TARGET") @InlineOnly @public suspend inline fun <T>

suspendCoroutine(crossinline block: (Continuation<T>) -> Unit): T =
    suspendCoroutineUninterceptedOrReturn {
        c: Continuation<T> ->
        val safe = SafeContinuation(c.intercepted())
        block(safe)
    safe.getOrThrow()
}

/**
 * Returns the context of the current coroutine.

*/
@SinceKotlin("1.3")
@RequireInlineOnly
public suspend inline
public val coroutineContext: CoroutineContext
    get() {
        throw NotImplementedError("Implemented as

intrinsic")
    }

/**
 * Creates unintercepted coroutine without a receiver and with result type [T].
 * This function creates a new, fresh instance of suspendable computation every time it is
 * invoked. The suspended coroutine is started immediately in the invoker's thread
 * without going through the [ContinuationInterceptor] that might be present in the
 * completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation context is
 * established.
 * This function returns unintercepted continuation.
 * Invocation of `resume(Unit)` on the suspended
 * coroutine starts coroutine immediately in the invoker's call stack without going through the
 * [ContinuationInterceptor] that might be present in the
 * completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation context is
 * established.
 * This function can be used to acquire the intercepted continuation.
 * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of
 * both the coroutine and [completion] happens in the
 * invocation context established by
 * [ContinuationInterceptor].
 * Repeated invocation of any resume function on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or
 * exception.

* An @SinceKotlin("1.3") @public @suppress("WrongModifierTarget") @inline Only @public suspended inline func <T>

suspendCoroutineUninterceptedOrReturn {
    c: Continuation<T> ->
    val coroutineContext: CoroutineContext
    get() {
        throw NotImplementedError("Implemented as

 intrinsic")
    }

    suspendCoroutine(crossinline block: (Continuation<T>) -> Unit): T =
    suspendCoroutineUninterceptedOrReturn {
        c: Continuation<T> ->
        val safe = SafeContinuation(c.intercepted())
        block(safe)
    safe.getOrThrow()
}
createCoroutineFromSuspendFunction(completion) { \n    this.asDynamic()(completion)\n} \n\n/*
 * Creates unintercepted coroutine with receiver type [R] and result type [T].\n * This function creates a new, fresh instance of suspendable computation every time it is invoked.\n */

/*
 * This function returns unintercepted continuation.\n */

Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the\n * [ContinuationInterceptor] that might be present in the continuation's [CoroutineContext].\n * It is the invoker's responsibility to ensure that a proper invocation context is established.\n * Note that [completion] of this function may get invoked in an arbitrary context.\n * [Continuation.intercepted] can be used to acquire the intercepted continuation.\n */
@SinceKotlin("1.3")
public actual fun <R, T>
    (suspend R.() -> T).createCoroutineUnintercepted(\n        receiver: R,\n        completion: Continuation<T>\n    ): Continuation<Unit> =
    // Kotlin/JS suspend lambdas have an extra parameter `suspended`\n    if (this.asDynamic().length == 3) {\n        // When `suspended` is true the continuation is created, but not executed\n        this.asDynamic()(receiver, completion, true)\n    } else {\n        createCoroutineFromSuspendFunction(completion) {\n            this.asDynamic()(receiver, completion)\n        }\n    }

/*
 * This function shall be used on the immediate result of\n * [createCoroutineUnintercepted] or [suspendCoroutineUninterceptedOrReturn],\n * in which case it checks for\n * [ContinuationInterceptor] in the continuation's [context].\n */
@SinceKotlin("1.3")
public actual\n    fun <T> Continuation<T>.intercepted(): Continuation<T> =
    (this as? CoroutineImpl)?.intercepted() ?: this

private inline fun <T> createCoroutineFromSuspendFunction(\n    completion: Continuation<T>,\n    crossinline block: () -> Any?\n): Continuation<Unit> {
    @Suppress("UNCHECKED_CAST")
    return object : CoroutineImpl(completion as Continuation<Any?>) {
        override fun doResume(): Any? {
            exception?.let { throw it }
            return block()\n        }
    }
}
public actual fun Double.toBits(): Long = definedExternally

/**
 * Returns a bit representation of the specified floating-point value as [Long] * preserving NaN values exact layout.
 */
public actual fun Double.toRawBits(): Long = definedExternally

/**
 * Returns the Double value corresponding to a given bit representation.
 */

/**
 * Returns a bit representation of the specified floating-point value as [Int] * according to the IEEE 754 floating-point \"single format\" bit layout.
 */
public actual fun Float.toBits(): Int = definedExternally

/**
 * Returns a bit representation of the specified floating-point value as [Int] * according to the IEEE 754 floating-point \"single format\" bit layout.
 */
public actual fun Float.toRawBits(): Int = definedExternally

/**
 * Returns the Float value corresponding to a given bit representation.
 */
public actual inline fun Float.Companion.fromBits(bits: Int): Float = js("Kotlin").floatFromBits(bits).unsafeCast<Float>()

internal inline fun Long(low: Int, high: Int) = js("Kotlin").Long.fromBits(low, high).unsafeCast<Long>()
internal inline val Long.low: Int get() = this.asDynamic().getLowBits().unsafeCast<Int>()
internal inline val Long.high: Int get() = this.asDynamic().getHighBits().unsafeCast<Int>()
Open Source Used In DNAC Platform Antelope 4005
(which can't be `external`) the following rule applies: all the declarations in the file must be `external`. By applying `@JsModule(...)` on a file you tell the compiler to import a JavaScript object that contain all the declarations from the file. By applying `@JsModule("jquery")` on a file you tell the compiler to import a JavaScript object that contain all the declarations from the file.

Example: `@JsModule("jquery")` external abstract class JQuery() {
    // some declarations here
}

`@JsModule("jquery")` external fun JQuery(element: Element): JQuery

`@property` import name of a module to import declaration from. To tell the Kotlin compiler to accept both cases, you can augment `@JsModule` with the `@JsNonModule` annotation. For example:

```kotlin
@JsModule("jquery")
@JsNonModule
external abstract class JQuery() {
    // some declarations here
}

@JsModule("jquery")
@JsNonModule
external fun JQuery(element: Element): JQuery
```

`@property` value the qualifier to add to the declarations in the generated code.

```kotlin
@file:JsQualifier("my.jsPackageName")
package some.kotlinPackage

external fun foo(x: Int)

external fun bar(): String
```

It must be a sequence of valid JavaScript identifiers separated by the `.` character. Examples of valid qualifiers are: `foo`, `bar.Baz`, `~$0.f`.

```kotlin
@file:JsQualifier("my.jsPackageName")
package some.kotlinPackage

external fun foo(x: Int)

external fun bar(): String
```

See also `JsModule`.

```kotlin
@file:JsQualifier("my.jsPackageName")
package some.kotlinPackage

external fun foo(x: Int)

external fun bar(): String
```

It must be a sequence of valid JavaScript identifiers separated by the `.` character.
invariant on its element type.\n */\npublic actual abstract class AbstractMutableCollection<E> protected actual
constructor() : AbstractCollection<E>(), MutableCollection<E> {\n\n actual abstract override fun add(element: E):
Boolean\n\n actual override fun remove(element: E): Boolean {\n
val iterator = iterator()\n
while
(iterator.hasNext()) {\n
if (iterator.next() == element) {\n
iterator.remove()\n
return true\n
}\n
}\n
return false\n }\n\n actual override fun addAll(elements: Collection<E>): Boolean {\n
var modified = false\n
for (element in elements) {\n
if (add(element)) modified = true\n
}\n
return modified\n }\n\n actual override fun removeAll(elements: Collection<E>): Boolean = (this as
MutableIterable<E>).removeAll { it in elements }\n actual override fun retainAll(elements: Collection<E>):
Boolean = (this as MutableIterable<E>).removeAll { it !in elements }\n\n actual override fun clear(): Unit {\n
val iterator = this.iterator()\n
while (iterator.hasNext()) {\n
iterator.next()\n
iterator.remove()\n
}\n }\n\n @JsName(\"toJSON\")\n open fun toJSON(): Any = this.toArray()\n}\n\n","/*\n * Copyright 20102018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n/*\n * Based on GWT AbstractList\n *
Copyright 2007 Google Inc.\n*/\n\n\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of
the [MutableList] interface.\n *\n * @param E the type of elements contained in the list. The list is invariant on its
element type.\n */\npublic actual abstract class AbstractMutableList<E> protected actual constructor() :
AbstractMutableCollection<E>(), MutableList<E> {\n protected var modCount: Int = 0\n\n abstract override fun
add(index: Int, element: E): Unit\n abstract override fun removeAt(index: Int): E\n abstract override fun
set(index: Int, element: E): E\n\n /**\n * Adds the specified element to the end of this list.\n *\n * @return
`true` because the list is always modified as the result of this operation.\n */\n actual override fun add(element:
E): Boolean {\n
add(size, element)\n
return true\n }\n\n actual override fun addAll(index: Int, elements:
Collection<E>): Boolean {\n
var _index = index\n
var changed = false\n
for (e in elements) {\n
add(_index++, e)\n
changed = true\n
}\n
return changed\n }\n\n actual override fun clear() {\n
removeRange(0, size)\n }\n\n actual override fun removeAll(elements: Collection<E>): Boolean = removeAll {
it in elements }\n actual override fun retainAll(elements: Collection<E>): Boolean = removeAll { it !in elements
}\n\n\n actual override fun iterator(): MutableIterator<E> = IteratorImpl()\n\n actual override fun
contains(element: E): Boolean = indexOf(element) >= 0\n\n actual override fun indexOf(element: E): Int {\n
for (index in 0..lastIndex) {\n
if (get(index) == element) {\n
return index\n
}\n
}\n
return -1\n }\n\n actual override fun lastIndexOf(element: E): Int {\n
for (index in lastIndex downTo 0) {\n
if (get(index) == element) {\n
return index\n
}\n
}\n
return -1\n }\n\n actual
override fun listIterator(): MutableListIterator<E> = listIterator(0)\n actual override fun listIterator(index: Int):
MutableListIterator<E> = ListIteratorImpl(index)\n\n\n actual override fun subList(fromIndex: Int, toIndex: Int):
MutableList<E> = SubList(this, fromIndex, toIndex)\n\n /**\n * Removes the range of elements from this list
starting from [fromIndex] and ending with but not including [toIndex].\n */\n protected open fun
removeRange(fromIndex: Int, toIndex: Int) {\n
val iterator = listIterator(fromIndex)\n
repeat(toIndex fromIndex) {\n
iterator.next()\n
iterator.remove()\n
}\n }\n\n /**\n * Compares this list with
another list instance with the ordered structural equality.\n *\n * @return true, if [other] instance is a [List] of
the same size, which contains the same elements in the same order.\n */\n override fun equals(other: Any?):
Boolean {\n
if (other === this) return true\n
if (other !is List<*>) return false\n\n
return
AbstractList.orderedEquals(this, other)\n }\n\n /**\n * Returns the hash code value for this list.\n */\n
override fun hashCode(): Int = AbstractList.orderedHashCode(this)\n\n\n private open inner class IteratorImpl :
MutableIterator<E> {\n
/** the index of the item that will be returned on the next call to [next]`()` */\n
protected var index = 0\n
/** the index of the item that was returned on the previous call to [next]`()`\n
* or
[ListIterator.previous]`()` (for `ListIterator`),\n
* -1 if no such item exists\n
*/\n
protected var last = 1\n\n
override fun hasNext(): Boolean = index < size\n\n
override fun next(): E {\n
if (!hasNext())
throw NoSuchElementException()\n
last = index++\n
return get(last)\n
}\n\n
override fun
remove() {\n
check(last != -1) { \"Call next() or previous() before removing element from the iterator.\" }\n\n
removeAt(last)\n
index = last\n
last = -1\n
}\n }\n\n /**\n * Implementation of

Open Source Used In DNAC Platform Antelope 4007


private inner class ListIteratorImpl(index: Int) : IteratorImpl(), MutableListIterator<E>

    init {
        AbstractList.checkPositionIndex(index, this@AbstractMutableList.size)
        override fun hasPrevious(): Boolean = index > 0
        override fun nextIndex(): Int = index
        override fun previous(): E {
            if (!hasPrevious()) throw NoSuchElementException()
            last = --index
            return get(last)
        }

        override fun previousIndex(): Int = index - 1
        override fun add(element: E) {
            add(index, element)
            index++
            last = -1
        }
    }

    private class SubList<E>(private val list: AbstractMutableList<E>, private val fromIndex: Int, toIndex: Int) : AbstractMutableList<E>(), RandomAccess {
        private var _size: Int = 0

        init {
            AbstractList.checkRangeIndexes(fromIndex, toIndex, list.size)
            this._size = toIndex - fromIndex
        }

        override fun add(index: Int, element: E) {
            AbstractList.checkPositionIndex(index, _size)
            list.add(fromIndex + index, element)
            _size++
        }

        override fun get(index: Int): E {
            AbstractList.checkElementIndex(index, _size)
            return list[fromIndex + index]
        }

        override fun removeAt(index: Int): E {
            AbstractList.checkElementIndex(index, _size)
            val result = list.removeAt(fromIndex + index)
            _size--
            return result
        }

        override fun set(index: Int, element: E): E {
            AbstractList.checkElementIndex(index, _size)
            return list.set(fromIndex + index, element)
        }

        override val size: Int get() = _size
    }

    actual override fun clear() {
        entries.clear()
    }

    private var _keys: MutableSet<K>? = null
    actual override val keys: MutableSet<K>
        get() {
            if (_keys == null) {
                _keys = object : AbstractMutableSet<K>() {
                    override fun add(element: K): Boolean = throw
                        UnsupportedOperationException("Add is not supported on keys")
                    override fun clear() {
                        this@AbstractMutableMap.clear()
                    }

                    override operator fun contains(element: K): Boolean = containsKey(element)

                    override fun remove(element: K): Boolean {
                        if (containsKey(element)) {
                            this@AbstractMutableMap.remove(element)
                            return true
                        }
                        return false
                    }

                    override fun values: MutableCollection<V>
                        get() {
                            if (containsKey(element)) {
                                this@AbstractMutableMap.remove(element)
                                return true
                            }
                            return false
                        }
                    override fun entries: AbstractMap.Entry<K, V>
                        get() {
                            return null
                        }
                    override fun remove(element: K): Boolean {
                        if (containsKey(element)) {
                            this@AbstractMutableMap.remove(element)
                            return true
                        }
                        return false
                    }
                    override fun put(key: K, value: V): V? {
                        return null
                    }
                    override fun size: Int get() = _size
                }
            }
            return _keys!!
        }

        actual override fun putAll(from: Map<out K, V>) {
            for ((key, value) in from) {
                put(key, value)
            }
        }

        private var _values: MutableCollection<V>? = null
        actual override val values: MutableCollection<V>
            get() {
                if (_values == null) {
                    _values = object :
AbstractMutableCollection<V>() {
  override fun add(element: V): Boolean = throw

  override fun clear() =
    this@AbstractMutableMap.clear()

  override operator fun contains(element: V): Boolean =
    containsValue(element)

  override operator fun iterator(): MutableIterator<V> {
    val entryIterator = entries.iterator()
    return object : MutableIterator<V> {
      override fun hasNext(): Boolean = entryIterator.hasNext()

      override fun next(): V = entryIterator.next().value

      override fun remove() = entryIterator.remove()
    }

  }

  override val size: Int get() = this@AbstractMutableMap.size

  // TODO: should we implement them this way? Currently it's unspecified in JVM
  override fun equals(other: Any?): Boolean {
    if (this === other) return true
    if (other !is Collection<*>) return false
    return AbstractList.orderedEquals(this, other)

  }

  override fun hashCode(): Int =
    AbstractList.orderedHashCode(this)
}

actual override

fun remove(key: K): V? {
  val iter = entries.iterator()
  while (iter.hasNext()) {
    val entry = iter.next()
    val k = entry.key
    if (key == k) {
      val value = entry.value
      iter.remove()
      return value
    }
  }
  return null
}

// © 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
// Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
package kotlin.collections

/**
 * Provides a skeletal implementation of the [MutableSet] interface.
 *
 * @param E the type of elements contained in the set. The set is invariant on its element type.
 */
public actual abstract class AbstractMutableSet<E> internal constructor() : AbstractMutableCollection<E>(), MutableSet<E> {

  /**
   * Compares this set with another set instance with the unordered structural equality.
   *
   * @return `true`, if [other] instance is a [Set] of the same size, all elements of which are contained in this set.
   */
  override fun equals(other: Any?): Boolean {
    if (this === other) return true
    if (other !is Set<*>) return false
    return AbstractSet.orderedEquals(this, other)

  }

  /**
   * Returns the hash code value for this set.
   */
  override fun hashCode(): Int =
    AbstractSet.unorderedHashCode(this)
}

// © 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
// Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
package kotlin.collections

/**
 * Provides a [MutableList] implementation, which uses a resizable array as its backing storage.
 *
 * This implementation doesn't provide a way to manage capacity, as backing JS array is resizeable itself. There is no speed advantage to pre-allocating array sizes in JavaScript, so this implementation does not include any of the capacity and \"growth increment\" concepts.
 */
public actual open class ArrayList<E> internal constructor(private var array: Array<Any?>) :
  AbstractMutableList<E>(), MutableList<E>, RandomAccess {

  public actual constructor() : this(emptyArray()) {}

  public actual constructor(initialCapacity: Int = 0) : this(emptyArray()) {}

  public actual constructor(elements: Collection<E>) : this(elements.toTypedArray<Any?>()) {}

  actual override val size: Int get() = array.size

  @Suppress("UNCHECKED_CAST")
  actual override fun get(index: Int): E =
    array[rangeCheck(index)] as E

  actual override fun set(index: Int, element: E): E {
    rangeCheck(index)

    @Suppress("UNCHECKED_CAST")
    return array[insertionRangeCheck(index)] as E

  }

  actual override fun add(element: E): Boolean =
    array.asDynamic().push(element)

  actual override fun add(index: Int, element: E) {
    array.asDynamic().splice(insertionRangeCheck(index), 0, element)

  }

  actual override fun addAll(elements: Collection<E>): Boolean =
    elements.toTypedArray<Any?>().forEach { array += it }

  actual override fun addAll(index: Int, elements: Collection<E>): Boolean =
    elements.toTypedArray<Any?>().forEach { array += it }

  @Suppress("UNCHECKED_CAST")
  actual override fun get(index: Int): E =
    array[rangeCheck(index)]

  @Suppress("UNCHECKED_CAST")
  actual override fun get(index: Int): E =
    array[rangeCheck(index)]

  @Suppress("UNCHECKED_CAST")
  actual override fun get(index: Int): E =
    array[rangeCheck(index)]
Int, elements: Collection<E>): Boolean {
    if (elements.isEmpty()) return false
    when (index) {
        size -> return
        0 -> array = array.copyOfRange(0, index).asDynamic().concat(elements, array.copyOfRange(index, size))
        else -> array = array.asDynamic().splice(index, 1)[0]
    }
    return true
}
return target
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.collections

@UseExperimental(ExperimentalUnsignedTypes::class)
@SinceKotlin("1.3")
@kotlin.js
@JsName("contentDeepHashCodeImpl")
internal fun <T> Array<out T>.contentDeepHashCodeImpl(): Int {
    var result = 1
    for (element in this) {
        val elementHash = when {
            element == null -> 0
            isArrayish(element) -> (element.unsafeCast<Array<*>>()).contentDeepHashCodeImpl()
            element is UByteArray   -> element.contentHashCode()
            element is UShortArray  -> element.contentHashCode()
            element is UIntArray    -> element.contentHashCode()
            element is ULongArray   -> element.contentHashCode()
            else                    -> element.hashCode()
        }
        result = 31 * result + elementHash
    }
    return result
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.collections

internal interface EqualityComparator {
    /**
     * Subclasses must override to return a value indicating
     * whether or not two keys or values are equal.
     */
    abstract fun equals(value1: Any?, value2: Any?): Boolean

    /**
     * Subclasses must override to return the
     * hash code of a given key.
     */
    abstract fun getHashCode(value: Any?): Int

    object HashCode :
        EqualityComparator {
            override fun equals(value1: Any?, value2: Any?): Boolean = value1 == value2

            override fun getHashCode(value: Any?): Int = value?.hashCode() ?: 0
        }
}
equality.equals(it.value, value) }

    private var _entries: MutableSet<MutableMap.MutableEntry<K, V>>? =
        null
    actual override val entries: MutableSet<MutableMap.MutableEntry<K, V>>
        get() {
        if (_entries == null) {
            _entries = createEntrySet()
        }
        return _entries!!
    }

    protected open fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> =
        EntrySet()

    actual override operator fun get(key: K): V? = internalMap.get(key)

    actual override fun put(key: K, value: V): V? =
        internalMap.put(key, value)

    actual override val size: Int get() = internalMap.size

/**
 * Constructs the specialized implementation of [HashMap] with [String] keys, which stores the keys as properties of
 * JS object without hashing them.
 */
actual fun stringMapOf(vararg pairs: Pair<String, V>): HashMap<String, V> {
    return HashMap<String, V>(InternalStringMap(EqualityComparator.HashCode)).apply {
        putAll(pairs)
    }
}

/**
 * Creates a new instance of the specialized implementation of [HashSet] with the specified [String] elements,
 * which elements the keys as properties of JS object without hashing them.
 */
actual fun stringSetOf(vararg elements: String): HashSet<String> {
    return HashSet(stringMapOf<Any>()).apply {
        addAll(elements)
    }
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
/*
 * Based on GWT InternalHashCodeMap
 * Copyright 2008 Google Inc.
 */

package kotlin.collections

import kotlin.collections.MutableMap.MutableEntry
import kotlin.collections.AbstractMutableMap.SimpleEntry

internal class InternalHashCodeMap<K, V>(override val equality: EqualityComparator) :
    InternalMap<K, V> {

    private var backingMap: dynamic = createJsMap()
    override var size: Int = 0
        private set

    override fun put(key: K, value: V): V? {
        val hashCode =
            equality.getHashCode(key)
        val chainOrEntry =
            getChainOrEntryOrNull(hashCode)
        if (chainOrEntry == null) {
            // This is a new chain, put it to the map.
            backingMap[hashCode] = SimpleEntry(key, value)
        } else {
            if (chainOrEntry is Array<*>) {
                val entry:
SimpleEntry<K, V> = chainOrEntry
    if (equality.equals(entry.key, key))
        return entry.setValue(value)
    else
        backingMap[hashCode] = arrayOf(entry, SimpleEntry(key, value))
        size++
        return null
    else
        // Chain already exists, perhaps key also exists.
        val chain: Array<MutableEntry<K, V>> = chainOrEntry
        val entry = chain.findEntryInChain(key)
        if (entry != null)
            return entry.setValue(value)
        else
            backingMap[hashCode] = arrayOf(entry, SimpleEntry(key, value))
            size++
            return null
    }

// structureChanged(host)
return null

val chainOrEntry = getChainOrEntryOrNull(hashCode) ?: return null
if (chainOrEntry !is Array<*>) {
    val entry: MutableEntry<K, V> = chainOrEntry
    if (equality.equals(entry.key, key))
        jsDeleteProperty(backingMap, hashCode)
        size--
        return entry.value
    else
        return null
}
else
    val chain: Array<MutableEntry<K, V>> = chainOrEntry
    for (index in chain.indices)
        if (equality.equals(key, entry.key))
            jsDeleteProperty(backingMap, hashCode)
        val chain[index] = entry
        size--
        return null
    val chain = chainOrEntry
    return null
}

chain.asDynamic().push(SimpleEntry(key, value))
size++
// structureChanged(host)
return null
val chainOrEntry = getChainOrEntryOrNull(hashCode) ?: return null
if (chainOrEntry !is Array<*>) {
    val entry: MutableEntry<K, V> = chainOrEntry
    if (equality.equals(entry.key, key))
        return entry.setValue(value)
    else
        backingMap[hashCode] = arrayOf(entry, SimpleEntry(key, value))
        size++
        return null
    }
else
    val chain: Array<MutableEntry<K, V>> = chainOrEntry
    return null
    }

chain.asDynamic().splice(index, 1)
size--
// structureChanged(host)
    return entry.value
    else
        return null
    }

// override fun clear()
    backingMap = createJsMap()
    size = 0

// override fun contains(key: K):
    return getEntry(key) != null

// override fun get(key: K): V? = getEntry(key)?.value

// private fun getEntry(key: K): MutableEntry<K, V>? {
    val chainOrEntry = getChainOrEntryOrNull(equality.getHashCode(key)) ?: return null
    if (chainOrEntry !is Array<*>) {
        val entry: MutableEntry<K, V> = chainOrEntry
        if (equality.equals(entry.key, key))
            return entry
        else
            return null
    } else {
        val chain: Array<MutableEntry<K, V>> = chainOrEntry
        return chain.findEntryInChain(key)
    }

// Array<MutableEntry<K, V>>.findEntryInChain(key: K):
    return firstOrNull { entry ->
        equality.equals(entry.key, key) }

// override fun iterator():
    return object : MutableIterator<MutableEntry<K, V>> {
        var state = -1 // -1 not ready, 0 - ready, 1 - done
        var chainOrEntry: dynamic = null
        var isChain = false
        var itemIndex = -1
        var lastEntry: MutableEntry<K, V>? = null

        private fun computeNext(): Int {
            if (chainOrEntry != null && isChain) {
                val chainSize = chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>().size
                if (++itemIndex < chainSize)
                    return 0
            }

            if (++keyIndex < keys.size)
                chainOrEntry = backingMap[keys[keyIndex]].unsafeCast<Array<MutableEntry<K, V>>>()
            else
                chainOrEntry = null
            itemIndex = 0
            return 0
        }

        override fun hasNext(): Boolean {
            if (state == -1)
                state = computeNext()
            return state == 0
        }

        override fun next(): MutableEntry<K, V> {
            if (!hasNext()) throw NoSuchElementException()
            val lastEntry = if (isChain) {
                chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>()[itemIndex]
            } else {
                chainOrEntry.unsafeCast<MutableEntry<K, V>>()
            }
            this.lastEntry = lastEntry
            state = -1
            return lastEntry
        }

        override fun remove() {
            checkNotNull(lastEntry)
            this@InternalHashCodeMap.remove(lastEntry!!.key)
            lastEntry = null
            return
        }

    }

// private fun getChainOrEntryOrNull(hashCode: Int): dynamic {
    return backingMap[hashCode]
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

/**
 * The common interface of InternalStringMap and InternalHashCodeMap.
 */
internal interface InternalMap<K, V> :
MutableIterable<MutableMap.MutableEntry<K, V>> {
    val equality: EqualityComparator
    val size: Int

    operator fun contains(key: K): Boolean
    operator fun get(key: K): V?

    fun put(key: K, value: V): V?
    fun remove(key: K): V?
    fun clear(): Unit

    fun createJsMap(): dynamic {
        val result = js("Object.create(null)")
        // force to switch object representation to dictionary mode
        result["foo"] = 1
        jsDeleteProperty(result, "foo")
        return result
    }
}

package kotlin.collections

import kotlin.collections.MutableMap.MutableEntry

/**
 * A simple wrapper around JavaScript Map for key type is string.
 * Though this map is instantiated only with K=String, the K type is not fixed to String statically,
 * because we want to have it erased to Any? in order not to generate type-safe override bridges for
 * [get], [contains], [remove] etc, if they ever are generated.
 */
class InternalStringMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {

    private var backingMap: dynamic = createJsMap()
    override var size: Int = 0

    override operator fun contains(key: K): Boolean {
        if (key !is String) return false
        return backingMap[key] !== undefined
    }

    override operator fun get(key: K): V? {
        if (key !is String) return null
        val value = backingMap[key]
        return if (value !== undefined) value.unsafeCast<V>() else null
    }

    override fun put(key: K, value: V): V? {
        require(key is String)
        val oldValue = backingMap[key]
        backingMap[key] = value
        if (oldValue === undefined) {
            size++
            return null
        } else {
            return oldValue.unsafeCast<V>()
        }
    }

    override fun remove(key: K): V? {
        if (key !is String) return null
        val value = backingMap[key]
        if (value !== undefined) {
            jsDeleteProperty(backingMap, key)
            size--
            return value.unsafeCast<V>()
        } else {
            return null
        }
    }

    override fun clear() {
        backingMap = createJsMap()
        size = 0
    }

    override fun iterator(): MutableIterator<MutableEntry<K, V>> {
        return object : MutableIterator<MutableEntry<K, V>> {
            private val keys: Array<String> = js("Object").keys(backingMap)
            private val iterator = keys.iterator()
            private var lastKey: String? = null

            override fun hasNext(): Boolean = iterator.hasNext()

            override fun next(): MutableEntry<K, V> {
                val key = iterator.next()
                lastKey = key
                @Suppress("UNCHECKED_CAST")
                return newMapEntry(key as K)
            }

            override fun remove() {
                @Suppress("UNCHECKED_CAST")
                this@InternalStringMap.remove(checkNotNull(lastKey) as K)
            }
        }
    }

    private fun newMapEntry(key: K): MutableEntry<K, V> = object : MutableEntry<K, V> {
        override val key: K get() = key
        override val value: V get() = this@InternalStringMap[key].unsafeCast<V>()

        override fun setValue(newValue: V): V = this@InternalStringMap.put(key, newValue).unsafeCast<V>()
        override fun hashCode(): Int = AbstractMap.entryHashCode(this)
        override fun equals(other: Any?): Boolean = AbstractMap.entryEquals(this, other)
    }
}

Open Source Used In DNAC Platform Antelope 4014
underlying hash map so we can find\n * the eldest entry. The alternative would have been to modify HashMap so
 more\n * of the code was directly usable here, but this would have added some
 * overhead to HashMap, or to
 reimplementation most of the HashMap code here within small modifications. Paying a small storage cost only if
 you use\n * LinkedHashMap and minimizing code size seemed like a better tradeoff\n * in private class
 ChainEntry<K, V>(key: K, value: V) : AbstractMutableMap.SimpleEntry<K, V>(key, value) {\n \n \n \n \n \n }\n \n \n } under internal var next: ChainEntry<K, V>? = null\n \n \n \n }\n \n \n } \n \n \n } internal class EntryIterator : MutatableIterator<MutableEntry<K, V>> {\n \n \n \n }\n \n \n } \n \n \n } override fun clear() {\n \n super.clear() \n \n \n \n \n }\n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n } \n \n \n }\n
}
override fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()

    actual override fun get(key: K): V? = map.get(key)

    actual override fun containsKey(key: K): Boolean {
        var node: ChainEntry<K, V> = head
        while (node != null) {
            if (node.key == key) {
                return true
            }
            node = node.next
        }
        return false
    }

    actual override fun containsValue(value: V): Boolean {
        var node: ChainEntry<K, V> = head
        do {
            if (node.value == value) {
                return true
            }
            node = node.next
        } while (node != head)
        return false
    }

    override fun createEntrySet(): Collection<MutableMap.MutableEntry<K, V>> = EntrySet()

    actual override operator fun get(key: K): V? = map.get(key)
String(message)
        outputStream.write(messageString)
    }
}

/** JsName used to make the declaration available outside of module to test it */
@JsName("OutputToConsoleLog")
internal class OutputToConsoleLog

    override fun print(message: Any?) {
        console.log(message)
    }

    override fun println(message: Any?) {
        console.log(message +"")
        buffer += String(message)
    }

    override fun println() {
        console.log("")
    }

/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in */
@JsName("BufferedOutput")
internal open class BufferedOutput : BaseOutput() {
    var buffer = ""

    override fun print(message: Any?) {
        buffer += String(message)
    }

    override fun flush() {
        buffer = ""
    }
}

/** JsName used to make the declaration available outside of module to test it */
@JsName("BufferedOutputToConsoleLog")
internal class BufferedOutputToConsoleLog : BufferedOutput() {
    override fun print(message: Any?) {
        var s = String(message) val i = s.lastIndexOf('"
') if (i > 0) {
            buffer += s.substring(0, i)
            println(s.substring(i + 1))
            buffer += s
        }
    }

    override fun flush() {
        console.log(buffer)
        buffer = ""
    }
}

/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in */
@JsName("output")
internal var output = run {
    val isNode: Boolean = js("typeof process !== 'undefined' && process.versions &&
    !!process.versions.node")
    if (isNode) NodeJsOutput(js("process.stdout")) else
        BufferedOutputToConsoleLog()
}

@kotlin.internal.InlineOnly
private inline fun String(value: Any?): String =
    js("String")(value)

/** Prints the line separator to the standard output stream. */
public actual fun println() {
    output.println()
}

/** Prints the given [message] and the line separator to the standard output stream. */
public actual fun println(message: Any?) {
    output.println(message)
}

/** Prints the given [message] to the standard output stream. */
public actual fun print(message: Any?) {
    output.print(message)
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.coroutines

import kotlin.coroutines.intrinsics.CoroutineSingletons.*
import kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED

@PublishedApi
@SinceKotlin("1.3")
internal actual class SafeContinuation<in T>
internal actual constructor(
    delegate: Continuation<T>,
    initialResult: Any?
) : Continuation<T> {
    @PublishedApi
    internal actual constructor(delegate: Continuation<T>) : this(delegate, UNDECIDED)

    public actual override val context: CoroutineContext
        get() = delegate.context

    private var result: Any? = initialResult

    public actual override fun resumeWith(result: Result<T>) {
        val cur = this.result
        when {
            cur === UNDECIDED -> {
                this.result = result.value
            }
            cur === COROUTINE_SUSPENDED -> {
                this.result = RESUMED
                delegate.resumeWith(result)
            }
            else -> throw IllegalStateException("Already resumed")
        }
    }

    internal actual fun getOrThrow(): Any? {
    if (result === UNDECIDED) {
        result = COROUTINE_SUSPENDED
    return COROUTINE_SUSPENDED
            val result = this.result
        return when {
            result === RESUMED
        delegate.resumeWith(result)
        }
    else -> throw
    throw
    }

    return COROUTINE_SUSPENDED // already called continuation, indicate COROUTINE_SUSPENDED
    upstream
    result is Result.Failure -> throw result.exception
    else -> result // either
}

package kotlin.coroutines.js.internal

import kotlin.coroutines.Continuation
import kotlin.coroutines.EmptyCoroutineContext

@PublishedApi
@SinceKotlin("1.3")
internal val EmptyContinuation = Continuation<Any?>(EmptyCoroutineContext) { result ->
    result.getOrThrow()
}

package kotlin.js

/**
 * Exposes the [Date API](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date) to Kotlin.
 */
@Suppress("NOT_DOCUMENTED")
public external class Date() {
    public constructor(milliseconds: Number)

    public constructor(dateString: String)

    public constructor(year: Int, month: Int)

    public constructor(year: Int, month: Int, day: Int)

    public
return true

} return false

```kotlin
/**
 * Converts the string into a regular expression [Regex] with the default options.
 */
@JvmMultifileClass
@JvmName("StringsKt")
package kotlin

/**
 * Converts the string into a regular expression [Regex] with the specified single [option].
 */
public fun Element.removeClass(vararg cssClasses: String): Boolean {
    return cssClasses.any { hasClass(it) }
}

/**
 * Removes all [cssClasses] from element. Has no effect if all specified classes are missing in class attribute of the element
 */
public fun Element.removeClass(vararg cssClasses: String): Boolean {
    val toBeRemoved = cssClasses.toSet().filter { it !in toBeRemoved }.map { toClass(it) }
    className = className.trim().split(\\s+\+.toRegex()).filter { it !in toBeRemoved }.joinToString(\\")
    return true
}

/**
 * Builds new string by populating newly created [StringBuilder] using provided [builderAction] and then converting it to [String].
 */
public fun StringBuilder.append(vararg value: String?): StringBuilder {
    for (item in value)
        append(item)
    return this
}

/**
 * Builds new string by populating newly created [StringBuilder] initialized with the given [capacity] using provided [builderAction] and then converting it to [String].
 */
public fun StringBuilder.append(vararg value: Any?): StringBuilder {
    for (item in value)
        append(item)
    return this
}

internal fun <T> Appendable.appendElement(element: T, transform: ((T) -> CharSequence)?) {
    when {
        transform != null -> append(transform(element))
        element is CharSequence? -> append(element)
        element is Char -> append(element)
        else -> append(element.toString())
    }
}
```

---

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

`@file:kotlin.jvm.JvmMultifileClass`
`@file:kotlin.jvm.JvmName("StringsKt")`
```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin

/**
 * Converts the string into a regular expression [Regex] with the default options.
 */
public fun Element.removeClass(vararg cssClasses: String): Boolean {
    return cssClasses.any { hasClass(it) }
}

/**
 * Removes all [cssClasses] from element. Has no effect if all specified classes are missing in class attribute of the element
 */
public fun Element.removeClass(vararg cssClasses: String): Boolean {
    val toBeRemoved = cssClasses.toSet().filter { it !in toBeRemoved }.map { toClass(it) }
    className = className.trim().split(\\s+\+.toRegex()).filter { it !in toBeRemoved }.joinToString(\\")
    return true
}

/**
 * Builds new string by populating newly created [StringBuilder] using provided [builderAction] and then converting it to [String].
 */
public fun StringBuilder.append(vararg value: String?): StringBuilder {
    for (item in value)
        append(item)
    return this
}

/**
 * Builds new string by populating newly created [StringBuilder] initialized with the given [capacity] using provided [builderAction] and then converting it to [String].
 */
public fun StringBuilder.append(vararg value: Any?): StringBuilder {
    for (item in value)
        append(item)
    return this
}

internal fun <T> Appendable.appendElement(element: T, transform: ((T) -> CharSequence)?) {
    when {
        transform != null -> append(transform(element))
        element is CharSequence? -> append(element)
        element is Char -> append(element)
        else -> append(element.toString())
    }
}
```

---

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.w3c.dom

import org.w3c.dom.*

/**
 * Gets a value indicating whether this node is a TEXT_NODE or a CDATA_SECTION_NODE.
 */
public val Node.isText: Boolean
    get() = nodeType == Node.TEXT_NODE || nodeType == Node.CDATA_SECTION_NODE

/**
 * Gets a value indicating whether this node is an [Element].
 */
public val Node.isElement: Boolean
    get() = nodeType == Node.ELEMENT_NODE

```java
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.w3c.dom.events

public fun EventListener(handler: (Event) -> Unit): EventListener = EventListenerHandler(handler)

private class EventListenerHandler(private val handler: (Event) -> Unit) : EventListener {
    public override fun handleEvent(event: Event) {
        handler(event)
    }

    public override fun toString(): String = "EventListenerHandler($handler)"
}
```

---

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.w3c.dom

public
external interface ItemArrayLike<T> {
    val length: Int
    fun item(index: Int): T
}

/**
 * Returns the view of this `ItemArrayLike<T>` collection as `List<T>`
 */
operator fun <T> ItemArrayLike<T>.asList(): List<T> = object : AbstractList<T>() {
    override val size: Int get() = this@asList.length

    override fun get(index: Int): T = when (index) {
        in 0..lastIndex -> this@asList.item(index).unsafeCast<T>()
        else -> throw IndexOutOfBoundsException("index $index is not in range [0..$lastIndex]"
    }
}
element to get its key. A [Grouping] structure serves as an intermediate step in group-and-fold operations:

* they group elements by their keys and then fold each group with some aggregating operation.

* It is created by attaching `keySelector: (T) -> K` function to a source of elements.

* To get an instance of [Grouping] use one of `groupingBy` extension functions:

  * `- [Iterable.groupingBy]`
  * `- [Sequence.groupingBy]`

  * `- [Array.groupingBy]`
  * `- [CharSequence.groupingBy]`

* For the list of group-and-fold operations available, see the [extension functions](#extension-functions) for `Grouping`.

```kotlin
/**
 * @param operation function is invoked on each element with the following parameters:
 *  - `key`: the key of the group this element belongs to;
 *  - `accumulator`: the current value of the accumulator of the group, can be `null` if it's the first `element` encountered in the group;
 *  - `element`: the element from the source being aggregated;
 *  - `first`: indicates whether it's the first `element` encountered in the group.
 * @return a [Map] associating the key of each group with the result of aggregation of the group elements.
 * @sample samples.collections.Grouping.aggregateByRadix
 */
@SinceKotlin("1.1")
public inline fun <T, K, R>
    Grouping<T, K>.aggregate(
        operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R
    ): Map<K, R> {
    return aggregateTo(mutableMapOf<K, R>(), operation)
}
```

```kotlin
/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially.
 * The key for each element is provided by the [Grouping.keyOf] function.
 * @param operation function is invoked on each element with the following parameters:
 *  - `key`: the key of the group this element belongs to;
 *  - `accumulator`: the current value of the accumulator of the group;
 *  - `element`: the element from the source being aggregated;
 *  - `first`: indicates whether it's the first `element` encountered in the group.
 * @return a [Map] associating the key of each group with the result of aggregation of the group elements.
 * @sample samples.collections.Grouping.aggregateByRadixTo
 */
@SinceKotlin("1.1")
public inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.aggregateTo(
    destination: M,
    operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R
): M {
    for (e in this.sourceIterator()) {
        val key = keyOf(e)
        val accumulator = destination[key] as?
        destination[key] = operation(key, accumulator, e, accumulator == null && !destination.containsKey(key))
    }
    return destination
}
```

```kotlin
/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially.
 * An initial value of accumulator is provided by [initialValueSelector] function.
 * @param initialValueSelector a function that provides an initial value of accumulator for each group.
 *  It's invoked with parameters:
 *  - `key`: the key of the group this element belongs to;
 *  - `element`: the first element being encountered in that group.
 * @param operation a function that is invoked on each element with the following parameters:
 *  - `key`: the key of the group this element belongs to;
 *  - `accumulator`: the current value of the accumulator of the group;
 *  - `element`: the element from the source being accumulated.
 * @return a [Map] associating the key of each group with the result of accumulating the group elements.
 * @sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue
 */
@SinceKotlin("1.1")
public inline fun <T, K, R> Grouping<T, K>.fold(
    initialValueSelector: (key: K, element: T) -> R,
    operation: (key: K, accumulator: R, element: T) -> R
): Map<K, R> =
    @Suppress("UNCHECKED_CAST") aggregate { key, acc, e -> operation(key, if (first)
        initialValueSelector(key, e) else acc as R, e) }
```

---

Open Source Used In DNAC Platform Antelope 4022
current element as arguments,\n* and stores the results in the given [destination] map.\n* An initial value of accumulator is provided by [initialValueSelector] function.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group;\n* - 'element': the first element being encountered in that group.\n* If the [destination] map already has a value corresponding to some key, that value is used as an initial value of the accumulator for that group and the [initialValueSelector] function is not called for that group.\n* - 'accumulator': the current value of the accumulator of the group.\n* - 'element': the element from the source being accumulated.\n* @return the [destination] map associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return the [destination] map associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return the [destination] map associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.\n* @param initialValueSelector a function that provides an initial value of accumulator for each group.\n* It's invoked with parameters:\n* - 'key': the key of the group this element belongs to;\n* - 'element': the element from the source being accumulated.\n* @return a [Map] associating the key of each group with the result of accumulating the group elements.
sequentially starting from the second element of the group.\n
* passing the previously accumulated value and the current element as arguments;\n
* and stores the results in the given \[destination\] map.\n
* An initial value of accumulator is the first element of the group.\n
* If the \[destination\] map already has a value corresponding to the key of some group,\n
* that value is used as an initial value of the accumulator for that group and the first element of that group is also not subjected to the \[operation\].\n
* @param \[operation\] a function that is invoked on each subsequent element of the group with the following parameters: \n
*  - \[key\] \n
*  - \[accumulator\]: the element from the source being folded; \n
*  - \[element\].\n
* \n
* @return the \[destination\] map associating the key of each group with the result of accumulating the group elements.\n
* \n
* @sample samples.collections.Grouping.reduceByMaxVowelsTo\n
* @SinceKotlin("1.1")\n
* public inline fun <S, T : K, M : MutableMap<K, S>> Grouping<T, K>.reduceTo(\n
*    destination: M,\n
*    operation: (key: K, accumulator: S, element: T) -> S)\n
* @Suppress("UNCHECKED_CAST")\n
*    if (first) e else operation(key, acc as S, e)\n
* \n
* Groups elements from the \[Grouping\] source by key and counts elements in each group to the given \[destination\] map.\n
* \n
* @param \[valueSelector\] function for elements in each group.\n
* \n
* @return the \[destination\] map associating the key of each group with the sum of elements in the group.\n
* @sample samples.collections.Grouping.groupingByEachCount\n
* @SinceKotlin("1.1")\n
* public fun <T, K, M : MutableMap<K, Int>> Grouping<T, K>.eachCountTo(destination: M): M =

* \n
* Groups elements from the \[Grouping\] source by key and sums values provided by the \[valueSelector\] function for elements in each group.\n
* \n
* @return the \[destination\] map associating the key of each group with the sum of elements in the group.\n
* @sample samples.collections.Grouping.groupingByEachSumOf\n
* @SinceKotlin("1.1")\n
* public inline fun <T, K, M :.MutableMap<K, Int>> Grouping<T, K>.eachSumOfTo(destination: M, valueSelector: (T) -> Int): M =

* \n
* Groups elements from the \[Grouping\] source by key and sums values provided by the \[valueSelector\] function for elements in each group.\n
* \n
* @return the \[destination\] map associating the key of each group with the sum of elements in the group.\n
* @sample samples.collections.Grouping.groupingByEachDoubleSumOf\n
* @SinceKotlin("1.1")\n
* public inline fun <T, K, M :MutableMap<K, Double>> Grouping<T, K>.eachDoubleSumOfTo(destination: M, valueSelector: (T) -> Double): M =

* \n
* \n
* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
* Exposes the JavaScript \[Math object\](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Math) to Kotlin.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
* Exposes the JavaScript \[Math object\](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Math) to Kotlin.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
atan2(y: Double, x: Double): Double
    @Deprecated("Use kotlin.math.cos instead.", ReplaceWith("\"cos\"(value)\", \"kotlin.math.cos\"), level = DeprecationLevel.ERROR)\n    public fun cos(value: Double): Double
    @Deprecated("Use kotlin.math.sin instead.\", ReplaceWith("\"sin\"(value)\", \"kotlin.math.sin\"), level = DeprecationLevel.ERROR)\n    public fun sin(value: Double): Double
    @Deprecated("Use kotlin.math.exp instead.\", ReplaceWith("\"exp\"(value)\", \"kotlin.math.exp\"), level = DeprecationLevel.ERROR)\n    public fun exp(value: Double): Double
    @Deprecated("Use kotlin.math.max instead", level = DeprecationLevel.ERROR)
    public fun max(vararg values: Int): Int\n    @Deprecated("Use kotlin.math.max instead", level = DeprecationLevel.ERROR)
    public fun max(vararg values: Float): Float\n    @Deprecated("Use kotlin.math.max instead", level = DeprecationLevel.ERROR)
    public fun max(vararg values: Double): Double
    @Deprecated("Use kotlin.math.min instead.", ReplaceWith("\"min\"(value)\", \"kotlin.math.min\"), level = DeprecationLevel.ERROR)
    public fun min(vararg values: Int): Int
    @Deprecated("Use kotlin.math.min instead", level = DeprecationLevel.ERROR)
    public fun min(vararg values: Float): Float
    @Deprecated("Use kotlin.math.min instead", level = DeprecationLevel.ERROR)
    public fun min(vararg values: Double): Double
    @Deprecated("Use kotlin.math.sqrt instead.", ReplaceWith("\"sqrt\"(value)\", \"kotlin.math.sqrt\"), level = DeprecationLevel.ERROR)
    public fun sqrt(value: Double): Double
    @Deprecated("Use kotlin.math.tan instead.", ReplaceWith("\"tan\"(value)\", \"kotlin.math.tan\"), level = DeprecationLevel.ERROR)
    public fun tan(value: Double): Double
    @Deprecated("Use kotlin.math.ln instead.", ReplaceWith("\"ln\"(value)\", \"kotlin.math.ln\"), level = DeprecationLevel.ERROR)
    public fun log(value: Double): Double
    @Deprecated("Use kotlin.math.pow instead.", ReplaceWith("\"\"base\".pow(exp)\", \"kotlin.math.pow\"), level = DeprecationLevel.ERROR)
    public fun pow(base: Double, exp: Double): Double
    @Deprecated("Use kotlin.math.round instead.", ReplaceWith("\"round\"(value)\", \"kotlin.math.round\"), level = DeprecationLevel.ERROR)
    public fun round(value: Number): Int
    @Deprecated("Use kotlin.math.floor instead.", ReplaceWith("\"floor\"(value)\", \"kotlin.math.floor\"), level = DeprecationLevel.ERROR)
    public fun floor(value: Number): Int
    @Deprecated("Use kotlin.math.ceil instead.", ReplaceWith("\"ceil\"(value)\", \"kotlin.math.ceil\"), level = DeprecationLevel.ERROR)
    public fun ceil(value: Number): Int

    @PublishedApi
    internal fun trunc(value: Number): Double
    @PublishedApi
    internal fun sign(value: Number): Double

    @PublishedApi
    internal fun sinh(value: Double): Double
    @PublishedApi
    internal fun cosh(value: Double): Double
    @PublishedApi
    internal fun tanh(value: Double): Double
    @PublishedApi
    internal fun asinh(value: Double): Double
    @PublishedApi
    internal fun acosh(value: Double): Double
    @PublishedApi
    internal fun atanh(value: Double): Double

    @PublishedApi
    internal fun hypot(x: Double, y: Double): Double
    @PublishedApi
    internal fun expm1(value: Double): Double
    @PublishedApi
    internal fun log10(value: Double): Double
    @PublishedApi
    internal fun log2(value: Double): Double
    @PublishedApi
    internal fun log1p(value: Double): Double

    @PublishedApi
    internal fun clz32(value: Int): Int

/**
 * Returns the smaller of two values.
 */
@Suppress("DEPRECATION")
@Deprecated("Use minOf or kotlin.math.min instead", ReplaceWith("\"min\"(a, b)\", \"kotlin.math.min\"), level = DeprecationLevel.ERROR)
public fun Math.min(a: Long, b: Long): Long = if (a <= b) a else b

/**
 * Returns the greater of two values.
 */
@Suppress("DEPRECATION")
@Deprecated("Use maxOf or kotlin.math.max instead", ReplaceWith("\"max\"(a, b)\", \"kotlin.math.max\"), level = DeprecationLevel.ERROR)
public fun Math.max(a: Long, b: Long): Long = if (a >= b) a else b

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * LICENSE.txt file.
 * @package kotlin.js
 * **/
of the function will be translated to an assignment of [value] to the receiver indexed (with square brackets/index operation) with [propertyName].

```kotlin
fun test(j: Json, p: String, newValue: Any) {
    j["prop"] = 1
    j.set(p, newValue)
}
```

The following code:

```kotlin
fun test(j: Json, p: String, newValue: Any) {
    j["prop"] = 1
    j[p] = newValue
}
```

will be generated:

```js
function test(j, p, newValue) {
    j["prop"] = 1;
    j[p] = newValue;
}
```

### Public Functions

- `public fun json(vararg pairs: Pair<String, Any?>): Json`:

```kotlin
public fun json(vararg pairs: Pair<String, Any?>): Json {
    val res: dynamic = js("({})")
    for ((name, value) in pairs) {
        res[name] = value
    }
    return res
}
```

- `public fun Json.add(other: Json): Json`:

```kotlin
public fun Json.add(other: Json): Json {
    val keys: Array<String> = js("Object").keys(other)
    for (key in keys) {
        if (other.asDynamic().hasOwnProperty(key)) {
            this[key] = other[key];
        }
    }
    return this
}
```

### External Objects

- `public external object JSON {
    public fun stringify(o: Any?): String
    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)?): String
    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? = definedExternally, space: Int): String
    public fun stringify(o: Any?, replacer: Array<String>): String
    public fun stringify(o: Any?, replacer: Array<String>, space: Int): String
    public fun stringify(o: Any?, replacer: Array<String>, space: String): String
    public fun <T> parse(text: String): T
    public fun <T> parse(text: String, reviver: ((key: String, value: Any?) -> Any?)?): T
}
```

---

`Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.`
`sinh(NaN)` is `NaN`  
- `sinh(+Inf)` is `+Inf`  
- `sinh(-Inf)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sinh(x: Double): Double = 
nativeMath.sinh(x)

/**
 * Computes the hyperbolic cosine of the value [x].\n * Special cases:\n * - `cosh(NaN)` is `NaN`  
- `cosh(+Inf)` is `+Inf`  
- `cosh(-Inf)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun cosh(x: Double): Double = nativeMath.cosh(x)

/**
 * Computes the hyperbolic tangent of the value [x].\n * Special cases:\n * - `tanh(NaN)` is `NaN`  
- `tanh(+Inf)` is `1.0`  
- `tanh(-Inf)` is `-1.0`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun tanh(x: Double): Double = nativeMath.tanh(x)

/**
 * Computes the inverse hyperbolic sine of the value [x].\n * Special cases:\n * - `asinh(NaN)` is `NaN`  
- `asinh(+Inf)` is `+Inf`  
- `asinh(-Inf)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun asinh(x: Double): Double = 
nativeMath.asinh(x)

/**
 * Computes the inverse hyperbolic cosine of the value [x].\n * Special cases:\n * - `acosh(NaN)` is `NaN`  
- `acosh(x)` is `NaN` when `x < 1`  
- `acosh(+Inf)` is `+Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun acosh(x: Double): Double = 
nativeMath.acosh(x)

/**
 * Computes the inverse hyperbolic tangent of the value [x].\n * Special cases:\n * - `tanh(NaN)` is `NaN`  
- `tanh(x)` is `NaN` when `x > 1` or `x < -1`  
- `tanh(1.0)` is `1.0`  
- `tanh(-1.0)` is `-1.0`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun atanh(x: Double): Double = 
nativeMath.atanh(x)

/**
 * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.\n * Special cases:\n * - returns `+Inf` if any of arguments is infinite  
- returns `NaN` if any of arguments is `NaN`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun hypot(x: Double, y: Double): Double = 
nativeMath.hypot(x, y)

/**
 * Computes the positive square root of the value [x].\n * Special cases:\n * - `sqrt(x)` is `NaN` when `x < 0`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sqrt(x: Double): Double = 
nativeMath.sqrt(x)

/**
 * Computes Euler's number `e` raised to the power of the value [x].\n * Special cases:\n * - `exp(NaN)` is `NaN`  
- `exp(+Inf)` is `+Inf`  
- `exp(-Inf)` is `0.0`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun exp(x: Double): Double = 
nativeMath.exp(x)

/**
 * Computes `exp(x) - 1`.
 * Special cases:\n * - `expm1(NaN)` is `NaN`  
- `expm1(+Inf)` is `+Inf`  
- `expm1(-Inf)` is `-1.0`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun expm1(x: Double): Double = 
nativeMath.expm1(x)

/**
 * Computes the natural logarithm (base `E`) of the value [x].\n * Special cases:\n * - `ln(NaN)` is `NaN`  
- `ln(x)` is `NaN` when `x > 0`  
- `ln(+Inf)` is `+Inf`  
- `ln(0.0)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun ln(x: Double): Double = 
nativeMath.log(x)

/**
 * Computes the common logarithm (base 10) of the value [x].\n * Special cases:\n * - `log10(NaN)` is `NaN`  
- `log10(x)` is `NaN` when `x < 0`  
- `log10(+Inf)` is `+Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log10(x: Double): Double = 
nativeMath.log10(x)

/**
 * Computes the binary logarithm (base 2) of the value [x].\n * Special cases:\n * - `log2(NaN)` is `NaN`  
- `log2(x)` is `NaN` when `x < 0`  
- `log2(+Inf)` is `+Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log2(x: Double): Double = 
nativeMath.log2(x)

/**
 * Computes `ln1p(x)`.
 * Special cases:\n * - `ln1p(NaN)` is `NaN`  
- `ln1p(x)` is `NaN` when `x < -1`  
- `ln1p(+Inf)` is `+Inf`  
- `ln1p(-Inf)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun ln1p(x: Double): Double = 
nativeMath.log1p(x)

/**
 * Rounds the given value [x] to an integer towards positive infinity.\n * Special cases:\n * - `round(x)` is `NaN` when `x \neq round(x)`  
- `round(NaN)` is `NaN`  
- `round(-Inf)` is `-Inf`  
- `round(+Inf)` is `+Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun round(x: Double): Double = 
nativeMath.round(x)

/**
 * Returns the smallest integer that is greater than or equal to the given value [x] and is a mathematical integer.\n * Special cases:\n * - `ceil(NaN)` is `NaN`  
- `ceil(x)` is `NaN` when `x < 0`  
- `ceil(+Inf)` is `+Inf`  
- `ceil(-Inf)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun ceil(x: Double): Double = 
nativeMath.ceil(x)

/**
 * Returns the largest integer that is less than or equal to the given value [x] and is a mathematical integer.\n * Special cases:\n * - `floor(NaN)` is `NaN`  
- `floor(x)` is `NaN` when `x > 0`  
- `floor(+Inf)` is `+Inf`  
- `floor(-Inf)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun floor(x: Double): Double = 
nativeMath.floor(x)

/**
 * Returns the nearest integer to the given value [x].\n * Special cases:\n * - `nearest(NaN)` is `NaN`  
- `nearest(x)` is `NaN` when `x \neq nearest(x)`  
- `nearest(+Inf)` is `+Inf`  
- `nearest(-Inf)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun nearest(x: Double): Double = 
nativeMath.nearest(x)

/**
 * Returns the smallest integer that is greater than the given value [x] and is a mathematical integer.\n * Special cases:\n * - `trunc(NaN)` is `NaN`  
- `trunc(x)` is `NaN` when `x \neq trunc(x)`  
- `trunc(+Inf)` is `+Inf`  
- `trunc(-Inf)` is `-Inf`  

*/
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun trunc(x: Double): Double = 
nativeMath.trunc(x)
Special cases:

- `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

Rounds the given value `x` to an integer towards negative infinity.

- `return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.`

- `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

- `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

- `abs(NaN)` is `NaN`

- `sign(NaN)` is `NaN`
Double.NEGATIVE_INFINITY -> this\n this == 0.0 -> -Double.MIN_VALUE\n else ->
Double.fromBits(this.toRawBits() + if (this > 0) -1 else 1)\n
Double.nextTowards(to: Double): Double = when (\n  this == NaN() || to.isNaN()) -> Double.NaN\n  to == this -> this\n  to > this -> this.nextUp()\n  to < this -> this.nextDown()\n\n/**
 * Returns the [Double] value nearest to this
 * value in direction from this value towards the value [to].\n \n * Special cases:\n * - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`\n * - `x.nextTowards(x) == x`\n
\n/**
 * Returns the [Double] value nearest to the nearest integer and converts the result to [Int].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `round(x)` is `NaN` if `x` is `NaN`\n
\n/**
 * Rounds this [Double] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `round(x)` is `NaN` if `x` is `NaN`\n
\n/**
 * Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond to the rectangular coordinates `(x, y)` by computing the arc tangent of the value `y / x`.\n * The returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n * Special cases:\n * - `atan2(0.0, 0.0)` is `0.0`\n
\n/**
 * Computes the sine of the angle [x] given in radians.\n * Special cases:\n * - `sin(NaN)` is `NaN`\n
\n/**
 * Computes the cosine of the angle [x] given in radians.\n * Special cases:\n * - `cos(NaN)` is `NaN`\n
\n/**
 * Computes the tangent of the angle [x] given in radians.\n * Special cases:\n * - `tan(NaN)` is `NaN`\n
\n/**
 * Computes the arc sine of the value [x] given in radians.\n * Special cases:\n * - `asin(NaN)` is `NaN`\n
\n/**
 * Computes the arc cosine of the value [x] given in radians.\n * Special cases:\n * - `acos(NaN)` is `NaN`\n
\n/**
 * Computes the arc tangent of the value [x] given in radians.\n * Special cases:\n * - `atan(NaN)` is `NaN`\n
\n/
nativeMath.sinh(x.toDouble()).toFloat()\n
### Open Source Used In DNAC Platform Antelope 4030

** `Math.sinh(x)`**: Computes the hyperbolic sine of the value \(x\).

** `Math.cosh(x)`**: Computes the hyperbolic cosine of the value \(x\).

** `Math.tanh(x)`**: Computes the hyperbolic tangent of the value \(x\).

** `Math.asinh(x)`**: Computes the inverse hyperbolic sine of the value \(x\).

** `Math.acosh(x)`**: Computes the inverse hyperbolic cosine of the value \(x\).

** `Math.atanh(x)`**: Computes the inverse hyperbolic tangent of the value \(x\).

** `Math.hypot(x, y)`**: Computes \(\sqrt{x^2 + y^2}\) without intermediate overflow or underflow.

** `Math.sqrt(x)`**: Computes the positive square root of the value \(x\).

** `Math.exp(x)`**: Computes \(e^x\), where \(e\) is Euler's number.

** `Math.log(x, base)`**: Computes the logarithm of the value \(x\) to the given \(base\).

** `Math.log10(x)`**: Computes the common logarithm (base 10) of the value \(x\).

** `Math.log2(x)`**: Computes the binary logarithm (base 2) of the value \(x\).

** `Math.log1p(x)`**: Computes \(\ln(1 + x)\) for values of \(x\) near zero.
infinity.

* @return the smallest Float value that is greater than or equal to the given value \([x]\) and is a
mathematical integer.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun ceil(x: Float): Float =
nativeMath.ceil(x.toDouble()).toFloat()

/**
 * Rounds the given value \([x]\) to an integer towards negative
infinity.

* @return the largest Float value that is smaller than or equal to the given value \([x]\) and is an
integer.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun floor(x: Float): Float =
nativeMath.floor(x.toDouble()).toFloat()

/**
 * Rounds the given value \([x]\) to an integer towards zero.

* @return the value \([x]\) having its fractional part truncated.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun truncate(x: Float): Float = truncate(x.toDouble()).toFloat()

/**
 * Rounds the given value \([x]\) towards the closest integer with ties rounded towards even integer.

* @return the closest integer with ties rounded towards even integer.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun round(x: Float): Float =
round(x.toDouble()).toFloat()

/**
 * Returns the absolute value of the given value \([x]\).

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun abs(x: Float): Float =
nativeMath.abs(x.toDouble()).toFloat()

/**
 * Returns the sign of the given value \([x]\):
 *  - `-1.0` if the value is negative,
 *  - `0.0` if the value is zero,
 *  - `1.0` if the value is positive

* @return the sign of the value \([x]\).

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun sign(x: Float): Float =
nativeMath.sign(x.toDouble()).toFloat()

/**
 * Returns the smaller of two values.

* @return the smaller of two values.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun min(a: Float, b: Float): Float =
nativeMath.min(a, b)

/**
 * Returns the greater of two values.

* @return the greater of two values.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun max(a: Float, b: Float): Float =
nativeMath.max(a, b)

// extensions

/**
 * Raises this value to the power \([x]\).  

* Special cases:
*  - `b.pow(0.0)` is `1.0`
*  - `b.pow(1.0) == b`
*  - `b.pow(NaN)` is `NaN`
*  - `b.pow(x)` is `NaN` for `x` is `NaN`
*  - `NaN.pow(x)` is `NaN` for `x` is `NaN`
*  - `NaN.pow(x)` is `NaN` for `x` is finite and not an integer
*  - `NaN.pow(x)` is `NaN` for `x` is `NaN`
*  - `NaN.pow(x)` is `NaN` for `x` is finite and not an integer

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun Float.pow(x: Float): Float =
nativeMath.pow(this.toDouble(), x.toDouble()).toFloat()

/**
 * Raises this value to the integer power \([n]\).

* See the other overload of [pow] for details.

* @return the integer power \([n]\).

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun Float.pow(n: Int): Float =
nativeMath.pow(this.toDouble(), n.toDouble()).toFloat()

/**
 * Returns the absolute value of this value.

* Special cases:
*  - `-1.0` if the value is negative,
*  - `-0.0` if the value is zero,
*  - `1.0` if the value is positive

* @return the absolute value of this value.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun Float абсолютValue: Float get() =
nativeMath.abs(this.toDouble()).toFloat()

/**
 * Returns the sign bit same as of the [sign] value.

* If [sign] is `NaN` the sign of the result is undefined.

* @return the sign bit same as of the [sign] value.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun Float.withSign(sign: Float): Float =
this.toDouble().withSign(sign.toDouble()).toFloat()

/**
 * Returns the sign bit same as of the [sign] value.

* If [sign] is `NaN` the sign of the result is undefined.

* @return the sign bit same as of the [sign] value.

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun Float.withSign(sign: Int): Float =
this.toDouble().withSign(sign.toDouble()).toFloat()

/**
 * Rounds this [Float] value to the nearest integer and
converts the result to [Int].

* Ties are rounded towards positive infinity.

* @return the nearest integer and converts the result to [Int].

* @since Kotlin("1.2")
* @inlineOnly
public actual inline fun Float.roundToInt(): Int =
toDouble().roundToInt().toDouble().roundToLong() ==

```kotlin

```
Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n - `x.roundToLong()` \n
  @SinceKotlin("1.2")\n  @InlineOnly
  public actual inline fun Float.roundToLong(): Long = 

toDouble().roundToLong()\

@SinceKotlin("1.2")\n@InlineOnly
public actual inline fun Float.roundToLong(): Long = 

toDouble().roundToLong()\

/**
 * Returns the absolute value of the given value 

 * Special cases:\n * - `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow
 * - `abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an overflow
 * - `Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow
 * - `Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow
 */
@SinceKotlin("1.2")
public actual fun abs(n: Int): Int = if (n < 0) (-n or 0) else n

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.2")
@Suppress("NOTHING_TO_INLINE")
public actual inline fun min(a: Long, b: Long): Long = if (a <= b) a else b

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.2")
@Suppress("NOTHING_TO_INLINE")
public actual inline fun max(a: Long, b: Long): Long = if (a >= b) a else b

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of
 * this `Int` number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.countOneBits(): Int {
    // Hacker's Delight 5-1 algorithm
    var v = this
    v = (v and 0x55555555) + (v.ushr(1) and 0x55555555)
    v = (v and 0x33333333) + (v.ushr(2) and 0x33333333)
    v = (v and 0x0F0F0F0F) + (v.ushr(4) and 0x0F0F0F0F)
    v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)
    v = (v and 0x0000FFFF) + (v.ushr(16))
    return v
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.countOneBits(): Int {
    // Hacker's Delight 5-1 algorithm
    var v = this
    v = (v and 0x55555555) + (v.ushr(1) and 0x55555555)
    v = (v and 0x33333333) + (v.ushr(2) and 0x33333333)
    v = (v and 0x0F0F0F0F) + (v.ushr(4) and 0x0F0F0F0F)
    v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)
    v = (v and 0x0000FFFF) + (v.ushr(16))
    return v
}

/**
 * Counts the number of set bits in the binary representation of this `Int` number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.countOneBits(): Int {
    // Hacker's Delight 5-1 algorithm
    var v = this
    v = (v and 0x55555555) + (v.ushr(1) and 0x55555555)
    v = (v and 0x33333333) + (v.ushr(2) and 0x33333333)
    v = (v and 0x0F0F0F0F) + (v.ushr(4) and 0x0F0F0F0F)
    v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)
    v = (v and 0x0000FFFF) + (v.ushr(16))
    return v
}

*/
public actual inline fun Int.countLeadingZeroBits(): Int = kotlin.js.Math.clz32(this)

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Int] number.
 */
public actual fun Int.countTrailingZeroBits(): Int =
    // Hacker's Delight 5-4 algorithm for expressing countTrailingZeroBits with countLeadingZeroBits
    Int.SIZE_BITS - (this or -this).inv().countLeadingZeroBits()

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [Int] number.
 */
public actual fun Int.takeHighestOneBit(): Int =
    if (this == 0) 0 else 1 shl (Int.SIZE_BITS - 1 - countLeadingZeroBits())

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [Int] number.
 */
public actual fun Int.takeLowestOneBit(): Int =
    // Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit
    this and -this

/**
 * Rotates the binary representation of this [Int] number left by the specified [bitCount] number of bits.
 * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.
 * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count
 */
public actual fun Int.rotateLeft(bitCount: Int): Int =
    shl(bitCount) or ushr(Int.SIZE_BITS - bitCount)

/**
 * Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits.
 * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count
 */
public actual fun Int.rotateRight(bitCount: Int): Int =
    shl(Int.SIZE_BITS - bitCount) or ushr(bitCount)

/**
 * Counts the number of set bits in the binary representation of this [Long] number.
 */
public actual fun Long.countOneBits(): Int =
    high.countOneBits() + low.countOneBits()

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.
 */
public actual fun Long.countLeadingZeroBits(): Int =
    when (val high = this.high) {
        0 -> Int.SIZE_BITS + low.countLeadingZeroBits()
        else ->
            high.countLeadingZeroBits() + low.countLeadingZeroBits()
    }

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.
 */
public actual fun Long.countTrailingZeroBits(): Int =
    when (val low = this.low) {
        0 -> Int.SIZE_BITS + high.countTrailingZeroBits()
        else -> low.countTrailingZeroBits()
    }

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [Long] number.
 */
public actual fun Long.takeHighestOneBit(): Long =
    when (val high = this.high) {
        0 -> Long(0, high.takeHighestOneBit())
        else -> Long(high, Long.valueOf(0))
    }

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [Long] number.
 */
public actual fun Long.takeLowestOneBit(): Long =
    when (val low = this.low) {
        0 -> Long(0, high.takeLowestOneBit())
        else ->
            Long(low, Long.valueOf(0))
    }

/**
 * Rotates the binary representation of this [Long] number left by the specified [bitCount] number of bits.
 */
public actual fun Long.rotateLeft(bitCount: Int): Long =
    when (val high = this.high) {
        0 -> Long(0, high.rotateLeft(bitCount))
        else ->
            Long(high, Long.valueOf(0))
    }

/**
 * Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.
 */
public actual fun Long.rotateRight(bitCount: Int): Long =
    when (val low = this.low) {
        0 -> Long(0, high.rotateRight(bitCount))
        else ->
            Long(low, Long.valueOf(0))
    }

Authors: Antelope 4033
fun Long.rotateLeft(bitCount: Int): Long {
    if ((bitCount and 31) != 0) {
        // Rotate by multiple of 64
        val low = this.low
        val high = this.high
        val newLow = low.shl(bitCount) or high.ushr(-bitCount)
        val newHigh = high.shl(bitCount) or low.ushr(-bitCount)
        return if ((bitCount and 32) == 0) Long(newLow, newHigh) else Long(high, low)
    } else {
        return if ((bitCount and 32) == 0) this else Long(high, low)
    }
}

fun Long.rotateRight(bitCount: Int): Long = rotateLeft(-bitCount)

package kotlin.js

import kotlin.math.pow

internal actual fun defaultPlatformRandom(): Random =
    Random(js("(Math.random() * Math.pow(2, 32)) | 0").unsafeCast<Int>())

private val INV_2_26: Double = 2.0.pow(-26)
private val INV_2_53: Double = 2.0.pow(-53)
internal actual fun doubleFromParts(hi26: Int, low27: Int): Double =
    hi26 * INV_2_26 + low27 * INV_2_53

package kotlin

import getKClass
import kotlin.reflect.KClass
import kotlin.reflect.js.internal.KClassImpl

/**
 * Represents the constructor of a class. Instances of `JsClass` can be
 * passed to JavaScript APIs that expect a constructor reference.
 */
external interface JsClass<T : Any> {
    /**
     * Returns the unqualified name of the class represented by this instance.
     */
    val name: String
}

@Deprecated("Use class literal and extension property `js` instead.", replaceWith =
    ReplaceWith("T::class.js"), level = DeprecationLevel.ERROR)
external fun <T : Any> jsClass(): JsClass<T>

@Deprecated("Use class literal and extension property `js` instead.", replaceWith =
    ReplaceWith("this::class.js"), level = DeprecationLevel.ERROR)
val <T : Any> T.jsClass: JsClass<T>
    get() = when (jsTypeOf(this)) {
        "string" -> js("String")
        "number" -> js("Number")
        "boolean" -> js("Boolean")
        else -> js("Object").getPrototypeOf(this).constructor
    }

/**
 * Obtains a constructor reference for the given `KClass`
 */
val <T : Any> KClass<T>.js: JsClass<T>
    get() = (this as KClassImpl<T>).jsClass

val <T : Any> KClassImpl<T>.js: JsClass<T>
    get() = (this as KClassImpl<T>).jsClass
Any> JsClass<T>.kotlin: KClass<T>`

    get() = getKClass(this) /* Copyright 2010-2018 JetBrains s.r.o.
    and Kotlin Programming Language contributors. * Use of this source code is governed by the Apache 2.0 license
    that can be found in the license/LICENSE.txt file. */

import kotlin.reflect.*

package internal

import

internal abstract class KClassImpl<T : Any>(
    internal open val jClass: JsClass<T>
) : KClass<T> {
    override val annotations: List<Annotation>
        get() = TODO()
    override val constructors:
        Collection<KFunction<T>>
        get() = TODO()
    override val isAbstract: Boolean
        get() = TODO()
    override val isCompanion: Boolean
        get() = TODO()
    override val isData: Boolean
        get() = TODO()
    override val isFinal: Boolean
        get() = TODO()
    override val isInner: Boolean
        get() = TODO()
    override val isOpen: Boolean
        get() = TODO()
    override val isSealed: Boolean
        get() = TODO()
    override val members: Collection<KCallable<*>>
        get() = TODO()
    override val nestedClasses:
        Collection<KClass<*>>
        get() = TODO()
    override val objectInstance: T?
        get() = TODO()
    override val qualifiedName: String?
        get() = TODO()
    override val supertypes: List<KType>
        get() = TODO()
    override val typeParameters: List<KTypeParameter>
        get() = TODO()
    override val sealedSubclasses: List<KClass<out T>>
        get() = TODO()
    override val visibility: KVisibility?
        get() = TODO()

    override fun equals(other: Any?): Boolean {
        return other is KClassImpl<*> && jClass ==
    }

    // TODO: use FQN
    override fun hashCode(): Int = simpleName?.hashCode() ?: 0

    override fun toString(): String {
        // TODO: use FQN
        return "class $simpleName"
    }
}

internal class SimpleKClassImpl<T : Any>(jClass: JsClass<T>) : KClassImpl<T>(jClass) {
    override val simpleName: String? = jClass.asDynamic().`$metadata$`?.simpleName.unsafeCast<String?>()

    override fun
        isInstance(value: Any?): Boolean {
        return jsIsType(value, jClass)
    }
}

internal class

PrimitiveKClassImpl<T : Any>(
    jClass: JsClass<T>,
    private val givenSimpleName: String,
    private val
        isInstanceFunction: (Any?) -> Boolean
) : KClassImpl<T>(jClass) {
    override fun equals(other: Any?): Boolean
        {
            if (other !is PrimitiveKClassImpl<*>) return false
            return super.equals(other) && givenSimpleName
            == other.givenSimpleName
        }

    override val simpleName: String? get() = givenSimpleName

    override
        fun isInstance(value: Any?): Boolean {
        return isInstanceFunction(value)
    }
}

internal object

NothingKClassImpl : KClassImpl<Nothing>(js("Object")) {
    override val simpleName: String = "Nothing"

    override fun isInstance(value: Any?): Boolean = false

    override val jClass:
        JsClass<Nothing>
        get() = throw UnsupportedOperationException("There's no native JS class for Nothing type")

    override fun equals(other: Any?): Boolean = other === this

    override fun hashCode(): Int = 0
}

@file:Suppress("Unsupported")

import

@JsName("createKType")
internal fun createKType(
    classifier: KClassifier,
    arguments: Array<KTypeProjection>,
    isMarkedNullable: Boolean
) =

    KTypeImpl(classifier, arguments.asList(), isMarkedNullable)

@JsName("createDynamicKType")
internal fun

    createDynamicKType(): KType = DynamicKType

@JsName("markKTypeNullable")
internal fun

    markKTypeNullable(kType: KType) = KTypeImpl(kType.classifier!!, kType.arguments,

    true)

@JsName("createKTypeParameter")
internal fun createKTypeParameter(
    name: String,

    upperBounds: Array<KType>,

    variance: String
): KTypeParameter {
    val kVariance = when (variance) {
        "in" -> KVariance.IN
        "out" -> KVariance.OUT
        else -> KVariance.INVARIANT
    }

    return

    KTypeParameterImpl(name, upperBounds.asList(), kVariance,

    false)
}

@file:Suppress("UnstableApiUsage")

public

Kotlin

createContravariantKTypeProjection(type: KType) =
KTypeProjection.\\n
KTypeProjection.\\ns\\n
override val classifier: KClassifier,\\n
override val arguments: List<KTypeProjection>,\\n
override fun hashCode(): Int =

(\\n classifier.hashCode() * 31 + arguments.hashCode() * 31 +

isMarkedNullable.hashCode().\\n
override fun toString(): String =
\\n val classifierName = when (\\n kClass && kClass.simpleName != null -> kClass.simpleName\\n else -> "(non-denotable type)"

val arguments = when (arguments.isEmpty() -> ""
else arguments.joinToString("", ",", ",">") { it.asString() }

val nullable = if (isMarkedNullable) "?" else ""

// TODO: this should be the implementation of KTypeProjection.toString, see KT-30071
private fun KTypeProjection.asString(): String {
    if (variance == null) return "*"
    return variance.prefixString() +
    type.toString()
}

internal class KTypeImpl(
    override val classifier: KClassifier,
    override val arguments: List<KTypeProjection>,
    override val isMarkedNullable: Boolean)

    override fun get() =
        emptyList() // KClass && other is KTypeImpl &&
        val classifier = other.classifier &&
        arguments = other.arguments &&
        isMarkedNullable = other.isMarkedNullable

override fun hashCode(): Int =
    (classifier.hashCode() * 31 + arguments.hashCode() * 31 +

isMarkedNullable.hashCode().\\n
override fun toString(): String =
\\n val classifierName = when (\\n kClass && kClass.simpleName != null -> kClass.simpleName\\n else -> "(non-denotable type)"

val arguments = when (arguments.isEmpty() -> ""
else arguments.joinToString("", ",", ",">") { it.asString() }

val nullable = if (isMarkedNullable) "?" else ""

// TODO: this should be the implementation of KTypeProjection.toString, see KT-30071
private fun KTypeProjection.asString(): String {
    if (variance == null) return "*"
    return variance.prefixString() +
    type.toString()
}

internal class KTypeImpl(
    override val classifier: KClassifier,
    override val arguments: List<KTypeProjection>,
    override val isMarkedNullable: Boolean)

    override fun get() =
        emptyList() // KClass && other is KTypeImpl &&
        val classifier = other.classifier &&
        arguments = other.arguments &&
        isMarkedNullable = other.isMarkedNullable

override fun hashCode(): Int =
    (classifier.hashCode() * 31 + arguments.hashCode() * 31 +

isMarkedNullable.hashCode().\\n
override fun toString(): String =
\\n val classifierName = when (\\n kClass && kClass.simpleName != null -> kClass.simpleName\\n else -> "(non-denotable type)"

val arguments = when (arguments.isEmpty() -> ""
else arguments.joinToString("", ",", ",">") { it.asString() }

val nullable = if (isMarkedNullable) "?" else ""

// TODO: this should be the implementation of KTypeProjection.toString, see KT-30071
private fun KTypeProjection.asString(): String {
    if (variance == null) return "*"
    return variance.prefixString() +
    type.toString()
// a package is omitted to get declarations directly under the module

import kotlin.reflect.*
import kotlin.reflect.js.internal.*

@JsName("getKClass")
internal fun <T : Any> getKClass(jClass: JsClass<T>): KClass<T> =
    when (jsTypeOf(jClass)) {
        "string" -> PrimitiveClasses.stringClass
        "number" -> if (jsBitwiseOr(jClass, 0).asDynamic() === jClass)
            PrimitiveClasses.intClass
        else
            PrimitiveClasses.doubleClass
        "boolean" -> PrimitiveClasses.booleanClass
        "function" -> PrimitiveClasses.functionClass(jClass.asDynamic().length)
        else -> {
            when {
                jClass is BooleanArray -> PrimitiveClasses.booleanArrayClass
                jClass is CharArray -> PrimitiveClasses.charArrayClass
                jClass is ByteArray -> PrimitiveClasses.byteArrayClass
                jClass is ShortArray -> PrimitiveClasses.shortArrayClass
                jClass is IntArray -> PrimitiveClasses.intArrayClass
                jClass is LongArray -> PrimitiveClasses.longArrayClass
                jClass is FloatArray -> PrimitiveClasses.floatArrayClass
                jClass is DoubleArray -> PrimitiveClasses.doubleArrayClass
                jClass is KClass<*> -> jClass
                jClass is Array<*> -> PrimitiveClasses.arrayClass
                else -> {
                    val constructor = js("Object").getPrototypeOf(jClass).constructor
                    when {
                        constructor === js("Object") -> PrimitiveClasses.anyClass
                        constructor === js("Error") -> PrimitiveClasses.throwableClass
                        else -> {
                            val jsClass: JsClass<T> = constructor
                            getOrCreateKClass(jsClass)
                        }
                    }
                }
            }
        }
    }.unsafeCast<KClass<T>>()

private fun <T : Any> getOrCreateKClass(jClass: JsClass<T>): KClass<T> {
    if (jClass === js("String")) return PrimitiveClasses.stringClass.unsafeCast<KClass<T>>()

    val metadata = jClass.asDynamic().`$metadata$`
    return if (metadata != null) {
        if (metadata.`$kClass$` == null) {
            val kClass = SimpleKClassImpl(jClass)
            metadata.`$kClass$` = kClass
            kClass
        } else {
            metadata.`$kClass$`
        }
    } else {
        SimpleKClassImpl(jClass)
    }
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

/**
 * Exposes the JavaScript RegExp
 */

```kotlin
@Suppress("NOT_DOCUMENTED")
public external class RegExp(pattern: String, flags: String? =
    definedExternally) {

    public fun test(str: String): Boolean

    public fun exec(str: String): RegExpMatch?

    public override fun toString(): String

    /**
     * The lastIndex is a read/write integer property of regular
     * expressions that specifies the index at which to start the next match.
     * @public var lastIndex: Int
     * @public val global: Boolean
     * @public val ignoreCase: Boolean
     * @public val multiline: Boolean
     */
    public var lastIndex: Int

    public val global: Boolean
    public val ignoreCase: Boolean
    public val multiline: Boolean
}

/**
 * Resets the regular expression so that subsequent [RegExp.test] and [RegExp.exec] calls will match starting with the beginning
 * of the input string.
 * @public fun RegExp.reset() {
 *    lastIndex = 0
 *}
 */

// TODO: Inherit from array or introduce asArray() extension
/**
 * Represents the return value of [RegExp.exec].
 */
@Suppress("NOT_DOCUMENTED")
public external interface RegExpMatch {
    public val index: Int
    public val input: String
    public val length: Int
}

/**
 * Returns the entire text matched by [RegExp.exec] if
 * the [index] parameter is 0, or the text matched by the capturing parenthesis
 * at the given index.
 */
public inline

/**
 * Converts the result of
 * [RegExp.exec] to an array where the first element contains the entire matched text and each subsequent
 * element is the text matched by each capturing parenthesis.
 */
public inline fun RegExpMatch.asArray(): Array<out String?> = unsafeCast<Array<out String?>>()

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
 * found in the license/LICENSE.txt file.
 */

package kotlin.sequences

internal actual class ConstrainedOnceSequence<T> actual constructor(sequence: Sequence<T>) : Sequence<T> {
    private var sequenceRef: Sequence<T>? = sequence

    actual override fun iterator(): Iterator<T> {
        val sequence = sequenceRef ?: throw IllegalStateException("This sequence can be consumed only once.")
        sequenceRef = null
        return sequence.iterator()
    }
}

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
 * found in the license/LICENSE.txt file.
 */

package kotlin.text

// actually \s is enough to match all whitespace, but \xA0 added because of different regexp behavior of Rhino used in Selenium tests
public actual fun Char.isWhitespace(): Boolean = toString().matches("[\s\xA0]"")

@kotlin.internal.InlineOnly
public actual inline fun Char.toLowerCase(): Char = js("String.fromCharCode")(toInt()).toLowerCase().charCodeAt(0).unsafeCast<Int>().toChar()

@kotlin.internal.InlineOnly
public actual inline fun Char.toUpperCase(): Char = js("String.fromCharCode")(toInt()).toUpperCase().charCodeAt(0).unsafeCast<Int>().toChar()

/**
 * Returns `true` if this character is a Unicode high-surrogate code unit (also known as leading-surrogate code unit).
 */
public actual fun Char.isHighSurrogate(): Boolean = this in Char.MIN_HIGH_SURROGATE..Char.MAX_HIGH_SURROGATE

/**
 * Returns `true` if this character is a Unicode low-surrogate code unit (also known as trailing-surrogate code unit).
 */
public actual fun Char.isLowSurrogate(): Boolean = this in Char.MIN_LOW_SURROGATE..Char.MAX_LOW_SURROGATE

/**
 * Returns `true` if the contents of this string is equal to the word "true", ignoring case, and `false` otherwise.
 */
public actual fun String.toBoolean(): Boolean = toLowerCase() == "true"

/**
 * Parses the string as a signed [Byte] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toByte(): Byte = toByteOrNull() ?: numberFormatException(this)

/**
 * Parses the string as a signed [Byte] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toByte(radix: Int): Byte = toByteOrNull(radix) ?: numberFormatException(this)

/**
 * Parses the string as a [Short] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toShort(): Short = toShortOrNull() ?: numberFormatException(this)
```
number and returns the result.
@throws NumberFormatException if the string is not a valid representation of a number.
@throws IllegalArgumentException when \[radix\] is not a valid radix for string to number conversion.

Parses the string as an [Int] number and returns the result.
@throws NumberFormatException if the string is not a valid representation of a number.

Parses the string as an [Int] number and returns the result.
@throws NumberFormatException if the string is not a valid representation of a number.

Parses the string as a [Double] number and returns the result.
@throws NumberFormatException if the string is not a valid representation of a number.

Parses the string as a [Double] number and returns the result.
@throws NumberFormatException if the string is not a valid representation of a number.

Parses the string as a [Byte] value in the specified [radix].
@throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.

@SinceKotlin("1.2")
public actual fun Byte.toString(radix: Int): String = this.toInt().toString(radix)

@SinceKotlin("1.2")
public actual fun Short.toString(radix: Int): String = this.toInt().toString(radix)

public actual fun Int.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))

private fun String.isNaN(): Boolean = when (this.toLowerCase()) {
    "nan", "+nan", "-nan" -> true
    else -> false
}

/**
 * Checks whether the given [radix] is valid radix for string to number and number to string conversion.
 */
@PublishedApi
internal actual fun checkRadix(radix: Int): Int {
    if (radix !in 2..36) {
        throw IllegalArgumentException("radix $radix was not in valid range 2..36")
    }
    return radix
}

digitOf(char: Char, radix: Int): Int {
    when {
        char >= '0' && char <= '9' -> char - '0'
        char >= 'A' && char <= 'Z' -> char - 'A' + 10
        char >= 'a' && char <= 'z' -> char - 'a' + 10
        else -> -1
    }.let { if (it >= radix) -1 else it }
}

internal actual fun String.toDoubleOrNull(): Double? = (+(this.asDynamic())).unsafeCast<Double>().takeIf {
    !(it.isNaN() && !this.isNaN() || it == 0.0 && this.isBlank())
}

@kotlin.internal.InlineOnly
public actual inline fun String.toFloatOrNull(): Float? = toDoubleOrNull().unsafeCast<Float?>()

/**
 * Returns a string representation of this [Byte] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.2")
public actual fun Byte.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Short] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.2")
public actual fun Short.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Int] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.2")
public actual fun Int.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))

internal actual fun String.toDouble(): Double = (+(this.asDynamic())).unsafeCast<Double>().also {
    if (it.isNaN() && !this.isNaN() || it == 0.0 && this.isBlank())
        numberFormatError(this)
}

/**
 * Parses the string as a [Double] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toDouble(): Double = (+(this.asDynamic())).unsafeCast<Double>().also {
    if (it.isNaN() && !this.isNaN() || it == 0.0 && this.isBlank())
        numberFormatError(this)
}

/**
 * Parses the string as a [Float] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
@kotlin.internal.InlineOnly
public actual inline fun String.toFloat(): Float = toDouble().unsafeCast<Float>()

/**
 * Parses the string as a [Double] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toDouble(): Double = toDoubleOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a [Float] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toFloat(): Float = toFloatOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a [Double] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toDouble(): Double = toDoubleOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a [Float] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toFloat(): Float = toFloatOrNull() ?: numberFormatError(this)

/**
 * Returns a string representation of this [Byte] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual inline fun Byte.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Short] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual inline fun Short.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Int] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
public actual fun Int.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))

private fun String.isNaN(): Boolean = when (this.toLowerCase()) {
    "nan", "+nan", "-nan" -> true
    else -> false
}

/**
 * Checks whether the given [radix] is valid radix for string to number and number to string conversion.
 */
@PublishedApi
internal actual fun checkRadix(radix: Int): Int {
    if (radix !in 2..36) {
        throw IllegalArgumentException("radix $radix was not in valid range 2..36")
    }
    return radix
}

digitOf(char: Char, radix: Int): Int {
    when {
        char >= '0' && char <= '9' -> char - '0'
        char >= 'A' && char <= 'Z' -> char - 'A' + 10
        char >= 'a' && char <= 'z' -> char - 'a' + 10
        else -> -1
    }.let { if (it >= radix) -1 else it }
}

internal actual fun String.toDoubleOrNull(): Double? = (+(this.asDynamic())).unsafeCast<Double>().takeIf {
    !(it.isNaN() && !this.isNaN() || it == 0.0 && this.isBlank())
}

@kotlin.internal.InlineOnly
public actual inline fun String.toFloatOrNull(): Float? = toDoubleOrNull().unsafeCast<Float?>()
public actual data class MatchGroup(actual val value: String)

/**
 * Represents a compiled regular expression. 
 * Provides functions to match strings in text with a pattern, replace the found occurrences and split text around matches. 
 * For pattern syntax reference see [MDN RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp#Special_characters_meaning_in_regular_expressions) and
[http://www.w3schools.com/jsref/jsref_obj_regexp.asp](https://www.w3schools.com/jsref/jsref_obj_regexp.asp).
 */
public actual class Regex constructor(pattern: String, option: RegexOption) {

    constructor(pattern: String, options: Set<RegexOption>) : this(pattern, setOf(option))

    constructor(pattern: String) : this(pattern, emptySet())

    /** The pattern string of this regular expression. */
    public actual val pattern: String = pattern
    /** The set of options that were used to create this regular expression. */
    public actual val options: Set<RegexOption> = options.toSet()
    private val nativePattern: RegExp = RegExp(pattern, options.map { it.value }.joinToString(separator = "") + "g")

    /** Indicates whether the regular expression matches the entire [input]. 
     * @param input A CharSequence that is matched. 
     * @return Boolean true if the regular expression matches the entire [input]. false otherwise.
     */
    public actual fun containsMatchIn(input: CharSequence): Boolean {
        nativePattern.reset()
        return nativePattern.test(input.toString())
    }

    /** Returns a sequence of all occurrences of a regular expression within the [input] string, beginning at the specified [startIndex]. 
     * @param input A CharSequence that is matched. 
     * @param startIndex An index to start search with, by default 0. Must be not less than zero and not greater than \( \text{input.length} \). 
     * @return A Sequence of MatchResult if found or an empty sequence otherwise.
     */
    public actual fun findAll(input: CharSequence, startIndex: Int = 0): Sequence<MatchResult> =
        generateSequence({ find(input, startIndex) }, { match -> match.next() })

    /** Attempts to match the entire [input] CharSequence against the pattern. 
     * @return An instance of MatchResult if the entire input matches or null otherwise.
     */
    public actual fun matchEntire(input: CharSequence): MatchResult? {
        if (pattern.startsWith('^') && pattern.endsWith('$'))
            return find(input)
        else
            return Regex("^${pattern.trimStart('^').trimEnd('$')}$", options).find(input)
    }

    /** Replaces all occurrences of this regular expression in the specified [input] string with specified [replacement] expression. 
     * @param input A CharSequence that is matched. 
     * @param replacement A replacement expression that can include substitutions. See [String.prototype.replace](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/replace) function docs for details.
     * @return A new string with all occurrences of the regular expression replaced.
     */
    public actual fun replace(input: CharSequence, replacement: String): String =
        input.toString().nativeReplace(nativePattern, replacement)

    /** Replaces the first occurrence of this regular expression in the specified [input] string with specified [replacement] expression, 
     * beginning at the specified [startIndex]. 
     * @param input A CharSequence that is matched. 
     * @param replacement A replacement expression that can include substitutions. See [String.prototype.replace](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/replace) function docs for details.
     * @return A new string with the first occurrence of the regular expression replaced.
     */
    public actual fun replaceFirst(input: CharSequence, replacement: String): String =
        input.toString().nativeReplace(nativePattern, replacement)[startIndex]
occurrence of this regular expression in the specified [input] string with specified [replacement] expression.

* @param replacement A replacement expression that can include substitutions. See


* @param limit Non-negative value specifying the maximum number of substrings the string can be split to.

* @suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")

* Returns the string representation of this regular expression, namely the [pattern] of this regular expression.

* Note that another regular expression constructed from the same pattern string may have different [options] and may match strings differently.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* @actual fun replaceFirst(input: CharSequence, replacement: String): String {

* @param replacement A replacement expression that can include substitutions. See


*/

* @suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")

* Returns a regular expression that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when it is used as a replacement string in [RegExp.replace] function.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.

* Returns a regular expression pattern string that matches the specified [literal] string literally.

* No characters of that string will have special meaning when searching for an occurrence of the regular expression.
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

package kotlin.text

import kotlin.js.RegExp

@kotlin.internal.InlineOnly
internal actual inline fun String.nativeIndexOf(ch: Char, fromIndex: Int): Int = nativeIndexOf(ch.toString(), fromIndex)

@kotlin.internal.InlineOnly
internal actual inline fun String.nativeLastIndexOf(ch: Char, fromIndex: Int): Int = nativeLastIndexOf(ch.toString(), fromIndex)

/**
 * Returns `true` if this string starts with the specified prefix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.startsWith(prefix: String, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeStartsWith(prefix, 0)
    else
        return regionMatches(0, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if a substring of this string starting at the specified offset [startIndex] starts with the specified prefix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.startsWith(prefix: String, startIndex: Int, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeStartsWith(prefix, startIndex)
    else
        return regionMatches(startIndex, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if this string ends with the specified suffix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.endsWith(suffix: String, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeEndsWith(suffix)
    else
        return regionMatches(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
}

public fun String.matches(regex: String): Boolean {
    val result = this.match(regex)
    return result != null && result.size != 0
}

public actual fun CharSequence.isBlank(): Boolean = length == 0 || (if (this is String) this else toString()).matches("^\[\\s\\xA0$\]$")

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.equals(other: String?, ignoreCase: Boolean = false): Boolean =
    if (this == null)
        other == null
    else if (!ignoreCase)
        this == other
    else
        other != null && this.toLowerCase() == other.toLowerCase()

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun CharSequence.regionMatches(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean = false): Boolean =
    regionMatchesImpl(thisOffset, other, otherOffset, length, ignoreCase)

/**
 * Returns a copy of this string having its first letter uppercased, or the original string, if it's empty or already starts with an upper case letter.
 */
public actual fun String.capitalize(): String {
    return if (isNotEmpty()) substring(0, 1).toUpperCase() + substring(1) else this
}

/**
 * Returns a copy of this string having its first letter lowercased, or the original string, if it's empty or already starts with a lower case letter.
 */
public actual fun String.decapitalize(): String {
    return if (isNotEmpty()) substring(0, 1).toLowerCase() + substring(1)
}

/**
 * Returns a string containing this char sequence repeated \[n\] times.
 * @throws [IllegalArgumentException] when \[n\] < 0.
 */
public actual fun CharSequence.repeat(n: Int): String {
    require(n >= 0) { "Count 'n' must be non-negative, but was $n." }
    return when (n) {
        0 -> ""
        1 -> this.toString()
        else -> {
            var result = ""
            if (!isEmpty()) {
                var s = this.toString()
                var count = n
                while (true) {
                    if ((count and 1) == 1) {
                        result += s
                    }
                    count = count ushr 1
                    if (count == 0) {
                        break
                    }
                }
            }
            return result
        }
    }
}

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =
    nativeReplace(RegExp(RegExp.escape(oldValue), if (ignoreCase) "gi" else "g"), RegExp.escapeReplacement(newValue))

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =
    nativeReplace(RegExp(RegExp.escape(oldChar.toString()), if (ignoreCase) "g" else "g"), newChar.toString())

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =
    nativeReplace(RegExp(RegExp.escape(oldValue), if (ignoreCase) "g" else "g"), first())
null @Support("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENT_SI")

public actual fun String.replaceFirst(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(Regex.escape(oldChar.toString()), if (ignoreCase) "i" else ""),
newChar.toString())

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENT_SI")
public actual fun String.replaceFirst(c: Char): Appendable
private var string = content
actual override fun get(index: Int): Char =
string.getOrElse(index) { throw IndexOutOfBoundsException("index: $index, length: $length") }

actual override fun subSequence(startIndex: Int, endIndex: Int): CharSequence =
string.substring(startIndex, endIndex)

actual fun reverse(): StringBuilder {
string = string.asDynamic().split("\"\"`).reverse().join("\"\"`) return this
}

/**
 * Clears the content of this string builder making it empty.
 *
 * @sample samples.text.Strings.clearStringBuilder
 */
@SinceKotlin("1.3")
@Suppress("EXTENSION_SHADOWED_BY_MEMBER",
"NOTHING_TO_INLINE")
public actual inline fun StringBuilder.clear(): StringBuilder = this.clear()

/**
 * Returns the negative [size] if [throwOnMalformed] is false, throws [CharacterCodingException]
other
wise.
 */
@UseExperimental(ExperimentalStdlibApi::class)
private fun malformed(size: Int, index: Int,
throwOnMalformed: Boolean): Int {
if (throwOnMalformed) throw CharacterCodingException("Malformed
sequence starting at ${index - 1}")
return -size
}

/**
 * Returns code point corresponding to UTF-16 surrogate pair,
 * where the first of the pair is the [high] and the second is in the [string] at the [index].
 * Returns zero if the pair is malformed and [throwOnMalformed] is false.
 *
 * @throws CharacterCodingException if the pair is malformed and [throwOnMalformed] is true.
 */
private fun codePointFromSurrogate(string: String, high: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
if (high !in 0xD800..0xDBFF || index >= endIndex) {
return malformed(0, index, throwOnMalformed)
}
val low = string[index].toInt()
if (low !in 0xDC00..0xDFFF) {
return malformed(0, index, throwOnMalformed)
}
return 0x10000 + ((high and 0x3FF) shl 10) or (low and 0x3FF)

/**
 * Returns code point corresponding to UTF-8 sequence of
 * two bytes,
 * where the first byte of the sequence is the [byte1] and the second byte is in the [bytes] array at the
 * [index].
 * Returns zero if the sequence is malformed and [throwOnMalformed] is false.
 *
 * @throws CharacterCodingException if the sequence of two bytes is malformed and [throwOnMalformed] is true.
 */
private fun codePointFrom2(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
if (byte1 and 0x1E == 0 || index >= endIndex) {
return malformed(0, index, throwOnMalformed)
}
val byte2 = bytes[index].toInt()
if (lowIn 0xDC00..0xDFFF) {
return malformed(0, index, throwOnMalformed)
}
code point corresponding to UTF-8 sequence of three bytes.

* where the first byte of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].

* Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence

* if the sequence is malformed and [throwOnMalformed] is false.

* @throws CharacterCodingException if the sequence of three bytes is malformed and [throwOnMalformed] is true.

*/
private fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }

    val byte2 = bytes[index].toInt()
    if (byte1 and 0xF == 0) {
        if (byte2 and 0xE0 != 0xA0) {
            // Non-shortest form
            return malformed(0, index, throwOnMalformed)
        }
    } else if (byte1 and 0xF == 0xD) {
        if (byte2 and 0xE0 != 0x80) {
            // Surrogate code point
            return malformed(0, index, throwOnMalformed)
        }
    } else if (byte2 and 0xC0 != 0x80) {
        return malformed(0, index, throwOnMalformed)
    }

    if (index + 1 == endIndex) {
        return malformed(1, index, throwOnMalformed)
    }
    val byte3 = bytes[index + 1].toInt()
    if (byte3 and 0xC0 != 0x80) {
        return malformed(1, index, throwOnMalformed)
    }

    if (index + 2 == endIndex) {
        return malformed(2, index, throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte4 and 0xC0 != 0x80) {
        return malformed(2, index, throwOnMalformed)
    }
    return (byte1 shl 12) xor (byte2 shl 6) xor byte3 xor -0x1E080
}

/**
 * Returns code point corresponding to UTF-8 sequence of four bytes,

* where the first byte of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].

* Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence

* if the sequence is malformed and [throwOnMalformed] is false.

* @throws CharacterCodingException if the sequence of four bytes is malformed and [throwOnMalformed] is true.

*/
    if (index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }

    val byte2 = bytes[index].toInt()
    if (byte1 and 0xF == 0x0) {
        if (byte2 and 0xF0 <= 0x80) {
            // Non-shortest form
            return malformed(0, index, throwOnMalformed)
        }
    } else if (byte1 and 0xF == 0x4) {
        if (byte2 and 0xF0 != 0x80) {
            // Out of Unicode code points domain (larger than U+10FFFF)
            return malformed(0, index, throwOnMalformed)
        }
    } else if (byte1 and 0xF > 0x4) {
        return malformed(0, index, throwOnMalformed)
    } else if (byte2 and 0xC0 != 0x80) {
        return malformed(0, index, throwOnMalformed)
    }

    if (index + 1 == endIndex) {
        return malformed(1, index, throwOnMalformed)
    }
    val byte3 = bytes[index + 1].toInt()
    if (byte3 and 0xC0 != 0x80) {
        return malformed(1, index, throwOnMalformed)
    }

    if (index + 2 == endIndex) {
        return malformed(2, index, throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte4 and 0xC0 != 0x80) {
        return malformed(2, index, throwOnMalformed)
    }
    return (byte1 shl 18) xor (byte2 shl 12) xor (byte3 shl 6) xor byte4 xor 0x381F80
}

/**
 * Maximum number of bytes needed to encode a single char.

* Code points in `0..0x7F` are encoded in a single byte.
* Code points in `0x80..0x7FF` are encoded in two bytes.
* Code points in `0x800..0xD7FF` or in `0xE000..0xFFFF` are encoded in three bytes.
* Surrogate code points in `0xD800..0xDFFF` are not Unicode scalar values, therefore aren't encoded.

* Code points in `0x10000..0x10FFFF` are represented by a pair of surrogate 'Char's and are encoded in four bytes.

* Anprivate const val MAX_BYTES_PER_CHAR = 3

* @param string the string to encode.
* @param startIndext the start offset (inclusive) of the substring to code.
* @param endIndext the end offset (exclusive) of the substring to encode.

* @param throwOnMalformed whether to throw on malformed char sequence or replace by the

* [REPLACEMENT_BYTE_SEQUENCE].

* @throws CharacterCodingException if the char sequence is malformed and [throwOnMalformed] is true.

*/
internal fun encodeUtf8(string: String, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): ByteArray {
    require(startIndex >= 0 && endIndex <= string.length && startIndex <= endIndex)
    val bytes = ByteArray((endIndex - startIndex) * MAX_BYTES_PER_CHAR)
    var byteIndex = 0
    var charIndex = startIndex
    while (charIndex < endIndex) {
        val code = string[charIndex++].toInt()
        when {
            code < 0x80 -> {
                if (throwOnMalformed) {
                    throw CharacterCodingException()
                } else {
                    bytes[byteIndex++] = code.toByte()
                }
                byteIndex += 1
            }
            code < 0x800 -> {
                if (throwOnMalformed) {
                    throw CharacterCodingException()
                } else {
                    bytes[byteIndex++] = (code shr 6) or 0x80
                    bytes[byteIndex++] = code and 0x3F
                }
                byteIndex += 2
            }
            code < 0x10000 -> {
                if (throwOnMalformed) {
                    throw CharacterCodingException()
                } else {
                    bytes[byteIndex++] = (code shr 12) or 0x80
                    bytes[byteIndex++] = (code shr 6) or 0x80
                    bytes[byteIndex++] = code and 0x3F
                }
                byteIndex += 3
            }
            code <= 0x11FFFF -> {
                if (throwOnMalformed) {
                    throw CharacterCodingException()
                } else {
                    bytes[byteIndex++] = (code shr 18) or 0x80
                    bytes[byteIndex++] = (code shr 12) or 0x80
                    bytes[byteIndex++] = (code shr 6) or 0x80
                    bytes[byteIndex++] = code and 0x3F
                }
                byteIndex += 4
            } else {
                throw CharacterCodingException()
            }
        }
        charIndex++
    }
    return bytes
}
definedExternally
var stencil: Boolean? /* = false */
set(value) = definedExternally
get() = definedExternally
set(value) = definedExternally
var antialias: Boolean? /* = true */
set(value) = definedExternally
get() = definedExternally
set(value) = definedExternally
var premultipliedAlpha: Boolean? /* = true */
set(value) = definedExternally
get() = definedExternally
set(value) = definedExternally
var preserveDrawingBuffer: Boolean? /* = false */
set(value) = definedExternally
get() = definedExternally
set(value) = definedExternally
var preferLowPowerToHighPerformance: Boolean? /* = false */
set(value) = definedExternally
get() = definedExternally
set(value) = definedExternally
var failIfMajorPerformanceCaveat: Boolean? /* = false */
set(value) = definedExternally
get() = definedExternally
set(value) = definedExternally

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["alpha"] = alpha
    o["depth"] = depth
    o["stencil"] = stencil
    o["antialias"] = antialias
    o["premultipliedAlpha"] = premultipliedAlpha
    o["preserveDrawingBuffer"] = preserveDrawingBuffer
    o["preferLowPowerToHighPerformance"] = preferLowPowerToHighPerformance
    o["failIfMajorPerformanceCaveat"] = failIfMajorPerformanceCaveat
    return o
}

public external abstract
class WebGLObject

/**
 * Exposes the JavaScript
[WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin
 */
public external abstract class WebGLBuffer : WebGLObject

/**
 * Exposes the JavaScript
[WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin
 */
public external abstract class WebGLFramebuffer : WebGLObject

/**
 * Exposes the JavaScript
[WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin
 */
public external abstract class WebGLProgram : WebGLObject

/**
 * Exposes the JavaScript
[WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin
 */
public external abstract class WebGLRenderbuffer : WebGLObject

/**
 * Exposes the JavaScript
[WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin
 */
public external abstract class WebGLShader : WebGLObject

/**
 * Exposes the JavaScript
[WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin
 */
public external abstract class WebGLTexture : WebGLObject

/**
 * Exposes the JavaScript
[WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin
 */
public external abstract class WebGLUniformLocation

/**
 * Exposes the JavaScript
[WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin
 */
public external abstract class WebGLActiveInfo {
    open val size: Int
    open val type: Int
    open val name: String
}

/**
 * Exposes the JavaScript
[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to Kotlin
 */
public external interface WebGLRenderingContextBase {
    val canvas: HTMLCanvasElement
    val drawingBufferWidth: Int
    val drawingBufferHeight: Int
    fun getContextAttributes(): WebGLContextAttributes?
    fun isContextLost(): Boolean
    fun getSupportedExtensions(): Array<String>
    fun attachShader(program: WebGLProgram?, shader: WebGLShader?)
    fun bindAttribLocation(program: WebGLProgram?, index: Int, name: String)
    fun bindBuffer(target: Int, buffer: WebGLBuffer?)
    fun bindFramebuffer(target: Int, framebuffer: WebGLFramebuffer?)
    fun bindRenderbuffer(target: Int, renderbuffer: WebGLRenderbuffer?)
    fun bindTexture(target: Int, texture: WebGLTexture?)
    fun blendColor(red: Float, green: Float, blue: Float, alpha: Float?)
    fun blendEquation(mode: Int)
    fun blendEquationSeparate(modeRGB: Int, modeAlpha: Int)
    fun blendFunc(sfactor: Int, dfactor: Int)
    fun blendFuncSeparate(srcRGB: Int, dstRGB: Int)
    fun bindBufferData(target: Int, usage: Int)
    fun bufferData(target: Int, data: BufferDataSource?,
fun texParameterf(target: Int, pname: Int, param: Float)
fun texParameteri(target: Int, pname: Int, param: Int)
fun texParameterf(target: Int, pname: Int, param: Float)
fun texParameteri(target: Int, pname: Int, param: Int)
fun uniform1f(location: WebGLUniformLocation?, x: Float)
fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform1f(location: WebGLUniformLocation?, x: Float)
fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform1i(location: WebGLUniformLocation?, x: Int)
fun uniform1iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform1i(location: WebGLUniformLocation?, x: Int)
fun uniform1iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform2f(location: WebGLUniformLocation?, x: Float, y: Float)
fun uniform2fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform2f(location: WebGLUniformLocation?, x: Float, y: Float)
fun uniform2fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform2i(location: WebGLUniformLocation?, x: Int, y: Int)
fun uniform2iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform2i(location: WebGLUniformLocation?, x: Int, y: Int)
fun uniform2iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float)
fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float)
fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int)
fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int)
fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float, w: Float)
fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float, w: Float)
fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform4i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int)
fun uniform4iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform4i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int)
fun uniform4iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun useProgram(program: WebGLProgram?)
fun validateProgram(program: WebGLProgram?)
fun vertexAttrib1f(index: Int, x: Float)
fun vertexAttrib1fv(index: Int, values: dynamic)
fun vertexAttrib2f(index: Int, x: Float, y: Float)
fun vertexAttrib2fv(index: Int, values: dynamic)
fun vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)
fun vertexAttrib3fv(index: Int, values: dynamic)
fun vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)
fun vertexAttrib4fv(index: Int, values: dynamic)
fun viewport(x: Int, y: Int, width: Int, height: Int)
fun STENCIL_BUFFER_BIT
fun COLOR_BUFFER_BIT
fun POINTS
fun LINES
fun LINE_LOOP
fun LINE_STRIP
fun TRIANGLES
fun TRIANGLE_STRIP
fun TRIANGLE_FAN
fun ZERO
fun ONE
fun SRC_COLOR
fun ONE_MINUS_SRC_COLOR
fun SRC_ALPHA
fun ONE_MINUS_SRC_ALPHA
fun DST_ALPHA
fun ONE_MINUS_DST_ALPHA
fun DST_COLOR
fun ONE_MINUS_DST_COLOR
fun SRC_ALPHA_SATURATE
fun FUNC_ADD
fun BLEND_EQUATION
fun BLEND_EQUATION_RGB
fun BLEND_EQUATION_ALPHA
fun BLEND_SUBTRACT
fun FUNC_SUBTRACT
fun BLEND_RGB
fun BLEND_RGBA
fun BLEND_DST_RGB
fun BLEND_SRC_RGB
fun BLEND_DST_ALPHA
fun BLEND_SRC_ALPHA
fun CONSTANT_COLOR
fun ONE_MINUS_CONSTANT_COLOR
fun CONSTANT_ALPHA
fun ONE_MINUS_CONSTANT_ALPHA
fun BLEND_COLOR
fun ARRAY_BUFFER
fun ARRAY_BUFFER_BINDING
fun STREAM_DRAW
fun STATIC_DRAW
fun DYNAMIC_DRAW
fun BUFFER_SIZE
fun BUFFER_USAGE
TEXTURE_CUBE_MAP_POSITIVE_Z: Int
val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int
val MAX_CUBE_MAP_TEXTURE_SIZE: Int
val TEXTURE0: Int
val TEXTURE1: Int
val TEXTURE2: Int
val TEXTURE3: Int
val TEXTURE4: Int
val TEXTURE5: Int
val TEXTURE6: Int
val TEXTURE7: Int
val TEXTURE8: Int
val TEXTURE9: Int
val TEXTURE10: Int
val TEXTURE11: Int
val TEXTURE12: Int
val TEXTURE13: Int
val TEXTURE14: Int
val TEXTURE15: Int
val TEXTURE16: Int
val TEXTURE17: Int
val TEXTURE18: Int
val TEXTURE19: Int
val TEXTURE20: Int
val TEXTURE21: Int
val TEXTURE22: Int
val TEXTURE23: Int
val TEXTURE24: Int
val TEXTURE25: Int
val TEXTURE26: Int
val TEXTURE27: Int
val TEXTURE28: Int
val TEXTURE29: Int
val TEXTURE30: Int
val TEXTURE31: Int
val ACTIVE_TEXTURE: Int
val REPEAT: Int
val CLAMP_TO_EDGE: Int
val MIRRORED_REPEAT: Int
val FLOAT_VEC2: Int
val FLOAT_VEC3: Int
val FLOAT_VEC4: Int
val INT_VEC2: Int
val INT_VEC3: Int
val INT_VEC4: Int
val BOOL: Int
val BOOLEAN: Int
val FLOAT_MAT2: Int
val FLOAT_MAT3: Int
val FLOAT_MAT4: Int
val SAMPLER_2D: Int
val SAMPLER_CUBE: Int
val VERTEX_ATTRIB_ARRAY_ENABLED: Int
val VERTEX_ATTRIB_ARRAY_SIZE: Int
val VERTEX_ATTRIB_ARRAY_STRIDE: Int
val VERTEX_ATTRIB_ARRAY_TYPE: Int
val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int
val VERTEX_ATTRIB_ARRAY_POINTER: Int
val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING: Int
val IMPLEMENTATION_COLOR_READ_TYPE: Int
val IMPLEMENTATION_COLOR_READ_FORMAT: Int
val COMPILE_STATUS: Int
val LOW_FLOAT: Int
val MEDIUM_FLOAT: Int
val HIGH_FLOAT: Int
val LOW_INT: Int
val MEDIUM_INT: Int
val HIGH_INT: Int
val FRAMEBUFFER: Int
val RENDERBUFFER: Int
val RGBA4: Int
val RGB5_A1: Int
val RGB565: Int
val DEPTH_COMPONENT16: Int
val STENCIL_INDEX: Int
val STENCIL_INDEX8: Int
val DEPTH_STENCIL: Int
val RENDERBUFFER_WIDTH: Int
val RENDERBUFFER_HEIGHT: Int
val RENDERBUFFER_INTERNAL_FORMAT: Int
val RENDERBUFFER_RED_SIZE: Int
val RENDERBUFFER_GREEN_SIZE: Int
val RENDERBUFFER_ALPHA_SIZE: Int
val RENDERBUFFER_DEPTH_SIZE: Int
val RENDERBUFFER_STENCIL_SIZE: Int
val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int
val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int
val FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int
val FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int
val COLOR_ATTACHMENT0: Int
val DEPTH_ATTACHMENT: Int
val STENCIL_ATTACHMENT: Int
val DEPTH_STENCIL_ATTACHMENT: Int
val NONE: Int
val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int
val FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT: Int
val FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int
val FRAMEBUFFER_UNSUPPORTED: Int
val FRAMEBUFFER_BINDING: Int
val RENDERBUFFER_BINDING: Int
val UNPACK_FLIP_Y_WEBGL: Int
val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int
val CONTEXT_LOST_WEBGL: Int
val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int
val CONTEXT_LOST_WEBGL: Int
val DEPTH_BUFFER_BIT: Int
val STENCIL_BUFFER_BIT: Int
val COLOR_BUFFER_BIT: Int
val POINTS: Int
val LINES: Int
val LINE_LOOP: Int
val LINE_STRIP: Int
val TRIANGLES: Int
val TRIANGLE_STRIP: Int
val TRIANGLE_FAN: Int
val ZERO: Int
val ONE: Int
val SRC_COLOR: Int
val ONE_MINUS_SRC_COLOR: Int

Open Source Used In DNAC Platform Antelopé 4051
```kotlin
val FRAMEBUFFER_BINDING: Int
val RENDERER_BINDING: Int
val MAX_FRAMEBUFFER_SIZE: Int
val INVALID_FRAMEBUFFER_OPERATION: Int
val UNPACK_FLIP_Y_WEBGL: Int
val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int
val CONTEXT_LOST_WEBGL: Int
val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int
val BROWSER_DEFAULT_WEBGL: Int

/**
 * Exposes the JavaScript
 * [WebGLContextEvent](https://developer.mozilla.org/en/docs/Web/API/WebGLContextEvent) to Kotlin
 */
public external open class WebGLContextEvent(type: String, eventInit: WebGLContextEventInit = definedExternally) : Event{
    open val statusMessage: String
    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface WebGLContextEventInit : EventInit {
    var statusMessage: String? /* = "" */
    get() = definedExternally
    set(value) = definedExternally
    val bubbles: Boolean
    get() = false
    set(value) = definedExternally
    val cancelable: Boolean
    get() = false
    set(value) = definedExternally
    get() = definedExternally
    set(value) = definedExternally
    val composed: Boolean
    get() = false
    set(value) = definedExternally
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["statusMessage"] = statusMessage
    o["bubbles"] = bubbles
    o["Cancelable"] = cancelable
    o["Composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 * [ArrayBuffer](https://developer.mozilla.org/en/docs/Web/API/ArrayBuffer) to Kotlin
 */
public external open class ArrayBuffer(length: Int) : BufferDataSource {
    open val byteLength: Int
    fun slice(begin: Int, end: Int = definedExternally): ArrayBuffer
    companion object {
        fun isView(value: Any?): Boolean
    }
}

/**
 * Exposes the JavaScript
 * [ArrayBufferView](https://developer.mozilla.org/en/docs/Web/API/ArrayBufferView) to Kotlin
 */
public external interface ArrayBufferView : BufferDataSource {
    val buffer: ArrayBuffer
    val byteOffset: Int
    val byteLength: Int
    fun set(array: ArrayBufferView, offset: Int = definedExternally)
    fun set(array: Array<Byte>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): ArrayBufferView
    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Int8Array.get(index: Int): Byte = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Int8Array.set(index: Int, value: Byte) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
 * [Uint8Array](https://developer.mozilla.org/en/docs/Web/API/Uint8Array) to Kotlin
 */
public external open class Uint8Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint8Array)
    constructor(array: Array<Byte>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int
    override val byteLength: Int
    fun set(array: Uint8Array, offset: Int = definedExternally)
    fun set(array: Array<Byte>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint8Array
    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint8Array.get(index: Int): Byte = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint8Array.set(index: Int, value: Byte) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
 * [Uint8ClampedArray](https://developer.mozilla.org/en/docs/Web/API/Uint8ClampedArray) to Kotlin
 */
public external open class Uint8ClampedArray : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint8ClampedArray)
    constructor(array: Array<Byte>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int
    override val byteLength: Int
    fun set(array: Uint8ClampedArray, offset: Int = definedExternally)
    fun set(array: Array<Byte>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint8ClampedArray
    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint8ClampedArray.get(index: Int): Byte = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint8ClampedArray.set(index: Int, value: Byte) { asDynamic()[index] = value }
```
external abstract class CSSGroupingRule : CSSRule {
    open var cssRules: CSSRuleList
    fun insertRule(rule: String, index: Int): Int
    fun deleteRule(index: Int)
    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin
 */
public external abstract class CSSMediaRule : CSSGroupingRule {
    open var media: MediaList
    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin
 */
public external abstract class CSSPageRule : CSSGroupingRule {
    open var selectorText: String
    open var style: CSSStyleDeclaration
    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSMarginRule](https://developer.mozilla.org/en/docs/Web/API/CSSMarginRule) to Kotlin
 */
public external abstract class CSSMarginRule : CSSRule {
    open var name: String
    open var style: CSSStyleDeclaration
    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin
 */
public external abstract class CSSNamespaceRule : CSSRule {
    open var namespaceURI: String
    open var prefix: String
    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin
 */
public external abstract class CSSStyleDeclaration : ItemArrayLike<String> {
    open var cssText: String
    open var parentRule: CSSRule?
    open var cssFloat: String
    open var alignContent: String
    open var alignItems: String
    open var alignSelf: String
    open var animation: String
    open var animationDelay: String
    open var animationDirection: String
    open var animationFillMode: String
    open var animationIterationCount: String
    open var animationName: String
    open var animationPlayState: String
    open var animationTimingFunction: String
    open var backfaceVisibility: String
    open var background: String
    open var backgroundAttachment: String
    open var backgroundClip: String
    open var backgroundColor: String
    open var backgroundImage: String
    open var backgroundOrigin: String
    open var backgroundPosition: String
    open var backgroundSize: String
    open var border: String
    open var borderBottom: String
    open var borderBottomColor: String
    open var borderBottomLeftRadius: String
    open var borderBottomRightRadius: String
    open var borderBottomStyle: String
    open var borderBottomWidth: String
    open var borderCollapse: String
    open var borderColor: String
    open var borderImage: String
    open var borderImageOutset: String
    open var borderImageRepeat: String
    open var borderImageSlice: String
    open var borderImageSource: String
    open var borderImageWidth: String
    open var borderCollapse: String
    open var borderLeft: String
    open var borderLeftColor: String
    open var borderLeftHeightStyle: String
    open var borderLeftWidth: String
    open var borderRadius: String
    open var borderRight: String
    open var borderRightColor: String
    open var borderRightStyle: String
    open var borderRightWidth: String
    open var borderSpacing: String
    open var borderStyle: String
    open var borderTop: String
    open var borderTopColor: String
    open var borderTopLeftRadius: String
    open var borderTopRightRadius: String
    open var borderTopStyle: String
    open var borderTopWidth: String
    open var boxDecorationBreak: String
    open var
CAPTURING_PHASE: Short
    val AT_TARGET: Short
    val BUBBLING_PHASE: Short

    }\n
    public external interface MouseEventInit : EventModifierInit {\n        var screenX: Int? /* = 0 */ \n        get() =
        definedExternally\n        set(value) = definedExternally\n        var screenY: Int? /* = 0 */ \n        get() =
        definedExternally\n        set(value) = definedExternally\n        var clientX: Int? /* = 0 */ \n        get() =
        definedExternally\n        set(value) = definedExternally\n        var clientY: Int? /* = 0 */ \n        get() =
        definedExternally\n        set(value) = definedExternally\n        var button: Short? /* = 0 */ \n        get() =
        definedExternally\n        set(value) = definedExternally\n        var buttons: Short? /* = 0 */ \n        get() =
        definedExternally\n        set(value) = definedExternally\n        var relatedTarget: EventTarget? /* = null */ \n        get() =
        definedExternally\n        set(value) = definedExternally\n        var region: String? /* = null */ \n        get() =
        definedExternally\n        set(value) = definedExternally\n        @kotlin.internal.InlineOnly
        public inline fun
                val o = js("({})")
                o["screenX"] = screenX
                o["screenY"] = screenY
                o["clientX"] = clientX
                o["clientY"] = clientY
                o["button"] = button
                o["buttons"] = buttons
                o["relatedTarget"] = relatedTarget
                o["region"] = region
                o["ctrlKey"] = ctrlKey
                o["shiftKey"] = shiftKey
                o["altKey"] = altKey
                o["metaKey"] = metaKey
                o["modifierAltGraph"] = modifierAltGraph
                o["modifierCapsLock"] = modifierCapsLock
                o["modifierFn"] = modifierFn
                o["modifierHyper"] = modifierHyper
                o["modifierNumLock"] = modifierNumLock
                o["modifierScrollLock"] = modifierScrollLock
                o["modifierSuper"] = modifierSuper
                o["modifierSymbol"] = modifierSymbol
                o["view"] = view
                o["detail"] = detail
                o["bubbles"] = bubbles
                o["cancelable"] = cancelable
                o["composed"] = composed
                return o
            }

    public external interface EventModifierInit : UIEventInit {
        var ctrlKey: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var shiftKey: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var altKey: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var metaKey: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierAltGraph: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierCapsLock: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierFn: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierFnLock: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierHyper: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierNumLock: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierScrollLock: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierSuper: Boolean?
        get() = definedExternally
        set(value) = definedExternally
        var modifierSymbol: Boolean?
        get() = definedExternally
        set(value) = definedExternally
    }

    @kotlin.internal.InlineOnly
    public inline fun
            val o = js("({})")
            o["ctrlKey"] = ctrlKey
            o["shiftKey"] = shiftKey
            o["altKey"] = altKey
            o["metaKey"] = metaKey
            o["modifierAltGraph"] = modifierAltGraph
            o["modifierCapsLock"] = modifierCapsLock
            o["modifierFn"] = modifierFn
            o["modifierHyper"] = modifierHyper
            o["modifierNumLock"] = modifierNumLock
            o["modifierScrollLock"] = modifierScrollLock
            o["modifierSuper"] = modifierSuper
            o["modifierSymbol"] = modifierSymbol
            o["view"] = view
            o["detail"] = detail
            o["bubbles"] = bubbles
            o["cancelable"] = cancelable
            o["composed"] = composed
            return o
        }


Open Source Used In DNAC Platform Antelope  4061
```javascript
o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph

    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierFnLock"] = modifierFnLock
    o["modifierHyper"] = modifierHyper
    o["modifierNumLock"] = modifierNumLock

    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper

    o["modifierSymbol"] = modifierSymbol
    o["modifierSymbolLock"] = modifierSymbolLock

    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [WheelEvent](https://developer.mozilla.org/en/docs/Web/API/WheelEvent) to Kotlin
 */
public external open class WheelEvent(type: String, eventInitDict: WheelEventInit = definedExternally) : MouseEvent {
    open val deltaX: Double
    open val deltaY: Double
    open val deltaZ: Double
    open val deltaMode: Int

    companion object {
        val DOM_DELTA_PIXEL: Int
        val DOM_DELTA_LINE: Int
        val DOM_DELTA_PAGE: Int
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface WheelEventInit : MouseEventInit {
        var deltaX: Double? /* = 0.0 */
            get() = definedExternally
            set(value) = definedExternally
        var deltaY: Double? /* = 0.0 */
            get() = definedExternally
            set(value) = definedExternally
        var deltaZ: Double? /* = 0.0 */
            get() = definedExternally
            set(value) = definedExternally
        var deltaMode: Int? /* = 0 */
            get() = definedExternally
            set(value) = definedExternally
    }

    @kotlin.internal.InlineOnly
        val o = js("({})")
        o["deltaX"] = deltaX
        o["deltaY"] = deltaY
        o["deltaZ"] = deltaZ
        o["deltaMode"] = deltaMode
        o["view"] = view
        o["detail"] = detail
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }

    /**
     * Exposes the JavaScript [InputEvent](https://developer.mozilla.org/en/docs/Web/API/InputEvent) to Kotlin
     */
    public external open class InputEvent(type: String, eventInitDict: InputEventInit = definedExternally) : UIEvent {
        open val data: String
        open val isComposing: Boolean

        companion object {
            val NONE: Short
            val CAPTURING_PHASE: Short
            val AT_TARGET: Short
            val BUBBLING_PHASE: Short
        }

        public external interface InputEventInit : UIEventInit {
            var data: String? /* = "" */
                get() = definedExternally
                set(value) = definedExternally
            var isComposing: Boolean? /* = false */
                get() = definedExternally
                set(value) = definedExternally
        }

        @kotlin.internal.InlineOnly
            val o = js("({})")
            o["data"] = data
            o["isComposing"] = isComposing
            o["view"] = view
            o["detail"] = detail
            o["bubbles"] = bubbles
            o["cancelable"] = cancelable
            o["composed"] = composed
            return o
        }
    }
```
[KeyboardEvent](https://developer.mozilla.org/en/docs/Web/API/KeyboardEvent) to Kotlin

```kotlin
/* */
public external open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent {
    open val key: String
    open val code: String
    open val location: Int
    open val ctrlKey: Boolean
    open val altKey: Boolean
    open val metaKey: Boolean
    open val repeat: Boolean
    open val isComposing: Boolean
    open val charCode: Int
    open val keyCode: Int
    open val which: Int
    fun getModifierState(keyArg: String): Boolean

    companion object {
        val DOM_KEY_LOCATION_STANDARD: Int
        val DOM_KEY_LOCATION_LEFT: Int
        val DOM_KEY_LOCATION_RIGHT: Int
        val DOM_KEY_LOCATION_NUMPAD: Int
    }
}

public external interface KeyboardEventInit : EventModifierInit {
    var key: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var code: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var location: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var repeat: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var isComposing: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var ctrlKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var shiftKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var altKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var metaKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierAltGraph: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierCapsLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierFn: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierHyper: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierNumLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierScrollLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierSuper: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierSymbol: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var view: Window? = null
    var detail: Int
    var bubbles: Boolean
    var cancelable: Boolean
    var composed: boolean
    fun initEvent(type: String, bubbles: Boolean, cancelable: Boolean)
}
```

[CompositionEvent](https://developer.mozilla.org/en/docs/Web/API/CompositionEvent) to Kotlin

```kotlin
/* */
public external open class CompositionEvent(type: String, eventInitDict: CompositionEventInit = definedExternally) : UIEvent {
    open val data: String

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface CompositionEventInit : UIEventInit {
    var data: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["data"] = data
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}
```

[Event](https://developer.mozilla.org/en/docs/Web/API/Event) to Kotlin

```kotlin
Event(type: String, eventInitDict: EventInit = definedExternally) {
    var open val type: String?
    var open val target: EventTarget?
    var open val currentTarget: EventTarget?
    var open val eventPhase: Short?
    var open val bubbles: Boolean?
    var open val cancelable: Boolean?
    var open val defaultPrevented: Boolean?
    var open val composed: Boolean?
    var open val isTrusted: Boolean?
    var open val timestamp: Number?
    var open val fun composedPath():
        Array<EventTarget>?
        fun stopPropagation():
        fun stopImmediatePropagation():
        fun preventDefault():
        fun initEvent(type: String?, bubbles: Boolean, cancelable: Boolean?)
}
```

Open Source Used in DNAC Platform Antelope 4063
public abstract class EventTarget {
    fun addEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)
    fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)
    fun removeEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)
    fun removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)
    fun dispatchEvent(event: Event): Boolean
}

public interface EventListener {
    fun handleEvent(event: Event)
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. *
 * NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT! /* See github.com/kotlin/dukat for details
 */

package org.w3c.dom

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.xhr.*

public abstract class HTMLAllCollection {
    open val length: Int
    fun item(nameOrIndex: String = definedExternally): UnionElementOrHTMLCollection?
    fun namedItem(name: String): UnionElementOrHTMLCollection?
}

@kotlin.internal.InlineOnly
public inline operator fun HTMLAllCollection.get(index: Int): Element? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun HTMLAllCollection.get(name: String): UnionElementOrHTMLCollection? =
    asDynamic()[name]

/**
 * Exposes the JavaScript 
 [HTMLFormControlsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLFormControlsCollection) to Kotlin
 */
public abstract class HTMLFormControlsCollection : HTMLCollection

/**
 * Exposes the JavaScript 
 [RadioNodeList](https://developer.mozilla.org/en/docs/Web/API/RadioNodeList) to Kotlin
 */
public abstract class RadioNodeList : NodeList,
    UnionElementOrRadioNodeList {
    open var value: String
}

/**
 * Exposes the JavaScript 
 [HTMLOptionsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionsCollection) to Kotlin
 */
public abstract class HTMLOptionsCollection : HTMLCollection {
    override var length: Int
    open
    var selectedIndex: Int
    fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before:
        dynamic = definedExternally)
    fun remove(index: Int)
}

@kotlin.internal.InlineOnly
public inline operator fun HTMLOptionsCollection.set(index: Int, option: HTMLOptionElement?) {
    asDynamic()[index] = option
}

/**
 * Exposes the JavaScript 
 [HTMLElement](https://developer.mozilla.org/en/docs/Web/API/HTMLElement) to Kotlin
 */
public abstract class HTMLElement : Element, GlobalEventHandlers,
    DocumentAndElementEventHandlers, ElementContentEditable, ElementCSSInlineStyle {
    open var title: String
    open var lang: String
    open var translate: Boolean
    open var dir: String
    open val dataset: DOMStringMap
    open var hidden: Boolean
    open var tabIndex: Int
    open var accessKey: String
    open val accessKeyLabel: String
    open var draggable: Boolean
    open val dropzone: DOMTokenList
    open var contextMenu: HTMLMenuElement?
    open var spellcheck: Boolean
    open var innerText: String
    open val offsetParent: Element?
    open val offsetTop: Int
    open val offsetLeft: Int
    open val offsetWidth: Int
    open val offsetHeight: Int
    fun click()
    fun focus()
    fun blur()
    fun forceSpellCheck()

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
    }
Open Source Used In DNAC Platform Antelope

Antelope 4065
**Open Source Used In DNAC Platform Antelope**

4067
Open Source Used In DNAC Platform Antelope 4068

```kotlin
short
val PROCESSING_INSTRUCTION_NODE: short
val COMMENT_NODE: short
val DOCUMENT_NODE: short
val DOCUMENT_TYPE_NODE: short
val DOCUMENT_FRAGMENT_NODE: short
val NOTATION_NODE: short
val DOCUMENT_POSITION_DISCONNECTED: short
val DOCUMENT_POSITION_PRECEDING: short
val DOCUMENT_POSITION_FOLLOWING: short
val DOCUMENT_POSITION_CONTAINS: short
val DOCUMENT_POSITION_CONTAINED_BY: short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: short
}
}

/**
 * Exposes the JavaScript [HTMLPreElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPreElement) to Kotlin */
public external abstract class HTMLPreElement : HTMLElement {
    open var width: Int

    companion object {

    val ELEMENT_NODE: short
    val ATTRIBUTE_NODE: short
    val TEXT_NODE: short
    val CDATA_SECTION_NODE: short
    val ENTITY_REFERENCE_NODE: short
    val ENTITY_NODE: short
    val PROCESSING_INSTRUCTION_NODE: short
    val COMMENT_NODE: short
    val DOCUMENT_NODE: short
    val DOCUMENT_TYPE_NODE: short
    val DOCUMENT_FRAGMENT_NODE: short
    val NOTATION_NODE: short
    val DOCUMENT_POSITION_DISCONNECTED: short
    val DOCUMENT_POSITION_PRECEDING: short
    val DOCUMENT_POSITION_FOLLOWING: short
    val DOCUMENT_POSITION_CONTAINS: short
    val DOCUMENT_POSITION_CONTAINED_BY: short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: short
    }
}

/**
 * Exposes the JavaScript [HTMLQuoteElement](https://developer.mozilla.org/en/docs/Web/API/HTMLQuoteElement) to Kotlin */
public external abstract class HTMLQuoteElement : HTMLElement {
    open var cite: String

    companion object {

    val ELEMENT_NODE: short
    val ATTRIBUTE_NODE: short
    val TEXT_NODE: short
    val CDATA_SECTION_NODE: short
    val ENTITY_REFERENCE_NODE: short
    val ENTITY_NODE: short
    val PROCESSING_INSTRUCTION_NODE: short
    val COMMENT_NODE: short
    val DOCUMENT_NODE: short
    val DOCUMENT_TYPE_NODE: short
    val DOCUMENT_FRAGMENT_NODE: short
    val NOTATION_NODE: short
    val DOCUMENT_POSITION_DISCONNECTED: short
    val DOCUMENT_POSITION_PRECEDING: short
    val DOCUMENT_POSITION_FOLLOWING: short
    val DOCUMENT_POSITION_CONTAINS: short
    val DOCUMENT_POSITION_CONTAINED_BY: short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: short
        }
}

/**
 * Exposes the JavaScript [HTMLOListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOListElement) to Kotlin */
public external abstract class HTMLOListElement : HTMLElement {
    open var reversed: Boolean
    open var start: Int
    open var type: String
    open var compact: Boolean

    companion object {

    val ELEMENT_NODE: short
    val ATTRIBUTE_NODE: short
    val TEXT_NODE: short
    val CDATA_SECTION_NODE: short
    val ENTITY_REFERENCE_NODE: short
    val ENTITY_NODE: short
    val PROCESSING_INSTRUCTION_NODE: short
    val COMMENT_NODE: short
    val DOCUMENT_NODE: short
    val DOCUMENT_TYPE_NODE: short
    val DOCUMENT_FRAGMENT_NODE: short
    val NOTATION_NODE: short
    val DOCUMENT_POSITION_DISCONNECTED: short
    val DOCUMENT_POSITION_PRECEDING: short
    val DOCUMENT_POSITION_FOLLOWING: short
    val DOCUMENT_POSITION_CONTAINS: short
    val DOCUMENT_POSITION_CONTAINED_BY: short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: short
        }
}

/**
 * Exposes the JavaScript [HTMLUListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUListElement) to Kotlin */
public external abstract class HTMLUListElement : HTMLElement {
    open var compact: Boolean
    open var type: String

    companion object {

    val ELEMENT_NODE: short
    val ATTRIBUTE_NODE: short
        val TEXT_NODE: short
        val CDATA_SECTION_NODE: short
        val ENTITY_REFERENCE_NODE: short
        val ENTITY_NODE: short
        val PROCESSING_INSTRUCTION_NODE: short
        val COMMENT_NODE: short
    val DOCUMENT_NODE: short
    val DOCUMENT_TYPE_NODE: short
    val DOCUMENT_FRAGMENT_NODE: short
    val NOTATION_NODE: short
    val DOCUMENT_POSITION_DISCONNECTED: short
    val DOCUMENT_POSITION_PRECEDING: short
    val DOCUMENT_POSITION_FOLLOWING: short
    val DOCUMENT_POSITION_CONTAINS: short
    val DOCUMENT_POSITION_CONTAINED_BY: short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: short
        }
```
DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript
[HTMLDataElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataElement) to Kotlin
 */
public external abstract class HTMLDataElement : HTMLElement {
    open var value: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
[HTMLTimeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTimeElement) to Kotlin
 */
public external abstract class HTMLTimeElement : HTMLElement {
    open var dateTime: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
[HTMLSpanElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSpanElement) to Kotlin
 */
public external abstract class HTMLSpanElement : HTMLElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
[HTMLBRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBRElement) to Kotlin
 */
public external abstract class HTMLBRElement : HTMLElement {
    open var clear: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
Open Source Used In DNAC Platform Antelope 4071
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

### HTMLTrackElement

Public abstract class HTMLTrackElement : HTMLElement {
open var kind: String
open var src: String
open var srclang: String
open var label: String
open var default: Boolean
open val readyState: Short
open val track: TextTrack

Companion object {
val NONE: Short
val LOADING: Short
val LOADED: Short
val ERROR: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLTrackElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTrackElement) to Kotlin
 */
public external abstract class HTMLTrackElement : HTMLElement {
open var kind: String
open var src: String
open var srclang: String
open var label: String
open var default: Boolean
open val readyState: Short
open val track: TextTrack

Companion object {
val NONE: Short
val LOADING: Short
val LOADED: Short
val ERROR: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLMediaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement) to Kotlin
 */
public external abstract class HTMLMediaElement : HTMLElement {
open val error: MediaError?
open var src: String
open var srcObject: dynamic
open val currentSrc: String
open var crossOrigin: String?
open val networkState: Short
open var preload: String
open val buffered: TimeRanges
open val readyState: Short
open var seeking: Boolean
open var currentTime: Double
open var duration: Double
open var paused: Boolean
open var defaultPlaybackRate: Double
open var loop: Boolean
open var controls: Boolean
open var volume: Double
open var muted: Boolean
open var defaultMuted: Boolean
open var audioTracks: AudioTrackList
open var videoTracks: VideoTrackList
open val textTracks: TextTrackList
fun load()fun canPlayType(type: String): CanPlayTypeResult
fun fastSeek(time: Double)
fun getStartDate(): dynamic
fun play(): Promise<Unit>
fun pause()
fun addTextTrack(kind: TextTrackKind, label: String = definedExternally, language: String = definedExternally): TextTrack

Companion object {
val NETWORK_EMPTY: Short
val NETWORK_IDLE: Short
val NETWORK_LOADING: Short
val NETWORK_NO_SOURCE: Short
val HAVE_NOTHING: Short
val HAVE_METADATA: Short
val HAVE_CURRENT_DATA: Short
val HAVE_FUTURE_DATA: Short
val HAVE_ENOUGH_DATA: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [MediaError](https://developer.mozilla.org/en/docs/Web/API/MediaError) to Kotlin
 */
public external abstract class MediaError {
open val code: Short

Companion object {
val MEDIA_ERR_ABORTED: Short
val MEDIA_ERR_NETWORK: Short
val MEDIA_ERR_DECODE: Short
val MEDIA_ERR_SRC_NOT_SUPPORTED: Short
}

/**
 * Exposes the JavaScript [AudioTrackList](https://developer.mozilla.org/en/docs/Web/API/AudioTrackList) to Kotlin
 */
public external abstract class AudioTrackList : EventTarget {
open val length: Int
open var onchange: ((Event) -> dynamic)?
open var onaddtrack: ((TrackEvent) -> dynamic)?
open var onremovetrack: ((TrackEvent) -> dynamic)?


fun getTrackById(id: String): AudioTrack?
}

@kotlin.internal.InlineOnly
public inline operator fun AudioTrackList.get(index: Int): AudioTrack? = asDynamic()[index]

/**
 * Exposes the JavaScript [AudioTrack](https://developer.mozilla.org/en/docs/Web/API/AudioTrack) to Kotlin
 */
public external abstract class VideoTrackList : EventTarget {
    open val length: Int
    open val selectedIndex: Int
    open var onchange: (Event) -> dynamic
    open var onaddtrack: (TrackEvent) -> dynamic
    open var onremovetrack: (TrackEvent) -> dynamic
    fun getTrackById(id: String): VideoTrack?
}

@kotlin.internal.InlineOnly
public inline operator fun VideoTrackList.get(index: Int): VideoTrack? = asDynamic()[index]

/**
 * Exposes the JavaScript [VideoTrack](https://developer.mozilla.org/en/docs/Web/API/VideoTrack) to Kotlin
 */
public external abstract class VideoTrack : UnionAudioTrackOrTextTrackOrVideoTrack {
    open val id: String
    open val kind: String
    open val label: String
    open val language: String
    open var selected: Boolean
}

/**
 * Exposes the JavaScript [TextTrack](https://developer.mozilla.org/en/docs/Web/API/TextTrack) to Kotlin
 */
public external abstract class TextTrack : EventTarget, UnionAudioTrackOrTextTrackOrVideoTrack {
    open val kind: TextTrackKind
    open val label: String
    open val language: String
    open val id: String
    open val inBandMetadataTrackDispatchType: String
    open val cues: TextTrackCueList?
    open val activeCues: TextTrackCueList?
    open var oncuechange: (Event) -> dynamic
    fun addCue(cue: TextTrackCue)
    fun removeCue(cue: TextTrackCue)
}

/**
 * Exposes the JavaScript [TimeRanges](https://developer.mozilla.org/en/docs/Web/API/TimeRanges) to Kotlin
 */
public external abstract class TimeRanges {
    open val length: Int
    fun start(index: Int): Double
    fun end(index: Int): Double
}

/**
 * Exposes the JavaScript [TrackEvent](https://developer.mozilla.org/en/docs/Web/API/TrackEvent) to Kotlin
 */
public external open class TrackEvent(type: String, eventInitDict: TrackEventInit = definedExternally) : Event {
    open val track: UnionAudioTrackOrTextTrackOrVideoTrack?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface TrackEventInit : EventInit {
    var track: UnionAudioTrackOrTextTrackOrVideoTrack? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    unreachable
    return false
}

/**
 * Exposes the JavaScript [HTMLMapElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMapElement) to Kotlin
 */
public external abstract class HTMLMapElement : HTMLElement {
    open var name: String
    open val areas: HTMLCollection

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
    }
}

Open Source Used In DNAC Platform Antelope  4075
/**
 * Exposes the JavaScript [HTMLTableColElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableColElement) to Kotlin
 *
 */
public external abstract class HTMLTableColElement : HTMLElement {
    open var span: Int
    open var align: String
    open var ch: String
    open var chOff: String
    open var vAlign: String
    open var width: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
var bgColor: String

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLFormElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFormElement) to Kotlin *
 */
public external abstract class HTMLFormElement : HTMLElement {
    open var acceptCharset: String
    open var action: String
    open var autocomplete: String
    open var enctype: String
    open var encoding: String
    open var method: String
    open var name: String
    open var noValidate: Boolean
    open var target: String
    open var elements: HTMLFormControlsCollection
    open val length: Int
    fun submit()
    fun reset()
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

@kotlin.internal.InlineOnly
public inline operator fun HTMLFormElement.get(index: Int): Element? =
    asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun HTMLFormElement.get(name: String): UnionElementOrRadioNodeList? =
    asDynamic()[name]

/**
 * Exposes the JavaScript [HTMLLabelElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLabelElement) to Kotlin *
 */
public external abstract class HTMLLabelElement : HTMLElement {
    open val form: HTMLFormElement?
    open var htmlFor: String
    open val control: HTMLElement?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to Kotlin *
 */
public external abstract class HTMLInputElement : HTMLElement {
    open var accept: String
    open var alt: String
    open var autocomplete: String
    open var autofocus: Boolean
    open var defaultChecked: Boolean
    open var checked: Boolean
    open var dirName: String
    open var disabled: Boolean
    open var formAction: String
    open var formEnctype: String
    open var formMethod: String
    open var formNoValidate: Boolean
    open var formTarget: String
    open var height: Int
    open var indeterminate: Boolean
    open var inputMode: String
    open var max: String
    open var maxLenLength: Int
    open var min: String
    open var
setCustomValidity(error: String?)
override fun item(index: Int): Element?
companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITYREFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENTTYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINEDBY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

@kotlin.internal.InlineOnly
public inline operator fun HTMLSelectElement.get(index: Int): Element? =
asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun HTMLSelectElement.set(index: Int, option: HTMLOptionElement?) {
asDynamic()[index] = option
}

/**
 * Exposes the JavaScript [HTMLDataListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataListElement) to Kotlin
 */
public external abstract class HTMLDataListElement : HTMLElement {
    open val options: HTMLCollection

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITYREFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINEDBY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLOptGroupElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptGroupElement) to Kotlin
 */
public external abstract class HTMLOptGroupElement : HTMLElement, UnionHTMLOptGroupElementOrHTMLOptionElement {
    open var disabled: Boolean
    open var label: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITYREFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINEDBY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLOptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionElement) to Kotlin
 */
public external abstract class HTMLOptionElement : HTMLElement, UnionHTMLOptGroupElementOrHTMLOptionElement {
    open var disabled: Boolean
    open val form: HTMLFormElement?
    open var label: String
    open var defaultSelected: Boolean
    open var selected: Boolean
    open var value: String
    open var text: String
    open val index: Int

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITYREFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINEDBY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript
 * [HTMLProgressElement](https://developer.mozilla.org/en/docs/Web/API/HTMLProgressElement) to Kotlin
 */
public external abstract class HTMLProgressElement : HTMLElement {
    open var value: Double
    open var max: Double
    open val position: Double
    open val labels: NodeList

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
 * [HTMLMeterElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMeterElement) to Kotlin
 */
public external abstract class HTMLMeterElement : HTMLElement {
    open var value: Double
    open var min: Double
    open var max: Double
    open var low: Double
    open var high: Double
    open var optimum: Double
    open val labels: NodeList

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
 * [HTMLFieldSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFieldSetElement) to Kotlin
 */
public external abstract class HTMLFieldSetElement : HTMLElement {
    open var disabled: Boolean
    open val form: HTMLFormElement?
    open var name: String
    open val type: String
    open val elements: HTMLCollection
    open val willValidate: Boolean
    open val validity: ValidityState
    open val validationMessage: String
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
 * [HTMLLegendElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLegendElement) to Kotlin
 */

`public external abstract class HTMLLegendElement : HTMLElement {
    open val form: HTMLFormElement?
    open var align: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript [ValidityState](https://developer.mozilla.org/en/docs/Web/API/ValidityState) to Kotlin
 */
public external abstract class ValidityState {
    open val valueMissing: Boolean
    open val typeMismatch: Boolean
    open val patternMismatch: Boolean
    open val tooLong: Boolean
    open val tooShort: Boolean
    open val rangeUnderflow: Boolean
    open val rangeOverflow: Boolean
    open val stepMismatch: Boolean
    open val badInput: Boolean
    open val customError: Boolean
    open val valid: Boolean
}

/**
 * Exposes the JavaScript [HTMLDetailsElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDetailsElement) to Kotlin
 */
public external abstract class HTMLDetailsElement : HTMLElement {
    open var open: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

public external abstract class HTMLMenuElement : HTMLElement {
    open var type: String
    open var label: String
    open var compact: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

public external abstract class HTMLMenuItemElement : HTMLElement {
    open var type: String
    open var label: String
    open var icon: String
    open var disabled: Boolean
    open var checked: Boolean
    open var radiogroup: String
    open var default: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

Open Source Used In DNAC Platform Antelope 4083
Open Source Used In DNAC Platform Antelope 4084
**Exposes the JavaScript [HTMLSlotElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSlotElement) to Kotlin**

```kotlin
public abstract external class HTMLSlotElement : HTMLElement {
    open var name: String
    fun assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>

    companion object {
        val ELEMENT_NODE: Short
        valATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public interface AssignedNodesOptions {
    var flatten: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun AssignedNodesOptions(flatten: Boolean? = false): AssignedNodesOptions {
    val o = js("({})")
    o["flatten"] = flatten
    return o
}

**Exposes the JavaScript [HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement) to Kotlin**

```kotlin
public abstract external class HTMLCanvasElement : HTMLElement, CanvasImageSource, TexImageSource {
    open var width: Int
    open var height: Int
    fun getContext(contextId: String, vararg arguments: Any?): RenderingContext?
    fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally): String
    fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? = definedExternally)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public interface CanvasRenderingContext2DSettings {
    var alpha: Boolean? /* = true */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {
    val o = js("({})")
    o["alpha"] = alpha
    return o
}

**Exposes the JavaScript [CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D) to Kotlin**

```kotlin
public abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform, CanvasCompositing, CanvasImageSmoothing, CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters, CanvasRect, CanvasDrawPath, CanvasUserInterface, CanvasText, CanvasDrawImage, CanvasHitRegion, CanvasImageData, CanvasPathDrawingStyles, CanvasTextDrawingStyles, CanvasPath, RenderingContext {
    open val canvas: HTMLCanvasElement
}

public interface CanvasState {
    fun save()
    fun restore()
}

public interface CanvasTransform {
    fun scale(x: Double, y: Double)
    fun rotate(angle: Double)
    fun translate(x: Double, y: Double)
    fun transform(a: Double, b: Double, c: Double, d: Double, e: Double, f: Double)
    fun getTransform(): DOMMatrix
    fun setTransform(a: Double, b: Double, c: Double, d: Double, e: Double, f: Double)
    fun resetTransform()
}

public interface CanvasCompositing {
    var globalAlpha: Double
    var globalCompositeOperation: String
}

public interface CanvasImageSmoothing {
    var imageSmoothingEnabled: Boolean
    var imageSmoothingQuality: ImageSmoothingQuality
}
```
Open Source Used In DNAC Platform Antelope

```kotlin
public external abstract class TextMetrics {
    open val width: Double
    open val actualBoundingBoxLeft: Double
    open val actualBoundingBoxRight: Double
    open val fontBoundingBoxAscent: Double
    open val fontBoundingBoxDescent: Double
    open valideographicBaseline: Double
}

public external interface HitRegionOptions {
    var path: Path2D? /* = null */
    var fillRule: CanvasFillRule? /* = CanvasFillRule.NONZERO */
    var id: String? /* = "" */
    var parentID: String? /* = null */
    var cursor: String? /* = "inherit" */
    var label: String? /* = null */
    var role: String? /* = null */
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["path"] = path
    o["fillRule"] = fillRule
    o["id"] = id
    o["parentID"] = parentID
    o["cursor"] = cursor
    o["label"] = label
    o["role"] = role
    return o
}

/**
 * Exposes the JavaScript [ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to Kotlin
 */
public external open class ImageData : ImageBitmapSource, TexImageSource {
    constructor(sw: Int, sh: Int)
    constructor(data: Uint8ClampedArray, sw: Int, sh: Int = definedExternally)
    open val width: Int
    open val height: Int
    open val data: Uint8ClampedArray
}

/**
 * Exposes the JavaScript [Path2D](https://developer.mozilla.org/en/docs/Web/API/Path2D) to Kotlin
 */
public external open class Path2D() : CanvasPath {
    constructor(path: Path2D)
    constructor(paths: Array<Path2D>, fillRule: CanvasFillRule = definedExternally)
    constructor(d: String)
    fun addPath(path: Path2D, transform: dynamic = definedExternally)
    override fun closePath()
    override fun moveTo(x: Double, y: Double)
    override fun lineTo(x: Double, y: Double)
    override fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)
    override fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)
    override fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)
    override fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)
    override fun rect(x: Double, y: Double, w: Double, h: Double)
    override fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally */) override fun ellipse(x: Double, y: Double, radiusX: Double, radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally */) override fun oval(x: Double, y: Double, w: Double, h: Double) override fun rect(x: Double, y: Double, w: Double, h: Double) override fun oval(x: Double, y: Double, w: Double, h: Double)
}
```

---

[ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to Kotlin

```kotlin
public external abstract class ImageBitmapRenderingContext {
    open val canvas: HTMLCanvasElement
    fun transferFromImageBitmap(bitmap: ImageBitmap?)
}

public external interface ImageBitmapRenderingContextSettings {
    var alpha: Boolean? /* = true */
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["alpha"] = alpha
    return o
}

/**
 * Exposes the JavaScript [CustomElementRegistry](https://developer.mozilla.org/en/docs/Web/API/CustomElementRegistry) to Kotlin
 */
public external abstract class CustomElementRegistry {
    fun define(name: String, constructor: () -> dynamic, options: ElementDefinitionOptions = definedExternally)
    fun get(name: String): Any
    fun whenDefined(name: String): Promise<Unit>
}

public external interface ElementDefinitionOptions {
    var extends: String?
}
```

---

[HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement) fun transferFromImageBitmap(bitmap: ImageBitmap?)
**PageTransitionEvent**

```kotlin
public external open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) : Event {
    open val persisted: Boolean?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface PageTransitionEventInit : EventInit {
    var persisted: Boolean? /* = false */
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["persisted"] = persisted
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}
```

**BeforeUnloadEvent**

```kotlin
public external open class BeforeUnloadEvent : Event {
    var returnValue: String?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external abstract class ApplicationCache : EventTarget {
    open val status: Short
    open var onchecking: ((Event) -> dynamic)?
    open var onerror: ((Event) -> dynamic)?

    open var onnoupdate: ((Event) -> dynamic)?
    open var ondownloading: ((Event) -> dynamic)?
    open var onprogress: ((ProgressEvent) -> dynamic)?
    open var onupdateready: ((Event) -> dynamic)?
    open var oncached: ((Event) -> dynamic)?
    open var onobsolete: ((Event) -> dynamic)?
    fun update()
    fun abort()
    fun swapCache()

    companion object {
        val UNCACHED: Short
        val IDLE: Short
        val CHECKING: Short
        val DOWNLOADING: Short
        val UPDATEREADY: Short
        val OBSOLETE: Short
    }
}
```

**NavigatorOnLine**

```kotlin
public external interface NavigatorOnLine {
    val onLine: Boolean?
}
```

**ErrorEvent**

```kotlin
public external open class ErrorEvent(type: String, eventInitDict: ErrorEventInit = definedExternally) : Event {
    open val message: String?
    open val filename: String?
    open val lineno: Int?
    open val colno: Int?
    open val error: Any?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface ErrorEventInit : EventInit {
        var message: String? /* = "" */
        var filename: String? /* = "" */
        var lineno: Int? /* = 0 */
        var colno: Int? /* = 0 */
        var error: Any? /* = null */
    }

    @kotlin.internal.InlineOnly
        val o = js("({})")
        o["message"] = message
        o["filename"] = filename
        o["lineno"] = lineno
        o["colno"] = colno
        o["error"] = error
        return o
    }
```

**PromiseRejectionEvent**

```kotlin
public external open class PromiseRejectionEvent(type: String, eventInitDict: PromiseRejectionEventInit) : Event {
    open val promise: Promise<Any?>
    open val reason: Any?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface PromiseRejectionEventInit : EventInit {
        var promise: Promise<Any?>?
        var reason: Any?
    }
```

**Promise**

```kotlin
open val promise: Promise<Any?>

@kotlin.internal.InlineOnly
public inline fun Promise(then: ((resolve: ((Promise<any>?) -> dynamic), reject: ((Promise<any>?) -> dynamic)) -> dynamic): Promise) {
    val o = js("{}")
    o["then"] = then
    return o
}
```

**Open Source Used In DNAC Platform Antelope**

4090
set(value) = definedExternally

var onplay: ((Event) -> dynamic)?
    get() = definedExternally

var onplaying: ((Event) -> dynamic)?
    get() = definedExternally

var onprogress: ((ProgressEvent) -> dynamic)?
    get() = definedExternally

var onratechange: ((Event) -> dynamic)?
    get() = definedExternally

var onreset: ((Event) -> dynamic)?
    get() = definedExternally

var onresize: ((Event) -> dynamic)?
    get() = definedExternally

var onscroll: ((Event) -> dynamic)?
    get() = definedExternally

var onseeked: ((Event) -> dynamic)?
    get() = definedExternally

var onseeking: ((Event) -> dynamic)?
    get() = definedExternally

var onselect: ((Event) -> dynamic)?
    get() = definedExternally

var onshow: ((Event) -> dynamic)?
    get() = definedExternally

var onstalled: ((Event) -> dynamic)?
    get() = definedExternally

var onsubmit: ((Event) -> dynamic)?
    get() = definedExternally

var onsuspend: ((Event) -> dynamic)?
    get() = definedExternally

var ontimeupdate: ((Event) -> dynamic)?
    get() = definedExternally

var ontoggle: ((Event) -> dynamic)?
    get() = definedExternally

var onvolumechange: ((Event) -> dynamic)?
    get() = definedExternally

var onwaiting: ((Event) -> dynamic)?
    get() = definedExternally

var ongotpointercapture: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onlostpointercapture: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onpointerdown: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onpointermove: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onpointercancel: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onpointerover: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onpointerout: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onpointerleave: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onpointerenter: ((PointerEvent) -> dynamic)?
    get() = definedExternally

var onpagehide: ((PageTransitionEvent) -> dynamic)?
    get() = definedExternally

var onpageshow: ((PageTransitionEvent) -> dynamic)?
    get() = definedExternally

var onpopstate: ((PopStateEvent) -> dynamic)?
    get() = definedExternally

var onrejectionhandled: ((Event) -> dynamic)?
    get() = definedExternally

var onstorage: ((StorageEvent) -> dynamic)?
    get() = definedExternally

var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?
    get() = definedExternally

var onunload: ((Event) -> dynamic)?
    get() = definedExternally

/* Exposes the JavaScript [WindowEventHandlers](https://developer.mozilla.org/en/docs/Web/API/WindowEvent Handlers) to Kotlin */

var onafterprint: ((Event) -> dynamic)?
    get() = definedExternally

var onbeforeprint: ((Event) -> dynamic)?
    get() = definedExternally

var onbeforeunload: ((BeforeUnloadEvent) -> String)?
    get() = definedExternally

var onhashchange: ((HashChangeEvent) -> dynamic)?
    get() = definedExternally

var onlanguagechange: ((Event) -> dynamic)?
    get() = definedExternally

var onmessage: ((MessageEvent) -> dynamic)?
    get() = definedExternally

var onoffline: ((Event) -> dynamic)?
    get() = definedExternally

var ononlinechange: ((Event) -> dynamic)?
    get() = definedExternally

var onpageshow: ((PageTransitionEvent) -> dynamic)?
    get() = definedExternally

var onpagehide: ((PageTransitionEvent) -> dynamic)?
    get() = definedExternally

var onpageload: ((Event) -> dynamic)?
    get() = definedExternally

var onstorage: ((StorageEvent) -> dynamic)?
    get() = definedExternally

var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?
    get() = definedExternally

var onunload: ((Event) -> dynamic)?
    get() = definedExternally

Open Source Used In DNAC Platform Antelope 4092
definedExternally

```kotlin
open public external interface DocumentAndElementEventHandlers {
    var oncropy:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncut:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onpaste:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript
 */
public external interface DocumentAndElementEventHandlers {
    var oncopy:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncut:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onpaste:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript
 */
public external interface DocumentAndElementEventHandlers {
    var oncopy:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncut:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onpaste:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript
 */
public external interface DocumentAndElementEventHandlers {
    var oncopy:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncut:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onpaste:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript
 */
public external interface DocumentAndElementEventHandlers {
    var oncopy:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncut:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onpaste:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript
 */
public external interface DocumentAndElementEventHandlers {
    var oncopy:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncut:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onpaste:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript
 */
public external interface DocumentAndElementEventHandlers {
    var oncopy:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncut:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onpaste:
        ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
}
```
class Plugin : ItemArrayLike<MimeType> {"in open val name: String\n open val description: String\n open val filename: String\n override fun item(index: Int): MimeType?\n  fun namedItem(name: String): MimeType?\n \n public inline operator fun Plugin.get(index: Int): MimeType? = asDynamic()\n  \n public inline operator fun Plugin.get(name: String): MimeType? = asDynamic()  

/**
 * Exposes the JavaScript
 [MimeType](https://developer.mozilla.org/en/docs/Web/API/MimeType) to Kotlin\n */
 public abstract class MimeType {
    open val type: String
    open val description: String
    open val suffixes: String
    open val enabledPlugin: Plugin
    
/**
 * Exposes the JavaScript
 [ImageBitmap](https://developer.mozilla.org/en/docs/Web/API/ImageBitmap) to Kotlin\n */
 public abstract class ImageBitmap : CanvasImageSource, TexImageSource {"in open val width: Int\n open val height: Int\n fun close()\n 
public external interface ImageBitmapOptions {
    var imageOrientation: ImageOrientation? /* = ImageOrientation.NONE */
        get() = definedExternally\n        set(value) = definedExternally
    var premultiplyAlpha: PremultiplyAlpha? /* = PremultiplyAlpha.DEFAULT */
        get() = definedExternally\n        set(value) = definedExternally
    var colorSpaceConversion: ColorSpaceConversion? /* = ColorSpaceConversion.DEFAULT */
        get() = definedExternally\n        set(value) = definedExternally
    var resizeWidth: Int? /* = undefined */
        get() = definedExternally
        set(value) = definedExternally
    var resizeHeight: Int? /* = undefined */
        get() = definedExternally
        set(value) = definedExternally
    var resizeQuality: ResizeQuality? /* = ResizeQuality.LOW */
        get() = definedExternally
        set(value) = definedExternally
    
/**
 * Exposes the JavaScript
 [MessageEvent](https://developer.mozilla.org/en/docs/Web/API/MessageEvent) to Kotlin\n */
 public open class MessageEvent(type: String, eventInitDict: MessageEventInit = definedExternally) : Event {"in open val data: Any?\n open val origin: String\n open val lastEventId: String?\n open val source: UnionMessagePortOrWindowProxy?\n open val ports: Array<MessagePort>?\n fun initMessageEvent(type: String, bubbles: Boolean, cancelable: Boolean, data: Any?, origin: String, lastEventId: String, source: UnionMessagePortOrWindowProxy?, ports: Array<MessagePort>?): MessageEventInit = definedExternally
}
Open Source Used In DNAC Platform Antelope 4096

[SharedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/SharedWorkerGlobalScope) to Kotlin

```kotlin
public external abstract class SharedWorkerGlobalScope : WorkerGlobalScope {
    open val name: String
    open val applicationCache: ApplicationCache
    open var onconnect: ((Event) -> dynamic)?
    fun close()
}

/**
 * Exposes the JavaScript [AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to Kotlin
 */
public external interface AbstractWorker {
    var onerror: ((Event) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var credentials: RequestCredentials
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript [Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin
 */
public external open class Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {
    var onmessage: ((MessageEvent) -> dynamic)?
    override var onerror: ((Event) -> dynamic)?
    fun terminate()
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}

public external interface WorkerOptions {
    var type: WorkerType
        get() = definedExternally
        set(value)
    var credentials: RequestCredentials
        get() = definedExternally
        set(value)
}

@kotlin.internal.InlineOnly
public inline fun WorkerOptions(type: WorkerType = WorkerType.CLASSIC, credentials: RequestCredentials = RequestCredentials.OMIT): WorkerOptions {
    val o = js("({})")
    o["type"] = type
    o["credentials"] = credentials
    return o
}

/**
 * Exposes the JavaScript [SharedWorker](https://developer.mozilla.org/en/docs/Web/API/SharedWorker) to Kotlin
 */
public external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {
    open val port: MessagePort
    override var onerror: ((Event) -> dynamic)?
}

/**
 * Exposes the JavaScript [NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to Kotlin
 */
public external interface NavigatorConcurrentHardware {
    val hardwareConcurrency: Number
}

/**
 * Exposes the JavaScript [WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin
 */
public external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorConcurrentHardware {
    open val serviceWorker: ServiceWorkerContainer
}

/**
 * Exposes the JavaScript [WorkerLocation](https://developer.mozilla.org/en/docs/Web/API/WorkerLocation) to Kotlin
 */
public external abstract class WorkerLocation {
    open val href: String
    open val origin: String
    open val protocol: String
    open val host: String
    open val hostname: String
    open val port: String
    open val pathname: String
    open val search: String
    open val hash: String
}

/**
 * Exposes the JavaScript [Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to Kotlin
 */
public external abstract class Storage {
    open val length: Int
    fun key(index: Int): String?
    fun removeItem(key: String)
    fun clear()
    fun getItem(key: String): String?
    fun setItem(key: String, value: String)
}

@kotlin.internal.InlineOnly
public inline operator fun Storage.get(key: String): String? = asDynamic()[key]

@kotlin.internal.InlineOnly
public inline operator fun Storage.set(key: String, value: String) { asDynamic()[key] = value
}

/**
 * Exposes the JavaScript [WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin
 */
public external interface WindowSessionStorage {
    val sessionStorage: Storage
}

/**
 * Exposes the JavaScript [WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage) to Kotlin
 */
public external interface WindowLocalStorage {
    val localStorage: Storage
}

/**
 * Exposes the JavaScript [StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin
 */
public external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) : Event {
    open val key: String
    open val oldValue: String
    open val newValue: String
    open val url: String
    open val storageArea: Storage
    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
        val AT_TARGET: Short
    }
    open val source: Storage
    open val sourceType: String
    open val type: String
    open val value: String
}

public external interface StorageEventInit : EventInit {
    var key: String
    var source: Storage
    var sourceType: String
    var type: String
    var value: String
}
```
definedExternally

```kotlin
    var newValue: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally

    var url: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally

    var storageArea: Storage? /* = null */
        get() = definedExternally
        set(value) = definedExternally

    var @kotlin.internal.InlineOnly open var align: String
        get() = "left"
        set(value) = "left"

    var open var alt: String
        get() = "alt"
        set(value) = "alt"

    var archive: String
        get() = "archive"
        set(value) = "archive"

    var code: String
        get() = "code"
        set(value) = "code"

    var codeBase: String
        get() = "codeBase"
        set(value) = "codeBase"

    var height: String
        get() = "height"
        set(value) = "height"

    var hspace: Int
        get() = "hspace"
        set(value) = "hspace"

    var name: String
        get() = "name"
        set(value) = "name"

    var _object: String
        get() = "_object"
        set(value) = "_object"

    var vspace: Int
        get() = "vspace"
        set(value) = "vspace"

    var width: String
        get() = "width"
        set(value) = "width"
```

**HTML Marquee Element**

```kotlin
    public external abstract class HTMLMarqueeElement : HTMLElement {
        open var behavior: String
            get() = "behavior"
            set(value) = "behavior"

        open var bgColor: String
            get() = "bgColor"
            set(value) = "bgColor"

        open var direction: String
            get() = "direction"
            set(value) = "direction"

        open var height: String
            get() = "height"
            set(value) = "height"

        open var hspace: Int
            get() = "hspace"
            set(value) = "hspace"

        open var loop: Int
            get() = "loop"
            set(value) = "loop"

        open var scrollAmount: Int
            get() = "scrollAmount"
            set(value) = "scrollAmount"

        open var scrollDelay: Int
            get() = "scrollDelay"
            set(value) = "scrollDelay"

        open var trueSpeed: Boolean
            get() = "trueSpeed"
            set(value) = "trueSpeed"

        open var vspace: Int
            get() = "vspace"
            set(value) = "vspace"

        open var width: String
            get() = "width"
            set(value) = "width"

        fun start()
        fun stop()```

**HTML Frame Set Element**

```kotlin
    public external abstract class HTMLFrameSetElement : HTMLElement, WindowEventHandlers {
        open var cols: String
            get() = "cols"
            set(value) = "cols"

        open var rows: String
            get() = "rows"
            set(value) = "rows"
```

---

Open Source Used In DNAC Platform Antelope 4097
definedExternally
set(value) = definedExternally
var subtree: Boolean? /* = false */
get() =
definedExternally
set(value) = definedExternally
var attributeOldValue: Boolean?
get() =
definedExternally
set(value) = definedExternally
var characterDataOldValue: Boolean? get() =
definedExternally
set(value) = definedExternally
var attributeFilter: Array<String>? get() =

@kotlin.internal.InlineOnly
public inline fun
    val o = js("({})")
    o["childList"] = childList
    o["attributes"] = attributes
    o["characterData"] = characterData
    o["subtree"] = subtree
    o["attributeOldValue"] = attributeOldValue
    o["characterDataOldValue"] = characterDataOldValue
    o["attributeFilter"] = attributeFilter
    return o
}

/**
 * Exposes the JavaScript
[MutationRecord](https://developer.mozilla.org/en/docs/Web/API/MutationRecord) to Kotlin
*/
public external abstract class MutationRecord {
    open val type: String
    open val target: Node
    open val addedNodes: NodeList
    open val removedNodes: NodeList
    open val previousSibling: Node?
    open val nextSibling: Node?
    open val attributeName: String?
    open val attributeNamespace: String?
    open val oldValue: String?
}

/**
 * Exposes the JavaScript
[Node](https://developer.mozilla.org/en/docs/Web/API/Node) to Kotlin
*/
public external abstract class Node : EventTarget {
    open val nodeType: Short
    open val nodeName: String
    open val baseURI: String
    open val isConnected: Boolean
    open val ownerDocument: Document?
    open val parentNode: Node?
    open val parentElement: Element?
    open val childNodes: NodeList
    open val firstChild: Node?
    open val lastChild: Node?
    open val previousSibling: Node?
    open val nextSibling: Node?
    open var nodeValue: String?
    open var textContent: String?
    fun getRootNode(options: GetRootNodeOptions = definedExternally): Node
    fun hasChildNodes(): Boolean
    fun normalize()
    fun cloneNode(deep: Boolean = definedExternally): Node
    fun isEqualNode(otherNode: Node?): Boolean
    fun isSameNode(otherNode: Node?): Boolean
    fun compareDocumentPosition(other: Node): Short
    fun contains(other: Node?): Boolean
    fun lookupPrefix(namespace: String?): String?
    fun lookupNamespaceURI(prefix: String?): String?
    fun isDefaultNamespace(namespace: String?): Boolean
    fun insertBefore(node: Node, child: Node?): Node
    fun appendChild(node: Node): Node
    fun replaceChild(node: Node, child: Node): Node
    fun removeChild(child: Node): Node

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface GetRootNodeOptions {
    var composed: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun GetRootNodeOptions(composed: Boolean? = false): GetRootNodeOptions {
    val o = js("({})")
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
[Document](https://developer.mozilla.org/en/docs/Web/API/Document) to Kotlin
*/
    open val implementation: DOMImplementation
    open val URL: String
    open val documentURI: String
    open val origin: String
    open val compatMode: String
    open val characterSet: String
    open val inputEncoding: String
    open val contentEncoding: String
    open val doctype: DocumentType
    open val elementId: String
    open val location: Location
    open val domain: String
    open val referrer: String
    var title: String
    var cookie: String
    var lastModified: String
    var readyState: DocumentReadyState
}
HTMLCollection

override val firstElementChild: Element?
override val lastElementChild: Element?

override val childElementCount: Int

fun getElementsByTagName(qualifiedName: String): HTMLCollection
fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection
fun createElement(localName: String, options: ElementCreationOptions = definedExternally): Element
fun createDocumentFragment(): DocumentFragment
fun createTextNode(data: String): Text
fun createCDATASection(data: String): CDATASection
fun createComment(data: String): Comment
fun createProcessingInstruction(target: String, data: String): ProcessingInstruction
fun importNode(node: Node, deep: Boolean = definedExternally): Node
fun adoptNode(node: Node): Node
fun createAttribute(localName: String): Attr
fun createAttributeNS(namespace: String?, qualifiedName: String): Attr
fun createEvent(`interface`: String): Event
fun createRange(): Range
fun createNodeIterator(root: Node, whatToShow: Int = definedExternally, filter: (Node) -> Short?): NodeIterator
fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter: NodeFilter?): TreeWalker
fun getElementsByName(elementName: String): NodeList
fun open(type: String = definedExternally, replace: String = definedExternally): Document
fun open(url: String, name: String, features: String): Window
fun close(): Document
fun write(vararg text: String)
fun writeln(vararg text: String)
fun hasFocus(): Boolean
fun execCommand(commandId: String, showUI: Boolean = definedExternally, value: String = definedExternally): Boolean
fun queryCommandEnabled(commandId: String): Boolean
fun queryCommandIndeterm(commandId: String): Boolean
fun queryCommandState(commandId: String): Boolean
fun queryCommandSupported(commandId: String): Boolean
fun queryCommandValue(commandId: String): String
fun clear(): Document
fun captureEvents(): Document
fun releaseEvents(): Document

fun elementFromPoint(x: Double, y: Double): Element?
fun elementsFromPoint(x: Double, y: Double): Array<Element>
fun caretPositionFromPoint(x: Double, y: Double): CaretPosition?
fun createTouchList(vararg touches: Touch): TouchList
fun exitFullscreen(): Promise<Unit>

fun getElementById(elementId: String): Element?
fun prepend(vararg nodes: dynamic)
fun append(vararg nodes: dynamic)
fun querySelector(selectors: String): Element?
fun querySelectorAll(selectors: String): NodeList
fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>
fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */) -> DOMQuad
fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */) -> DOMRect
fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */) -> DOMPoint
fun companion object {

val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NDE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

fun asDynamic(): dynamic
}@kotlin.internal.InlineOnly
public inline operator fun Document.get(name: String): dynamic = asDynamic()
open val host: Element
override val fullscreenElement: Element?
companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [Element](https://developer.mozilla.org/en/docs/Web/API/Element) to Kotlin
 */
public external abstract class Element : Node, ParentNode, NonDocumentTypeChildNode, ChildNode, Slotable, GeometryUtils,
UnionElementOrHTMLCollection, UnionElementOrRadioNodeList, UnionElementOrMouseEvent,
UnionElementOrProcessingInstruction {
open val namespaceURI: String?
open val prefix: String?
open val localName: String
open val tagName: String
open var id: String
open var className: String
open val classList: DOMTokenList
open var slot: String
open val attributes: NamedNodeMap
open val shadowRoot: ShadowRoot?
open var scrollTop: Double
open var scrollLeft: Double
open val scrollWidth: Int
open val scrollHeight: Int
open val clientTop: Int
open val clientLeft: Int
open val clientWidth: Int
open val clientHeight: Int
open var innerHTML: String
open var outerHTML: String
fun hasAttributes(): Boolean
fun getAttributeNames(): Array<String>
fun getAttribute(qualifiedName: String): String?
fun getAttributeNS(namespace: String?, localName: String): String?
fun setAttribute(qualifiedName: String, value: String)
fun setAttributeNS(namespace: String?, qualifiedName: String, value: String)
fun removeAttribute(qualifiedName: String)
fun removeAttributeNS(namespace: String?, localName: String)
fun hasAttribute(qualifiedName: String): Boolean
fun hasAttributeNS(namespace: String?, localName: String): Boolean
fun getAttributeNode(qualifiedName: String): Attr?
fun getAttributeNodeNS(namespace: String?, localName: String): Attr?
fun setAttributeNode(attr: Attr)
fun setAttributeNodeNS(attr: Attr)
fun removeAttributeNode(attr: Attr)
fun attachShadow(init: ShadowRootInit): ShadowRoot
fun closest(selectors: String): Element
fun matches(selectors: String): Boolean
fun getElementsByTagName(qualifiedName: String): HTMLCollection
fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection
fun insertAdjacentElement(where: String, element: Element): Element?
fun insertAdjacentText(where: String, data: String)
fun getClientRects(): Array<DOMRect>
fun getBoundingClientRect(): DOMRect
fun scrollIntoView()
fun scrollIntoView(arg: dynamic)
fun scroll(x: Double, y: Double)
fun scrollTo(options: ScrollToOptions = definedExternally)
fun scrollBy(x: Double, y: Double)
fun scrollBy(options: ScrollToOptions = definedExternally)
fun insertAdjacentHTML(position: String, text: String)
fun setPointerCapture(pointerId: Int)
fun releasePointerCapture(pointerId: Int)
fun requestFullscreen(): Promise<Unit>
companion object {
val ELEMENT_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short

DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

ShadowRootInit { 
    var mode: ShadowRootMode?
    get() = definedExternally
    set(value) = definedExternally
}:

public external interface ShadowRootInit {
    var mode: ShadowRootMode?
    get() = definedExternally
    set(value) = definedExternally
}

public external interface ShadowRootInit {
    var mode: ShadowRootMode?
    get() = definedExternally
    set(value) = definedExternally
}

/**
 * Exposes the JavaScript [CharacterData](https://developer.mozilla.org/en/docs/Web/API/CharacterData) to Kotlin
 */
public external abstract class CharacterData : Node, NonDocumentTypeChildNode, ChildNode {
    open var data: String
    open val length: Int
    fun substringData(offset: Int, count: Int): String
    fun appendData(data: String)
    fun insertData(offset: Int, data: String)
    fun deleteData(offset: Int, count: Int)
    fun replaceData(offset: Int, count: Int, data: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [Text](https://developer.mozilla.org/en/docs/Web/API/Text) to Kotlin
 */
public external open class Text(data: String = definedExternally) : CharacterData, Slotable, GeometryUtils {
    open val wholeText: String
    override val assignedSlot: HTMLSlotElement?
    override val previousElementSibling: Element?
    override val nextElementSibling: Element?
    fun splitText(offset: Int): Text
    override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>?
    override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
    override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
    override fun convertPointFromNode(point: DOMPoint, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint
    override fun before(vararg nodes: dynamic)
    override fun after(vararg nodes: dynamic)
    override fun replaceWith(vararg nodes: dynamic)
    ...
override fun remove()

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin
 */
public external open class CDATASection : Text {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin
 */
public external abstract class ProcessingInstruction : CharacterData, LinkStyle, UnionElementOrProcessingInstruction {
    open val target: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [Comment](https://developer.mozilla.org/en/docs/Web/API/Comment) to Kotlin
 */
public external open class Comment(data: String = definedExternally) : CharacterData {
    override val previousElementSibling: Element?
    override val nextElementSibling: Element?
    override fun before(vararg nodes: dynamic)
    override fun after(vararg nodes: dynamic)
    override fun replaceWith(vararg nodes: dynamic)
    override fun remove()

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [Range](https://developer.mozilla.org/en/docs/Web/API/Range) to Kotlin
 */
public external open class Range {
    open val startContainer: Node
    open val startOffset: Int
    open val endContainer: Node
    open val
endOffset: Int
    open val collapsed: Boolean
    open val commonAncestorContainer: Node
    fun setStart(node: Node, offset: Int)
    fun setEnd(node: Node)
    fun setStartBefore(node: Node, offset: Int)
    fun setStartAfter(node: Node)
    fun setEndBefore(node: Node)
    fun setEndAfter(node: Node)
    fun collapse(toStart: Boolean = definedExternally)
    fun selectNode(node: Node)
    fun selectNodeContents(node: Node)
    fun compareBoundaryPoints(how: Short, sourceRange: Range): Short
    fun deleteContents()
    fun extractContents(): DocumentFragment
    fun cloneContents(): DocumentFragment
    fun insertNode(node: Node)
    fun surroundContents(newParent: Node)
    fun cloneRange(): Range
    fun detach()
    fun isPointInRange(node: Node, offset: Int): Boolean
    fun comparePoint(node: Node, offset: Int): Short
    fun intersectsNode(node: Node): Boolean

    companion object {
        val START_TO_START: Short
        val START_TO_END: Short
        val END_TO_END: Short
        val END_TO_START: Short
    }
}

/**
 * Exposes the JavaScript

[NodeIterator](https://developer.mozilla.org/en/docs/Web/API/NodeIterator) to Kotlin
 */
public external abstract class NodeIterator {
    open val root: Node
    open val referenceNode: Node
    open val pointerBeforeReferenceNode: Boolean
    open val whatToShow: Int
    open val filter: NodeFilter?
    fun nextNode(): Node?
    fun previousNode(): Node?
    fun detach()
}

/**
 * Exposes the JavaScript

[TreeWalker](https://developer.mozilla.org/en/docs/Web/API/TreeWalker) to Kotlin
 */
public external abstract class TreeWalker {
    open val root: Node
    open val whatToShow: Int
    open val filter: NodeFilter?
    open var currentNode: Node
    fun parentNode(): Node?
    fun firstChild(): Node?
    fun lastChild(): Node?
    fun previousSibling(): Node?
    fun nextSibling(): Node?
    fun previousNode(): Node?
    fun nextNode(): Node?
}

/**
 * Exposes the JavaScript

[NodeFilter](https://developer.mozilla.org/en/docs/Web/API/NodeFilter) to Kotlin
 */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NodeFilter {
    fun acceptNode(node: Node): Short

    companion object {
        val FILTER_ACCEPT: Short
        val FILTER_REJECT: Short
        val FILTER_SKIP: Short
        val SHOW_ALL: Int
        val SHOW_ELEMENT: Int
        val SHOW_ATTRIBUTE: Int
        val SHOW_TEXT: Int
        val SHOW_CDATA_SECTION: Int
        val SHOW_ENTITY_REFERENCE: Int
        val SHOW_ENTITY: Int
        val SHOW_PROCESSING_INSTRUCTION: Int
        val SHOW_COMMENT: Int
        val SHOW_DOCUMENT: Int
        val SHOW_DOCUMENT_TYPE: Int
        val SHOW_DOCUMENT_FRAGMENT: Int
        val SHOW_NOTATION: Int
    }
}

/**
 * Exposes the JavaScript

[DOMTokenList](https://developer.mozilla.org/en/docs/Web/API/DOMTokenList) to Kotlin
 */
public external abstract class DOMTokenList : ItemArrayLike<String> {
    open var value: String
    fun contains(token: String): Boolean
    fun add(vararg tokens: String)
    fun remove(vararg tokens: String)
    fun toggle(token: String, force: Boolean = definedExternally): Boolean
    fun replace(token: String, newToken: String)
    fun supports(token: String): Boolean
    override fun item(index: Int): String?
}

@kotlin.internal.InlineOnly
public inline operator fun DOMTokenList.get(index: Int): String? = asDynamic()[index]

/**
 * Exposes the JavaScript

[DOMPointReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMPointReadOnly) to Kotlin
 */
public external open class DOMPointReadOnly(x: Double, y: Double, z: Double, w: Double) {
    open val x: Double
    open val y: Double
    open val z: Double
    open val w: Double
    fun matrixTransform(matrix: DOMMatrixReadOnly): DOMPoint
}

/**
 * Exposes the JavaScript

[DOMPoint](https://developer.mozilla.org/en/docs/Web/API/DOMPoint) to Kotlin
 */
public external open class DOMPoint : DOMPointReadOnly {
    constructor(point: DOMPointInit)
    constructor(x: Double = definedExternally, y: Double = definedExternally, z: Double = definedExternally, w: Double = definedExternally)
    override var x: Double
    override var y: Double
    override var z: Double
    override var w: Double
}

/**
 * Exposes the JavaScript

[DOMPointInit](https://developer.mozilla.org/en/docs/Web/API/DOMPointInit) to Kotlin
 */

definedExternally\n    var y: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var z: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var w: Double? /* = 1.0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun DOMPointInit(x: Double? = 0.0, y: Double? = 0.0, w: Double? = 1.0): DOMPointInit {
    val o = js("({})")
    o["x"] = x
    o["y"] = y
    o["w"] = w
    return o
}

/**
 * Exposes the JavaScript
 [DOMRect](https://developer.mozilla.org/en/docs/Web/API/DOMRect) to Kotlin
 */
public external open class DOMRect(x: Double = definedExternally, y: Double = definedExternally, width: Double = definedExternally, height: Double = definedExternally) : DOMRectReadOnly {
    override var x: Double
    override var y: Double
    override var width: Double
    override var height: Double
}

/**
 * Exposes the JavaScript
 [DOMRectReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMRectReadOnly) to Kotlin
 */
public external open class DOMRectReadOnly(x: Double, y: Double, width: Double, height: Double) {
    open val x: Double
    open val y: Double
    open val width: Double
    open val height: Double
    open val top: Double
    open val right: Double
    open val bottom: Double
    open val left: Double
}

public external interface DOMRectInit {
    var x: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var y: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var width: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var height: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun DOMRectInit(x: Double? = 0.0, y: Double? = 0.0, width: Double? = 0.0, height: Double? = 0.0): DOMRectInit {
    val o = js("({})")
    o["x"] = x
    o["y"] = y
    o["width"] = width
    o["height"] = height
    return o
}

public external interface DOMRectList : ArrayLike<DOMRect> {
    override fun item(index: Int): DOMRect?
}

@kotlin.internal.InlineOnly
public inline operator fun DOMRectList.get(index: Int): DOMRect? = asDynamic()[index]

/**
 * Exposes the JavaScript
 [DOMQuad](https://developer.mozilla.org/en/docs/Web/API/DOMQuad) to Kotlin
 */
public external open class DOMQuad {
    constructor(p1: DOMPointInit = definedExternally, p2: DOMPointInit = definedExternally, p3: DOMPointInit = definedExternally, p4: DOMPointInit = definedExternally)
    constructor(rect: DOMRectInit)
    open val p1: DOMPoint
    open val p2: DOMPoint
    open val p3: DOMPoint
    open val p4: DOMPoint
    open val bounds: DOMRectReadOnly
}

/**
 * Exposes the JavaScript
 [DOMMatrixReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMMatrixReadOnly) to Kotlin
 */
public external open class DOMMatrixReadOnly(numberSequence: Array<Double>) {
    open val a: Double
    open val b: Double
    open val c: Double
    open val d: Double
    open val e: Double
    open val f: Double
    open val m11: Double
    open val m12: Double
    open val m13: Double
    open val m14: Double
    open val m21: Double
    open val m22: Double
    open val m23: Double
    open val m24: Double
    open val m31: Double
    open val m32: Double
    open val m33: Double
    open val m34: Double
    open val m41: Double
    open val m42: Double
    open val m43: Double
    open val m44: Double
    open val is2D: Boolean
    open val isIdentity: Boolean
    fun translate(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix
    fun scale(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
    fun scale3d(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix
    fun scaleNonUniform(scaleX: Double, scaleY: Double = definedExternally, scaleZ: Double = definedExternally): DOMMatrix
    fun rotate(angle: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
    fun rotateFromVector(x: Double, y: Double): DOMMatrix
    fun rotateAxisAngle(x: Double, y: Double, z: Double, angle: Double): DOMMatrix
    fun skewX(sx: Double): DOMMatrix
    fun skewY(sy: Double): DOMMatrix
    fun multiply(other: DOMMatrix): DOMMatrix
    fun flipX(): DOMMatrix
    fun flipY(): DOMMatrix
    fun inverse(): DOMMatrix
    fun transformPoint(point: DOMPointInit = definedExternally): DOMPointInit
    fun toFloat32Array(): Float32Array
    fun toFloat64Array(): Float64Array
}
Open Source Used In DNAC Platform Antelope 4109

[DOMMatrix](https://developer.mozilla.org/en/docs/Web/API/DOMMatrix) to Kotlin

```kotlin
public external open
class DOMMatrix() : DOMMatrixReadOnly {
    constructor(transformList: String)
    constructor(array32: Float32Array)
    constructor(array64: Float64Array)
    constructor(numberSequence: Array<Double>)
    override var a: Double
    override var b: Double
    override var c: Double
    override var d: Double
    override var e: Double
    override var f: Double
    override var m11: Double
    override var m12: Double
    override var m13: Double
    override var m14: Double
    override var m21: Double
    override var m22: Double
    override var m23: Double
    override var m24: Double
    override var m31: Double
    override var m32: Double
    override var m33: Double
    override var m34: Double
    override var m41: Double
    override var m42: Double
    override var m43: Double
    override var m44: Double
    fun multiplySelf(other: DOMMatrix): DOMMatrix
    fun preMultiplySelf(other: DOMMatrix): DOMMatrix
    fun translateSelf(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix
    fun scaleSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
    fun scaleSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
    fun scale3dSelf(): DOMMatrix
    fun scaleNonUniformSelf(scaleX: Double, scaleY: Double = definedExternally, scaleZ: Double = definedExternally): DOMMatrix
    fun rotateSelf(angle: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
    fun rotateFromVectorSelf(x: Double, y: Double, z: Double): DOMMatrix
    fun skewXSelf(sx: Double): DOMMatrix
    fun skewYSelf(sy: Double): DOMMatrix
    fun invertSelf(): DOMMatrix
    fun setMatrixValue(transformList: String): DOMMatrix
}

public external
interface ScrollOptions {
    var behavior: ScrollBehavior? /* = ScrollBehavior.AUTO */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ScrollOptions(behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollOptions {
    val o = js("({})")
    o["behavior"] = behavior
    return o
}

/**
 * Exposes the JavaScript [ScrollToOptions](https://developer.mozilla.org/en/docs/Web/API/ScrollToOptions) to Kotlin
 */
public external
interface ScrollToOptions : ScrollOptions {
    var left: Double?
        get() = definedExternally
        set(value) = definedExternally
    var top: Double?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["left"] = left
    o["top"] = top
    o["behavior"] = behavior
    return o
}

/**
 * Exposes the JavaScript [MediaQueryList](https://developer.mozilla.org/en/docs/Web/API/MediaQueryList) to Kotlin
 */
public external abstract class MediaQueryList : EventTarget {
    open val media: String
    open val matches: Boolean
    open var onchange: ((Event) -> dynamic)?
    fun addListener(listener: EventListener?)
    fun addListener(listener: ((Event) -> Unit)?)
    fun removeListener(listener: EventListener?)
    fun removeListener(listener: ((Event) -> Unit)?)
}

/**
 * Exposes the JavaScript [MediaQueryListEvent](https://developer.mozilla.org/en/docs/Web/API/MediaQueryListEvent) to Kotlin
 */
public external open class MediaQueryListEvent(type: String, eventInitDict: MediaQueryListEventInit = definedExternally) : Event {
    open val media: String
    open val matches: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface MediaQueryListEventInit : EventInit {
        var media: String? /* = "" */
            get() = definedExternally
            set(value) = definedExternally
        var matches: Boolean? /* = false */
            get() = definedExternally
            set(value) = definedExternally
    }

    @kotlin.internal.InlineOnly
        val o = js("({})")
        o["media"] = media
        o["matches"] = matches
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }
}
```
open source used in DNAC platform Antelope 4111
Audio(src: String = definedExternally): HTMLAudioElement{
override var onabort: ((Event) -> dynamic)?
onoverride var onblur: ((FocusEvent) -> dynamic)?
onoverride var oncanplay: ((Event) -> dynamic)?
onoverride var oncanplaythrough: ((Event) -> dynamic)?
onoverride var onchange: ((Event) -> dynamic)?
onoverride var oncontextmenu: ((MouseEvent) -> dynamic)?
onoverride var oncuechange: ((Event) -> dynamic)?
onoverride var ondblclick: ((MouseEvent) -> dynamic)?
onoverride var ondrag: ((DragEvent) -> dynamic)?
onoverride var ondragend: ((DragEvent) -> dynamic)?
onoverride var ondragenter: ((DragEvent) -> dynamic)?
onoverride var ondragleave: ((DragEvent) -> dynamic)?
onoverride var ondragover: ((DragEvent) -> dynamic)?
onoverride var ondragstart: ((DragEvent) -> dynamic)?
onoverride var ondurationchange: ((Event) -> dynamic)?
onoverride var onemptied: ((Event) -> dynamic)?
onoverride var onerror: ((any, String, Int, Int, any?) -> dynamic)?
onoverride var onended: ((Event) -> dynamic)?
onoverride var onfocus: ((FocusEvent) -> dynamic)?
onoverride var oninput: ((InputEvent) -> dynamic)?
onoverride var oninvalid: ((Event) -> dynamic)?
onoverride var onkeydown: ((KeyboardEvent) -> dynamic)?
onoverride var onkeypress: ((KeyboardEvent) -> dynamic)?
onoverride var onkeyup: ((KeyboardEvent) -> dynamic)?
onoverride var onload: ((Event) -> dynamic)?
onoverride var onloadeddata: ((Event) -> dynamic)?
onoverride var onloadedmetadata: ((Event) -> dynamic)?
onoverride var onloadend: ((Event) -> dynamic)?
onoverride var onloadevent: ((Event) -> dynamic)?
onoverride var onmousedown: ((MouseEvent) -> dynamic)?
onoverride var onmouseenter: ((MouseEvent) -> dynamic)?
onoverride var onmouseleave: ((MouseEvent) -> dynamic)?
onoverride var onmousemove: ((MouseEvent) -> dynamic)?
onoverride var onmouseout: ((MouseEvent) -> dynamic)?
onoverride var onmouseover: ((MouseEvent) -> dynamic)?
onoverride var onmouseup: ((MouseEvent) -> dynamic)?
onoverride var onwheel: ((WheelEvent) -> dynamic)?
onoverride var onpause: ((Event) -> dynamic)?
onoverride var onplay: ((Event) -> dynamic)?
onoverride var onplaying: ((Event) -> dynamic)?
onoverride var onprogress: ((ProgressEvent) -> dynamic)?
onoverride var onrangedatetimechange: ((Event) -> dynamic)?
onoverride var onresize: ((Event) -> dynamic)?
onoverride var onscroll: ((Event) -> dynamic)?
onoverride var onselect: ((Event) -> dynamic)?
onoverride var onshow: ((Event) -> dynamic)?
onoverride var onstalled: ((Event) -> dynamic)?
onoverride var onsubmit: ((Event) -> dynamic)?
onoverride var onsuspend: ((Event) -> dynamic)?
onoverride var ontimeupdate: ((Event) -> dynamic)?
onoverride var ontoggle: ((Event) -> dynamic)?
onoverride var onvolumechange: ((Event) -> dynamic)?
onoverride var onwaiting: ((Event) -> dynamic)?
onoverride var onvolumedownloadchange: ((Event) -> dynamic)?
onoverride var onvolumemove: ((Event) -> dynamic)?
onoverride var onvolumeup: ((Event) -> dynamic)?
onoverride var onvolumeup: ((Event) -> dynamic)?
onoverride var onpointercancel: ((PointerEvent) -> dynamic)?
onoverride var onpointerdown: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointerup: ((PointerEvent) -> dynamic)?
onoverride var onpointercancel: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
onoverride var onpointermove: ((PointerEvent) -> dynamic)?
}
override fun querySelectorAll(selectors: String): NodeList
override fun before(vararg nodes: dynamic)
override fun replaceWith(vararg nodes: dynamic)
override fun remove()
override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>
override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
override fun convertRectFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint

companion object {
val NETWORK_EMPTY: Short
val NETWORK_IDLE: Short
val NETWORK_LOADING: Short
val NETWORK_NO_SOURCE: Short
val HAVE_NOTHING: Short
val HAVE_FUTURE_DATA: Short
val HAVE_ENOUGH_DATA: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [Option](https://developer.mozilla.org/en/docs/Web/API/Option) to Kotlin
 */
public external open class Option(text: String = definedExternally, value: String = definedExternally, defaultSelected: Boolean = definedExternally, selected: Boolean = definedExternally) : HTMLOptionElement {
override var onabort: ((Event) -> dynamic)?
override var onblur: ((FocusEvent) -> dynamic)?
override var oncancel: ((Event) -> dynamic)?
override var oncanplay: ((Event) -> dynamic)?
override var oncanplaythrough: ((Event) -> dynamic)?
override var onchange: ((Event) -> dynamic)?
override var onclick: ((MouseEvent) -> dynamic)?
override var onclose: ((Event) -> dynamic)?
override var oncontextmenu: ((MouseEvent) -> dynamic)?
override var oncuechange: ((Event) -> dynamic)?
override var ondblclick: ((MouseEvent) -> dynamic)?
override var ondrag: ((DragEvent) -> dynamic)?
override var ondragend: ((DragEvent) -> dynamic)?
override var ondragenter: ((DragEvent) -> dynamic)?
override var ondragexit: ((DragEvent) -> dynamic)?
override var ondragleave: ((DragEvent) -> dynamic)?
override var ondragover: ((DragEvent) -> dynamic)?
override var ondragstart: ((DragEvent) -> dynamic)?
override var ondurationchange: ((Event) -> dynamic)?
override var onemptied: ((Event) -> dynamic)?
override var onended: ((Event) -> dynamic)?
override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?
override var onfocus: ((FocusEvent) -> dynamic)?
override var oninput: ((InputEvent) -> dynamic)?
override var oninvalid: ((Event) -> dynamic)?
override var onkeydown: ((KeyboardEvent) -> dynamic)?
override var onkeypress: ((KeyboardEvent) -> dynamic)?
override var onkeyup: ((KeyboardEvent) -> dynamic)?
override var onload: ((Event) -> dynamic)?
override var onloadend: ((Event) -> dynamic)?
override var onloaderror: ((Event) -> dynamic)?
override var onloadstart: ((Event) -> dynamic)?
override var onload: ((Event) -> dynamic)?
override var onmessage: ((Event) -> dynamic)?
override var onmessageerror: ((Event) -> dynamic)?
override var onmousemove: ((MouseEvent) -> dynamic)?
override var onmouseout: ((MouseEvent) -> dynamic)?
override var onmouseover: ((MouseEvent) -> dynamic)?
override var onmouseup: ((MouseEvent) -> dynamic)?
override var onmessage: ((Event) -> dynamic)?
override var onmessageerror: ((Event) -> dynamic)?
override var onmousemove: ((MouseEvent) -> dynamic)?
override var onmouseout: ((MouseEvent) -> dynamic)?
override var onmouseover: ((MouseEvent) -> dynamic)?
override var onmouseup: ((MouseEvent) -> dynamic)?
override var onmessage: ((Event) -> dynamic)?
override var onmessageerror: ((Event) -> dynamic)?
override var onmousemove: ((MouseEvent) -> dynamic)?
override var onmouseout: ((MouseEvent) -> dynamic)?
override var onmouseover: ((MouseEvent) -> dynamic)?
override onscroll: ((Event) -> dynamic)?
override var onscroll: ((Event) -> dynamic)?
override var onse...
"complete".asDynamic().unsafeCast<DocumentReadyState>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanPlayTypeResult {
    companion object
}

public inline val CanPlayTypeResult.Companion.EMPTY:
    CanPlayTypeResult get() = "".asDynamic().unsafeCast<CanPlayTypeResult>()

public inline val CanPlayTypeResult.Companion.MAYBE: CanPlayTypeResult get() = "maybe".asDynamic().unsafeCast<CanPlayTypeResult>()

public inline val CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() = "probably".asDynamic().unsafeCast<CanPlayTypeResult>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface TextTrackMode {
    companion object
}

public inline val TextTrackMode.Companion.DISABLED: TextTrackMode get() = "disabled".asDynamic().unsafeCast<TextTrackMode>()

public inline val TextTrackMode.Companion.HIDDEN: TextTrackMode get() = "hidden".asDynamic().unsafeCast<TextTrackMode>()

public inline val TextTrackMode.Companion.SHOWING: TextTrackMode get() = "showing".asDynamic().unsafeCast<TextTrackMode>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface TextTrackKind {
    companion object
}

public inline val TextTrackKind.Companion.SUBTITLES: TextTrackKind get() = "subtitles".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.CAPTIONS: TextTrackKind get() = "captions".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.DESCRIPTIONS: TextTrackKind get() = "descriptions".asDynamic().unsafeCast<TextTrackKind>()

/* please, don't implement this
interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface SelectionMode {
    companion object
}

public inline val SelectionMode.Companion.SELECT: SelectionMode get() = "select".asDynamic().unsafeCast<SelectionMode>()

public inline val SelectionMode.Companion.START: SelectionMode get() = "start".asDynamic().unsafeCast<SelectionMode>()

public inline val SelectionMode.Companion.END: SelectionMode get() = "end".asDynamic().unsafeCast<SelectionMode>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasFillRule {
    companion object
}

public inline val CanvasFillRule.Companion.NONZERO: CanvasFillRule get() = "nonzero".asDynamic().unsafeCast<CanvasFillRule>()

public inline val CanvasFillRule.Companion.EVENODD: CanvasFillRule get() = "evenodd".asDynamic().unsafeCast<CanvasFillRule>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ImageSmoothingQuality {
    companion object
}


/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasLineCap {
    companion object
}

public inline val CanvasLineCap.Companion.BUTT: CanvasLineCap get() = "butt".asDynamic().unsafeCast<CanvasLineCap>()

public inline val CanvasLineCap.Companion.ROUND: CanvasLineCap get() = "round".asDynamic().unsafeCast<CanvasLineCap>()

public inline val CanvasLineCap.Companion.Square: CanvasLineCap get() = "square".asDynamic().unsafeCast<CanvasLineCap>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasLineJoin {
    companion object
}

public inline val CanvasLineJoin.Companion.MITER: CanvasLineJoin get() = "miter".asDynamic().unsafeCast<CanvasLineJoin>()

public inline val CanvasLineJoin.Companion.BEvel: CanvasLineJoin get() = "bevel".asDynamic().unsafeCast<CanvasLineJoin>()

public inline val CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() = "round".asDynamic().unsafeCast<CanvasLineJoin>()
CanvasLineCap.get() = "round".asDynamic().unsafeCast<CanvasLineCap>()

CanvasLineCap.Companion.SQUARE: CanvasLineCap get() = "square".asDynamic().unsafeCast<CanvasLineCap>()

public inline val CanvasLineCap.Companion.SQUARE: CanvasLineCap get() = "square".asDynamic().unsafeCast<CanvasLineCap>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasLineJoin
{
    companion object
}

public inline val CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() = "round".asDynamic().unsafeCast<CanvasLineJoin>()

public inline val CanvasLineJoin.Companion.BEVEL: CanvasLineJoin get() = "bevel".asDynamic().unsafeCast<CanvasLineJoin>()

public inline val CanvasLineJoin.Companion.MITER: CanvasLineJoin get() = "miter".asDynamic().unsafeCast<CanvasLineJoin>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasTextAlign
{
    companion object
}

public inline val CanvasTextAlign.Companion.START: CanvasTextAlign get() = "start".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.END: CanvasTextAlign get() = "end".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.LEFT: CanvasTextAlign get() = "left".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.RIGHT: CanvasTextAlign get() = "right".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.CENTER: CanvasTextAlign get() = "center".asDynamic().unsafeCast<CanvasTextAlign>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasTextBaseline
{
    companion object
}

public inline val CanvasTextBaseline.Companion.TOP: CanvasTextBaseline get() = "top".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.HANGING: CanvasTextBaseline get() = "hanging".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.MIDDLE: CanvasTextBaseline get() = "middle".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.ALPHABETIC: CanvasTextBaseline get() = "alphabetic".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.IDEOGRAPHIC: CanvasTextBaseline get() = "ideographic".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.BOTTOM: CanvasTextBaseline get() = "bottom".asDynamic().unsafeCast<CanvasTextBaseline>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasDirection
{
    companion object
}

public inline val CanvasDirection.Companion.LTR: CanvasDirection get() = "ltr".asDynamic().unsafeCast<CanvasDirection>()

public inline val CanvasDirection.Companion.RTL: CanvasDirection get() = "rtl".asDynamic().unsafeCast<CanvasDirection>()

public inline val CanvasDirection.Companion.INHERIT: CanvasDirection get() = "inherit".asDynamic().unsafeCast<CanvasDirection>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollRestoration
{
    companion object
}

public inline val ScrollRestoration.Companion.AUTO: ScrollRestoration get() = "auto".asDynamic().unsafeCast<ScrollRestoration>()


/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface PremultiplyAlpha
{
    companion object
}

public inline val PremultiplyAlpha.Companion.NONE: PremultiplyAlpha get() = "none".asDynamic().unsafeCast<PremultiplyAlpha>()

public inline val PremultiplyAlpha.Companion.FLIPY: PremultiplyAlpha get() = "flipY".asDynamic().unsafeCast<PremultiplyAlpha>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ImageOrientation
{
    companion object
}


/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ImageOrientation
{
    companion object
}


Open Source Used In DNAC Platform Antelope 4116
PremultiplyAlpha get() = "none".asDynamic().unsafeCast<PremultiplyAlpha>()

public inline val PremultiplyAlpha.Companion.PREMULTIPLY: PremultiplyAlpha get() =
"premultiply".asDynamic().unsafeCast<PremultiplyAlpha>()

public inline val PremultiplyAlpha.Companion.DEFAULT: PremultiplyAlpha get() =
"default".asDynamic().unsafeCast<PremultiplyAlpha>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ColorSpaceConversion {
    companion object
}

public inline val ColorSpaceConversion.Companion.NONE:
ColorSpaceConversion get() = "none".asDynamic().unsafeCast<ColorSpaceConversion>()

public inline val ColorSpaceConversion.Companion.DEFAULT:
ColorSpaceConversion get() = "default".asDynamic().unsafeCast<ColorSpaceConversion>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ResizeQuality {
    companion object
}

public inline val ResizeQuality.Companion.PIXELATED: ResizeQuality get() =
"pixelated".asDynamic().unsafeCast<ResizeQuality>()

public inline val ResizeQuality.Companion.LOW: ResizeQuality get() = "low".asDynamic().unsafeCast<ResizeQuality>()

public inline val ResizeQuality.Companion.MEDIUM: ResizeQuality get() =
"medium".asDynamic().unsafeCast<ResizeQuality>()

public inline val ResizeQuality.Companion.HIGH: ResizeQuality get() = "high".asDynamic().unsafeCast<ResizeQuality>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface BinaryType {
    companion object
}

public inline val BinaryType.Companion.BLOB: BinaryType get() =
"blob".asDynamic().unsafeCast<BinaryType>()

public inline val BinaryType.Companion.ARRAYBUFFER: BinaryType get() =
"arraybuffer".asDynamic().unsafeCast<BinaryType>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface WorkerType {
    companion object
}

public inline val WorkerType.Companion.CLASSIC: WorkerType get() =
"classic".asDynamic().unsafeCast<WorkerType>()

public inline val WorkerType.Companion.MODULE: WorkerType get() =
"module".asDynamic().unsafeCast<WorkerType>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ShadowRootMode {
    companion object
}

public inline val ShadowRootMode.Companion.OPEN: ShadowRootMode get() =
"open".asDynamic().unsafeCast<ShadowRootMode>()

public inline val ShadowRootMode.Companion.CLOSED: ShadowRootMode get() =
"closed".asDynamic().unsafeCast<ShadowRootMode>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollBehavior {
    companion object
}

public inline val ScrollBehavior.Companion.AUTO: ScrollBehavior get() =
"auto".asDynamic().unsafeCast<ScrollBehavior>()

public inline val ScrollBehavior.Companion.INSTANT: ScrollBehavior get() =
"instant".asDynamic().unsafeCast<ScrollBehavior>()

public inline val ScrollBehavior.Companion.SMOOTH: ScrollBehavior get() =
"smooth".asDynamic().unsafeCast<ScrollBehavior>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollLogicalPosition {
    companion object
}

public inline val ScrollLogicalPosition.Companion.START: ScrollLogicalPosition get() =
"start".asDynamic().unsafeCast<ScrollLogicalPosition>()

public inline val ScrollLogicalPosition.Companion.CENTER: ScrollLogicalPosition get() =
"center".asDynamic().unsafeCast<ScrollLogicalPosition>()

public inline val ScrollLogicalPosition.Companion.END: ScrollLogicalPosition get() =
"end".asDynamic().unsafeCast<ScrollLogicalPosition>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CSSBoxType {
    companion object
}

public inline val CSSBoxType.Companion.MARGIN: CSSBoxType get() =
"margin".asDynamic().unsafeCast<CSSBoxType>()

public inline val CSSBoxType.Companion.BORDER:
Antelope 4118

CSSBoxType get() = "border".asDynamic().unsafeCast<CSSBoxType>()

CSSBoxType.Companion.PADDING: CSSBoxType get() = "padding".asDynamic().unsafeCast<CSSBoxType>()

public inline val CSSBoxType.Companion.CONTENT: CSSBoxType get() = "content".asDynamic().unsafeCast<CSSBoxType>()

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.
 * NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
 */

// NOTE: THIS FILE IS AUTO-GENERATED,
DO NOT EDIT!

// See github.com/kotlin/dukat for details!

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.*
import org.w3c.dom.*
import org.w3c.dom.parsing.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*
import org.w3c.fetch.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript [Headers](https://developer.mozilla.org/en/docs/Web/API/Headers) to Kotlin
 */
public external open class Headers(init: dynamic = definedExternally) {
    fun append(name: String, value: String)
    fun delete(name: String)
    fun get(name: String): String?
    fun has(name: String): Boolean
    fun set(name: String, value: String)
}

/**
 * Exposes the JavaScript [Body](https://developer.mozilla.org/en/docs/Web/API/Body) to Kotlin
 */
public external interface Body {
    val bodyUsed: Boolean
    fun arrayBuffer(): Promise<ArrayBuffer>
    fun blob(): Promise<Blob>
    fun formData(): Promise<FormData>
    fun json(): Promise<Any?>
    fun text(): Promise<String>
}

/**
 * Exposes the JavaScript [Request](https://developer.mozilla.org/en/docs/Web/API/Request) to Kotlin
 */
public external open class Request(input: dynamic, init: RequestInit = definedExternally) : Body {
    open val method: String
    open val url: String
    open val headers: Headers
    open val type: RequestType
    open val destination: RequestDestination
    open val referrer: String
    open val referrerPolicy: RequestReferrerPolicy
    open val mode: RequestMode
    open val credentials: RequestCredentials
    open val cache: RequestCache
    open val redirect: RequestRedirect
    open val integrity: String
    open val keepalive: Boolean
    override val bodyUsed: Boolean
    fun clone(): Request
    override fun arrayBuffer(): Promise<ArrayBuffer>
    override fun blob(): Promise<Blob>
    override fun formData(): Promise<FormData>
    override fun json(): Promise<Any?>
    override fun text(): Promise<String>
}

public external interface RequestInit {
    var method: String?
    var headers: dynamic
    var body: dynamic
    var referrer: String?
    var referrerPolicy: dynamic
    var mode: RequestMode?
    var credentials: RequestCredentials?
    var cache: RequestCache?
    var redirect: RequestRedirect?
    var integrity: String?
    var keepalive: Boolean?
}

@kotlin.internal.InlineOnly
public inline fun RequestInit(method: String? = undefined,
    headers: dynamic = undefined,
    body: dynamic = undefined,
    referrer: String? = undefined,
    referrerPolicy: dynamic = undefined,
    mode: RequestMode? = undefined,
    credentials: RequestCredentials? = undefined,
    cache: RequestCache? = undefined,
    redirect: RequestRedirect? = undefined,
    integrity: String? = undefined,
    keepalive: Boolean? = undefined,
    window: Any? = undefined): RequestInit {
    val o = js("({})")
    o["method"] = method
    o["headers"] = headers
    o["body"] = body
    o["referrer"] = referrer
    o["referrerPolicy"] = referrerPolicy
    o["mode"] = mode
    o["credentials"] = credentials
    o["cache"] = cache
    o["redirect"] = redirect
    o["integrity"] = integrity
    o["keepalive"] = keepalive
    o["window"] = window
    return o
}

*/
RequestDestination.Companion.SERVICEWORKER: RequestDestination get() = "serviceworker".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.SHAREDWORKER: RequestDestination get() = "sharedworker".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.STYLE: RequestDestination get() = "style".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.WORKER: RequestDestination get() = "worker".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.XSLT: RequestDestination get() = "xslt".asDynamic().unsafeCast<RequestDestination>()

/* please, don’t implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestMode {
    companion object
}

public inline val RequestMode.Companion.NAVIGATE: RequestMode get() = "navigate".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.SAME_ORIGIN: RequestMode get() = "same-origin".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.NO_CORS: RequestMode get() = "no-cors".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.CORS: RequestMode get() = "cors".asDynamic().unsafeCast<RequestMode>()

/* please, don’t implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestCredentials {
    companion object
}

public inline val RequestCredentials.Companion.OMIT: RequestCredentials get() = "omit".asDynamic().unsafeCast<RequestCredentials>()

public inline val RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = "same-origin".asDynamic().unsafeCast<RequestCredentials>()

public inline val RequestCredentials.Companion.INCLUDE: RequestCredentials get() = "include".asDynamic().unsafeCast<RequestCredentials>()

/* please, don’t implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestCache {
    companion object
}

public inline val RequestCache.Companion.DEFAULT: RequestCache get() = "default".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.NO_STORE: RequestCache get() = "no-store".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.RELOAD: RequestCache get() = "reload".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.NO_CACHE: RequestCache get() = "no-cache".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.FORCE_CACHE: RequestCache get() = "force-cache".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = "only-if-cached".asDynamic().unsafeCast<RequestCache>()

/* please, don’t implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestRedirect {
    companion object
}

public inline val RequestRedirect.Companion.FOLLOW: RequestRedirect get() = "follow".asDynamic().unsafeCast<RequestRedirect>()

public inline val RequestRedirect.Companion.ERROR: RequestRedirect get() = "error".asDynamic().unsafeCast<RequestRedirect>()

public inline val RequestRedirect.Companion.MANUAL: RequestRedirect get() = "manual".asDynamic().unsafeCast<RequestRedirect>

/* please, don’t implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ResponseType {
    companion object
}

public inline val ResponseType.Companion.BASIC: ResponseType get() = "basic".asDynamic().unsafeCast<ResponseType>()

public inline val ResponseType.Companion.CORS: ResponseType get() = "cors".asDynamic().unsafeCast<ResponseType>

public inline val ResponseType.Companion.DEFAULT: ResponseType get() = "default".asDynamic().unsafeCast<ResponseType>;

Open Source Used In DNAC Platform Antelope  4121

```kotlin
ResponseType get() = "error".asDynamic().unsafeCast<ResponseType>()

public inline val ResponseType.Companion.OPAQUE: ResponseType get() = "opaque".asDynamic().unsafeCast<ResponseType>()

public inline val ResponseType.Companion.OPAQUEREDIRECT: ResponseType get() = "opaqueredirect".asDynamic().unsafeCast<ResponseType>()

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that
 * can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details
package org.w3c.dom.mediacapture

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.events.*
import org.w3c.dom.media.*
import org.w3c.dom.parsing.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript [MediaStream](https://developer.mozilla.org/en/docs/Web/API/MediaStream) to Kotlin
 */
public external open class MediaStream() : EventTarget {
    constructor(stream: MediaStream)
    constructor(tracks: Array<MediaStreamTrack>)
    open val id: String
    open val active: Boolean
    var onaddtrack: ((MediaStreamTrackEvent) -> dynamic)?
    var onremovetrack: ((MediaStreamTrackEvent) -> dynamic)?
    fun getAudioTracks(): Array<MediaStreamTrack>
    fun getVideoTracks(): Array<MediaStreamTrack>
    fun getTracks(): Array<MediaStreamTrack>
    fun getTrackById(trackId: String): MediaStreamTrack?
    fun addTrack(track: MediaStreamTrack)
    fun removeTrack(track: MediaStreamTrack)
    fun clone(): MediaStream
}

/**
 * Exposes the JavaScript [MediaStreamTrack](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrack) to Kotlin
 */
public external abstract class MediaStreamTrack : EventTarget {
    open val kind: String
    open val id: String
    open val label: String
    open var enabled: Boolean
    open var muted: Boolean
    open var onmute: ((Event) -> dynamic)?
    open var onunmute: ((Event) -> dynamic)?
    open val readyState: MediaStreamTrackState
    open var onended: ((Event) -> dynamic)?
    open var onoverconstrained: ((Event) -> dynamic)?
    fun clone(): MediaStreamTrack
    fun stop()
    fun getCapabilities(): MediaTrackCapabilities
    fun getConstraints(): MediaTrackConstraints
    fun getSettings(): MediaTrackSettings
    fun applyConstraints(constraints: MediaTrackSupportedConstraints = definedExternally): Promise<Unit>
}

/**
 * Exposes the JavaScript [MediaTrackSupportedConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSupportedConstraints) to Kotlin
 */
public external interface MediaTrackSupportedConstraints {
    var width: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var height: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var facingMode: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var resizeMode: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var volume: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var sampleSize: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var echoCancellation: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var autoGainControl: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var latency: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var deviceID: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var groupID: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    override fun @kotlin.internal.InlineOnly
    get(): MediaTrackSupportedConstraints = definedExternally
```
}
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}

public external interface MediaTrackCapabilities {
    var width: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var height: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var facingMode: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
    var resizeMode: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
    var volume: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var sampleSize: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var echoCancellation: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var autoGainControl: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var noiseSuppression: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var latency: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: String?
        get() = definedExternally
        set(value) = definedExternally
    var groupId: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}
o["advanced"] = advanced
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["videoMode"] = videoMode
    o["sampleRate"] = sampleRate
    o["volume"] = volume
    o["frame"] = frame
    o["aspect"] = aspect
    o["volumeControl"] = volumeControl
    o["videoMode"] = videoMode
    o["image"] = image
    o["media"] = media
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["videoMode"] = videoMode
    o["sampleRate"] = sampleRate
    o["volume"] = volume
    o["frame"] = frame
    o["aspect"] = aspect
    o["volumeControl"] = volumeControl
    o["videoMode"] = videoMode
    o["image"] = image
    o["media"] = media
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}

public external interface MediaTrackConstraintSet {
    var width: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var height: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var videoMode: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var volume: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var frame: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var image: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var media: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var latency: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var groupId: dynamic
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["videoMode"] = videoMode
    o["sampleRate"] = sampleRate
    o["volume"] = volume
    o["frame"] = frame
    o["image"] = image
    o["media"] = media
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}

/**
 * Exposes the JavaScript
[MediaTrackSettings](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSettings) to Kotlin
 */
public external interface MediaTrackSettings {
    var width: Int?
        get() = definedExternally
        set(value) = definedExternally
    var height: Int?
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: Double?
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: Double?
        get() = definedExternally
        set(value) = definedExternally
    var videoMode: String?
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: Int?
        get() = definedExternally
        set(value) = definedExternally
    var volume: Double?
        get() = definedExternally
        set(value) = definedExternally
    var frame: Double?
        get() = definedExternally
        set(value) = definedExternally
    var image: Int?
        get() = definedExternally
        set(value) = definedExternally
    var media: String?
        get() = definedExternally
        set(value) = definedExternally
    var latency: Double?
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: Int?
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: String?
        get() = definedExternally
        set(value) = definedExternally
    var groupId: String?
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: Double?
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: Double?
        get() = definedExternally
        set(value) = definedExternally
    var videoMode: String?
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: Int?
        get() = definedExternally
        set(value) = definedExternally
    var volume: Double?
        get() = definedExternally
        set(value) = definedExternally
    var frame: Double?
        get() = definedExternally
        set(value) = definedExternally
    var image: Int?
        get() = definedExternally
        set(value) = definedExternally
    var media: String?
        get() = definedExternally
        set(value) = definedExternally
    var latency: Double?
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: Int?
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: String?
        get() = definedExternally
        set(value) = definedExternally
    var groupId: String?
        get() = definedExternally
        set(value) = definedExternally
*/
Promise<Unit>\nnull/**\n* Exposes the JavaScript
[DoubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin\n*\npublic external interface DoubleRange \n{\n  var max: Double?\n  get() = definedExternally\n  set(value) =
definedExternally\n  var min: Double?\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n
/**\n * Exposes the JavaScript
[DoubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin\n*\npublic external interface DoubleRange \n{\n  var max: Double?\n  get() = definedExternally\n  set(value) =
definedExternally\n  var min: Double?\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n
/**\n * Exposes the JavaScript
[ConstrainBooleanParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainBooleanParameters) to Kotlin\n*\npublic external interface ConstrainBooleanParameters \n{\n  var exact: Boolean?\n  get() = definedExternally\n  set(value) =
definedExternally\n  var ideal: Boolean?\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n
/**\n * Exposes the JavaScript
[ConstrainDOMStringParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainDOMStringParameters) to Kotlin\n*\npublic external interface ConstrainDOMStringParameters \n{\n  var exact: dynamic\n  get() = definedExternally\n  set(value) =
definedExternally\n  var ideal: dynamic\n  get() = definedExternally\n  set(value) =
definedExternally\n}\n
/* please, don't implement this interface!\n*\n@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")\npublic external interface \nMediaStreamTrackState \n{
  companion object
}

public inline val MediaStreamTrackState.Companion.LIVE: MediaStreamTrackState get() = "live".asDynamic().unsafeCast<MediaStreamTrackState>()

public inline val MediaStreamTrackState.Companion.ENDED: MediaStreamTrackState get() = "ended".asDynamic().unsafeCast<MediaStreamTrackState>()

/* please, don't implement this interface!\n*\n*/

Open Source Used In DNAC Platform Antelope 4127

modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view:
    val o = js("({})")
    o["pointerId"] = pointerId
    o["width"] = width
    o["height"] = height
    o["pressure"] = pressure
    o["tangentialPressure"] = tangentialPressure
    o["tiltX"] = tiltX
    o["tiltY"] = tiltY
    o["twist"] = twist
    o["pointerType"] = pointerType
    o["isPrimary"] = isPrimary
    o["screenX"] = screenX
    o["screenY"] = screenY
    o["clientX"] = clientX
    o["clientY"] = clientY
    o["button"] = button
    o["buttons"] = buttons
    o["relatedTarget"] = relatedTarget
    o["region"] = region
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierFnLock"] = modifierFnLock
    o["modifierHyper"] = modifierHyper
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["modifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent) to Kotlin
 */
public external open class PointerEvent(type: String, eventInitDict: PointerEventInit = definedExternally) : MouseEvent {
    open val pointerId: Int
    open val width: Double
    open val height: Double
    open val pressure: Float
    open val tangentialPressure: Float
    open val tiltX: Int
    open val tiltY: Int
    open val twist: Int
    open val pointerType: String
    open val isPrimary: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}
val o = js("({})")
    o["fill"] = fill
    o["stroke"] = stroke
    o["markers"] = markers
    o["clipped"] = clipped
    return o
}

/**
 * Exposes the JavaScript [SVGGraphicsElement](https://developer.mozilla.org/en/docs/Web/API/SVGGraphicsElement) to Kotlin
 */
public external abstract class SVGGraphicsElement : SVGElement, SVGTests {
    open val transform: SVGAnimatedTransformList
    fun getBBox(options: SVGBoundingBoxOptions = definedExternally): DOMRect
    fun getCTM(): DOMMatrix?
    fun getScreenCTM(): DOMMatrix?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGGeometryElement](https://developer.mozilla.org/en/docs/Web/API/SVGGeometryElement) to Kotlin
 */
public external abstract class SVGGeometryElement : SVGGraphicsElement {
    open val pathLength: SVGAnimatedNumber
    fun isPointInFill(point: DOMPoint): Boolean
    fun isPointInStroke(point: DOMPoint): Boolean
    fun getTotalLength(): Float
    fun getPointAtLength(distance: Float): DOMPoint

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGNumber](https://developer.mozilla.org/en/docs/Web/API/SVGNumber) to Kotlin
 */
public external abstract class SVGNumber {
    open var value: Float
}

/**
 * Exposes the JavaScript [SVGLength](https://developer.mozilla.org/en/docs/Web/API/SVGLength) to Kotlin
 */
public external abstract class SVGLength {
    open val unitType: Short
    open var value: Float
    open var valueInSpecifiedUnits: Float
    open var valueAsString: String
    fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)
    fun convertToSpecifiedUnits(unitType: Short)

    companion object {
        val SVG_LENGTHTYPE_UNKNOWN: Short
        val SVG_LENGTHTYPE_NUMBER: Short
        val SVG_LENGTHTYPE_PERCENTAGE: Short
        val SVG_LENGTHTYPE_EMS: Short
        val SVG_LENGTHTYPE_EXS: Short
        val SVG_LENGTHTYPE_PX: Short
        val SVG_LENGTHTYPE_CM: Short
        val SVG_LENGTHTYPE_MM: Short
        val SVG_LENGTHTYPE_IN: Short
        val SVG_LENGTHTYPE_PT: Short
        val SVG_LENGTHTYPE_PC: Short
    }
}

/**
 * Exposes the JavaScript [SVGAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAngle) to Kotlin
 */
public external abstract class SVGAngle {
    open val unitType: Short
    open var value: Float
    open var valueInSpecifiedUnits: Float
    open var valueAsString: String
    fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)
    fun convertToSpecifiedUnits(unitType: Short)

    companion object {
        val SVG_ANGLETYPE_UNKNOWN: Short
        val SVG_ANGLETYPE_UNSPECIFIED: Short
        val SVG_ANGLETYPE_DEG: Short
        val SVG_ANGLETYPE_RAD: Short
    }

public external abstract class SVGNameList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: dynamic): dynamic
    fun
insertItemBefore(newItem: dynamic, index: Int): dynamic
fun replaceItem(newItem: dynamic, index: Int):
dynamic
fun appendItem(newItem: dynamic): dynamic
fun get_Item(index: Int): dynamic
fun clear(): Unit
fun initialize(newItem: SVGNumber): SVGNumber
fun insertItemBefore(newItem: SVGNumber, index: Int):
SVGNumber
fun replaceItem(newItem: SVGNumber, index: Int):
SVGNumber
fun removeItem(index: Int): SVGNumber
fun appendItem(newItem: SVGNumber): SVGNumber
fun get_Item(index: Int): SVGNumber

operator fun SVGNameList.get(index: Int): dynamic = asDynamic()[index]

operator fun SVGNameList.set(index: Int, newItem: dynamic) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGNumberList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber): SVGNumber
    fun insertItemBefore(newItem: SVGNumber, index: Int): SVGNumber
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun get_Item(index: Int): SVGNumber
}

operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript [SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGLengthList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGLength): SVGLength
    fun insertItemBefore(newItem: SVGLength, index: Int): SVGLength
    fun replaceItem(newItem: SVGLength, index: Int): SVGLength
    fun removeItem(index: Int): SVGLength
    fun appendItem(newItem: SVGLength): SVGLength
    fun get_Item(index: Int): SVGLength
}

operator fun SVGLengthList.get(index: Int): SVGLength? = asDynamic()[index]

operator fun SVGLengthList.set(index: Int, newItem: SVGLength) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript [SVGAnimatedBoolean](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedBoolean) to Kotlin
 */
public external abstract class SVGAnimatedBoolean {
    open var baseVal: Boolean
    open val animVal: Boolean
}

/**
 * Exposes the JavaScript [SVGAnimatedEnumeration](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedEnumeration) to Kotlin
 */
public external abstract class SVGAnimatedEnumeration {
    open var baseVal: Short
    open val animVal: Short
}

/**
 * Exposes the JavaScript [SVGAnimatedInteger](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedInteger) to Kotlin
 */
public external abstract class SVGAnimatedInteger {
    open var baseVal: Int
    open val animVal: Int
}

/**
 * Exposes the JavaScript [SVGAnimatedNumber](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumber) to Kotlin
 */
public external abstract class SVGAnimatedNumber {
    open var baseVal: Float
    open val animVal: Float
}

/**
 * Exposes the JavaScript [SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin
 */
public external abstract class SVGAnimatedLength {
    open val baseVal: SVGLength
    open val animVal: SVGLength
}

/**
 * Exposes the JavaScript [SVGAnimatedAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAngle) to Kotlin
 */
public external abstract class SVGAnimatedAngle {
    open val baseVal: SVGAngle
    open val animVal: SVGAngle
}

/**
 * Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect) to Kotlin
 */
public external abstract class SVGAnimatedRect {
    open val baseVal: DOMRect
    open val animVal: DOMRect
}

/**
 * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin
 */
public external abstract class SVGAnimatedNumberList {
    open val baseVal: SVGNumberList
    open val animVal: SVGNumberList
}
public abstract class SVGAnimatedLengthList {
    open val baseVal: SVGLengthList
    open val animVal: SVGLengthList
}

/**
 * Exposes the JavaScript SVGStringList to Kotlin
 */
public abstract class SVGStringList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: String): String
    fun insertItemBefore(newItem: String, index: Int): String
    fun replaceItem(newItem: String, index: Int): String
    fun appendItem(newItem: String): String
    fun getItem(index: Int): String
}

@kotlin.internal.InlineOnly
public inline operator fun SVGStringList.get(index: Int): String? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGStringList.set(index: Int, newItem: String) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript SVGUnitTypes to Kotlin
 */
public external interface SVGUnitTypes {
    companion object {
        val SVG_UNIT_TYPE_UNKNOWN: Short
        val SVG_UNIT_TYPE_USERSPACEONUSE: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
    }
}

/**
 * Exposes the JavaScript SVGZoomAndPan to Kotlin
 */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface SVGZoomAndPan {
    var zoomAndPan: Short

    companion object {
        val SVG_ZOOMANDPAN_UNKNOWN: Short
        val SVG_ZOOMANDPAN_DISABLE: Short
        val SVG_ZOOMANDPAN_MAGNIFY: Short
    }
}

/**
 * Exposes the JavaScript SVGURIReference to Kotlin
 */
public external interface SVGURIReference {
    val href: SVGAnimatedString
}

/**
 * Exposes the JavaScript SVGSVGElement to Kotlin
 */
public external abstract class SVGSVGElement : SVGGraphicsElement, SVGFitToViewBox, SVGZoomAndPan, WindowEventHandlers {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open var currentScale: Float
    open val currentTranslate: DOMPointReadOnly
    fun getIntersectionList(rect: DOMRectReadOnly, referenceElement: SVGElemen): NodeList
    fun checkIntersection(element: SVGElemen, rect: DOMRectReadOnly): Boolean
    fun checkEnclosure(element: SVGElemen, rect: DOMRectReadOnly): Boolean
    fun deselectAll()
    fun createSVGNumber(): SVGNumber
    fun createSVGLength(): SVGLength
    fun createSVGAngle(): SVGAngle
    fun createSVGPoint(): DOMPoint
    fun createSVGMatrix(): DOMMatrix
    fun createSVGRect(): DOMRect
    fun createSVGTransform(): SVGTransform
    fun createSVGTransformFromMatrix(matrix: DOMMatrix): SVGTransform
    fun get责任ByElementId(elementId: String): Element
    fun suspendRedraw(maxWaitMilliseconds: Int): Int
    fun unsuspendRedraw(suspendHandleID: Int): Boolean
    fun forceRedraw()

    companion object {
        val SVG_ZOOMANDPAN_UNKNOWN: Short
        val SVG_ZOOMANDPAN_DISABLE: Short
        val SVG_ZOOMANDPAN_MAGNIFY: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
    }
}
/**
 * Exposes the JavaScript [SVGTitleElement](https://developer.mozilla.org/en/docs/Web/API/SVGTitleElement) to Kotlin
 */
public external abstract class SVGTitleElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGSymbolElement](https://developer.mozilla.org/en/docs/Web/API/SVGSymbolElement) to Kotlin
 */
public external abstract class SVGSymbolElement : SVGGraphicsElement, SVGFitToViewBox {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGUseElement](https://developer.mozilla.org/en/docs/Web/API/SVGUseElement) to Kotlin
 */
public external abstract class SVGUseElement : SVGGraphicsElement, SVGURIReference {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open val instanceRoot: SVGElement?
    open val animatedInstanceRoot: SVGElement?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external open class SVGUseElementShadowRoot : ShadowRoot {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE Node: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

public external interface SVGElementInstance {
    val correspondingElement: SVGElement?
        get() = definedExternally
    val correspondingUseElement: SVGUseElement?
        get() = definedExternally

public external open class ShadowAnimation(source: dynamic, newTarget: dynamic) {
    open val sourceAnimation: dynamic
}

/**
 * Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement)
to Kotlin
 */
public external abstract class SVGSwitchElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface GetSVGDocument {
    fun getSVGDocument(): Document
}

/**
 * Exposes the JavaScript [SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin
 */
public external abstract class SVGStyleElement : SVGElement, LinkStyle {
    open var type: String
    open var media: String
    open var title: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin
 */
public external abstract class SVGTransform {
    open val type: Short
    open val matrix: DOMMatrix
    open val angle: Float
    fun setMatrix(matrix: DOMMatrixReadOnly)
    fun setTranslate(tx: Float, ty: Float)
    fun setScale(sx: Float, sy: Float)
    fun setRotate(angle: Float, cx: Float, cy: Float)
    fun setSkewX(angle: Float)
    fun setSkewY(angle: Float)

    companion object {
        val SVG_TRANSFORM_UNKNOWN: Short
        val SVG_TRANSFORM_MATRIX: Short
        val SVG_TRANSFORM_TRANSLATE: Short
        val SVG_TRANSFORM_SCALE: Short
        val SVG_TRANSFORM_ROTATE: Short
        val SVG_TRANSFORM_SKEWX: Short
        val SVG_TRANSFORM_SKEWY: Short
    }
}

/**
 * Exposes the JavaScript [SVGTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGTransformList) to Kotlin
 */
public external abstract class SVGTransformList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGTransform): SVGTransform
    fun insertItemBefore(newItem: SVGTransform, index: Int): SVGTransform
    fun replaceItem(newItem: SVGTransform, index: Int): SVGTransform
    fun removeItem(index: Int): SVGTransform
    fun appendItem(newItem: SVGTransform):
fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform

fun getItem(index: Int): SVGTransform?

fun consolidate(): SVGTransform?

fun getItem(index: Int): SVGTransform

@kotlin.internal.InlineOnly
public inline operator fun SVGTransformList.get(index: Int): SVGTransform? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGTransformList.set(index: Int, newItem: SVGTransform) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript [SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to Kotlin
 */
public external abstract class SVGAnimatedTransformList {
    open val baseVal: SVGTransformList
    open val animVal: SVGTransformList
}

/**
 * Exposes the JavaScript [SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin
 */
public external abstract class SVGPreserveAspectRatio {
    open var align: Short
    open var meetOrSlice: Short

    companion object {
        val SVG_PRESERVEASPECTRATIO_UNKNOWN: Short
        val SVG_PRESERVEASPECTRATIO_NONE: Short
        val SVG_PRESERVEASPECTRATIO_XMINYMIN: Short
        val SVG_PRESERVEASPECTRATIO_XMINYMID: Short
        val SVG_PRESERVEASPECTRATIO_XMINYMAX: Short
        val SVG_PRESERVEASPECTRATIO_XMIDYMIN: Short
        val SVG_PRESERVEASPECTRATIO_XMIDYMID: Short
        val SVG_PRESERVEASPECTRATIO_XMIDYMAX: Short
        val SVG_PRESERVEASPECTRATIO_XMAXYMIN: Short
        val SVG_PRESERVEASPECTRATIO_XMAXYMID: Short
        val SVG_PRESERVEASPECTRATIO_XMAXYMAX: Short
        val SVG_MEETORSLICE_UNKNOWN: Short
        val SVG_MEETORSLICE_MEET: Short
        val SVG_MEETORSLICE_SLICE: Short
    }
}

/**
 * Exposes the JavaScript [SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspectRatio) to Kotlin
 */
public external abstract class SVGAnimatedPreserveAspectRatio {
    open val baseVal: SVGPreserveAspectRatio
    open val animVal: SVGPreserveAspectRatio
}

/**
 * Exposes the JavaScript [SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin
 */
public external abstract class SVGPathElement : SVGGeometryElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGRectElement](https://developer.mozilla.org/en/docs/Web/API/SVGRectElement) to Kotlin
 */
public external abstract class SVGRectElement : SVGGeometryElement {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open val rx: SVGAnimatedLength
    open val ry: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript
[SVGCircleElement](https://developer.mozilla.org/en/docs/Web/API/SVGCircleElement) to Kotlin */
public external abstract class SVGCircleElement : SVGGeometryElement {
    open val cx: SVGAnimatedLength
    open val cy: SVGAnimatedLength
    open val r: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
[SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin */
public external abstract class SVGEllipseElement : SVGGeometryElement {
    open val cx: SVGAnimatedLength
    open val cy: SVGAnimatedLength
    open val rx: SVGAnimatedLength
    open val ry: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
[SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin */
public external abstract class SVGLineElement : SVGGeometryElement {
    open val x1: SVGAnimatedLength
    open val y1: SVGAnimatedLength
    open val x2: SVGAnimatedLength
    open val y2: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
[SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin */
public external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference {

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/*\n*/

/**
 * Exposes the JavaScript
[SvgPolylineElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolylineElement) to Kotlin
*/
public external abstract class SVGPolylineElement : SVGGeometryElement, SVGAnimatedPoints {

companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
[SvgPolygonElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolygonElement) to Kotlin
*/
public external abstract class SVGPolygonElement : SVGGeometryElement, SVGAnimatedPoints {

companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
[SvgTextContentElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextContentElement) to Kotlin
*/
public external abstract class SVGTextContentElement : SVGGraphicsElement {
    open val textLength: SVGAnimatedLength
    open val lengthAdjust: SVGAnimatedEnumeration
    fun getNumberOfChars(): Int
    fun getComputedTextLength(): Float
    fun getSubStringLength(charnum: Int, nchars: Int): Float
    fun getStartPositionOfChar(charnum: Int): DOMPoint
    fun getEndPositionOfChar(charnum: Int): DOMPoint
    fun getExtentOfChar(charnum: Int): DOMRect
    fun getRotationOfChar(charnum: Int): Float
    fun getCharNumAtPosition(point: DOMPoint): Int
    fun selectSubString(charnum: Int, nchars: Int)

    companion object {
        val LENGTHADJUST_UNKNOWN: Short
        val LENGTHADJUST_SPACING: Short
        val LENGTHADJUST_SPACINGANDGLYPHS: Short
        val ELEMENT_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
Short
val TEXTPATH_METHODTYPE_ALIGN: Short
val TEXTPATH_METHODTYPE_STRETCH: Short
val TEXTPATH_SPACINGTYPE_AUTO: Short
val TEXTPATH_SPACINGTYPE_EXACT: Short
val LENGTHADJUST_UNKNOWN: Short
val LENGTHADJUST_SPACING: Short
val LENGTHADJUST_SPACINGANDGLYPHS: Short
val ELEMENT_NODE: Short
val ATTRIBUTEE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_NODE: Short
val ENTITY_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [SVGImageElement](https://developer.mozilla.org/en/docs/Web/API/SVGImageElement) to Kotlin
 */
public external abstract class SVGImageElement : SVGGraphicsElement, SVGURIReference,
HTMLSVGElement {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open val preserveAspectRatio: SVGAnimatedPreserveAspectRatio
    open var crossOrigin: String?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGForeignObjectElement](https://developer.mozilla.org/en/docs/Web/API/SVGForeignObjectElement) to Kotlin
 */
public external abstract class SVGForeignObjectElement : SVGGraphicsElement {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGMarkerElement : SVGElement, SVGFitToViewBox {
    open val refX: SVGAnimatedLength
    open val refY: SVGAnimatedLength
    open val markerUnits: SVGAnimatedEnumeration
    open val markerWidth: SVGAnimatedLength
    open val markerHeight: SVGAnimatedLength
    open var orientType: SVGAnimatedEnumeration
    open var orientAngle: SVGAngle
    fun setOrientToAuto()
    fun setOrientToAngle(angle: SVGAngle)

    companion object {
        val SVG_MARKERUNITS_UNKNOWN: Short
        val SVG_MARKERUNITS_USERSPACEONUSE: Short
        val SVG_MARKERUNITS_STROKEDOWNANDGLYPHS: Short
        val SVG_MARKERUNITS_STROKEDOWNANDGLYPHS: Short
        val SVG_MARKERUNITS_STROKEWIDTH: Short
        val SVG_MARKERUNITS_UNKNOWN: Short
        val SVG_MARKERUNITS_USERSPACEONUSE: Short
        val SVG_MARKERUNITS_STROKEDOWNANDGLYPHS: Short
        val SVG_MARKERUNITS_STROKEWIDTH: Short
    }

val SVG_MARKER_Orient_AUTO: Short
val SVG_MARKER_Orient_ANGLE: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short

val SVG_MARKER_Orient_AUTO: Short
val SVG_MARKER_Orient_ANGLE: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short

/**
 * Exposes the JavaScript

[SVGGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGGradientElement) to Kotlin
*/
public external abstract class SVGGradientElement : SVGElement, SVGURIReference, SVGUnitTypes {
open val gradientUnits: SVGAnimatedEnumeration
open val gradientTransform: SVGAnimatedTransformList
open val spreadMethod: SVGAnimatedEnumeration

    companion object {
        val SVG_SPREADMETHOD_UNKNOWN: Short
        val SVG_SPREADMETHOD_PAD: Short
        val SVG_SPREADMETHOD_REFLECT: Short
        val SVG_SPREADMETHOD_REPEAT: Short
        val SVG_UNIT_TYPE_UNKNOWN: Short
        val SVG_UNIT_TYPE_USERSPACEONUSE: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }

/**
 * Exposes the JavaScript

[SVGLinearGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGLinearGradientElement) to Kotlin
*/
public external abstract class SVGLinearGradientElement : SVGGradientElement {
open val x1: SVGAnimatedLength
open val y1: SVGAnimatedLength
open val x2: SVGAnimatedLength
open val y2: SVGAnimatedLength

    companion object {
        val SVG_SPREADMETHOD_UNKNOWN: Short
        val SVG_SPREADMETHOD_PAD: Short
        val SVG_SPREADMETHOD_REFLECT: Short
        val SVG_SPREADMETHOD_REPEAT: Short
        val SVG_UNIT_TYPE_UNKNOWN: Short
        val SVG_UNIT_TYPE_USERSPACEONUSE: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }

/**
 * Exposes the JavaScript

[SVGSolidcolorElement](https://developer.mozilla.org/en/docs/Web/API/SVGSolidcolorElement) to Kotlin
*/
public external abstract class SVGSolidcolorElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}
/**
 * Exposes the JavaScript [SVGRadialGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGRadialGradientElement) to Kotlin
 * public external abstract class SVGRadialGradientElement : SVGradientElement {
 *   open val cx: SVGAnimatedLength
 *   open val cy: SVGAnimatedLength
 *   open val r: SVGAnimatedLength
 *   open val fx: SVGAnimatedLength
 *   open val fy: SVGAnimatedLength
 *   companion object {
 *     val SVG_SPREADMETHOD_UNKNOWN: Short
 *     val SVG_SPREADMETHOD_PAD: Short
 *     val SVG_SPREADMETHOD_REFLECT: Short
 *     val SVG_SPREADMETHOD_REPEAT: Short
 *     val SVG_UNIT_TYPE_UNKNOWN: Short
 *     val SVG_UNIT_TYPE_USERSPACEONUSE: Short
 *     val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
 *     val ELEMENT_NODE: Short
 *     val ATTRIBUTE_NODE: Short
 *     val TEXT_NODE: Short
 *     val CDATA_SECTION_NODE: Short
 *     val ENTITY_REFERENCE_NODE: Short
 *     val ENTITY_NODE: Short
 *     val PROCESSING_INSTRUCTION_NODE: Short
 *     val COMMENT_NODE: Short
 *     val DOCUMENT_NODE: Short
 *     val DOCUMENT_TYPE_NODE: Short
 *     val DOCUMENT_FRAGMENT_NODE: Short
 *     val NOTATION_NODE: Short
 *     val DOCUMENT_POSITION_DISCONNECTED: Short
 *     val DOCUMENT_POSITION_PRECEDING: Short
 *     val DOCUMENT_POSITION_FOLLOWING: Short
 *     val DOCUMENT_POSITION_CONTAINS: Short
 *     val DOCUMENT_POSITION_CONTAINED_BY: Short
 *     val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 *   }
 * }
 * public external abstract class SVGMeshGradientElement : SVGGradientElement {
 *   companion object {
 *     val SVG_SPREADMETHOD_UNKNOWN: Short
 *     val SVG_SPREADMETHOD_PAD: Short
 *     val SVG_SPREADMETHOD_REFLECT: Short
 *     val SVG_SPREADMETHOD_REPEAT: Short
 *     val SVG_UNIT_TYPE_UNKNOWN: Short
 *     val SVG_UNIT_TYPE_USERSPACEONUSE: Short
 *     val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
 *     val ELEMENT_NODE: Short
 *     val ATTRIBUTE_NODE: Short
 *     val TEXT_NODE: Short
 *     val CDATA_SECTION_NODE: Short
 *     val ENTITY_REFERENCE_NODE: Short
 *     val ENTITY_NODE: Short
 *     val PROCESSING_INSTRUCTION_NODE: Short
 *     val COMMENT_NODE: Short
 *     val DOCUMENT_NODE: Short
 *     val DOCUMENT_TYPE_NODE: Short
 *     val DOCUMENT_FRAGMENT_NODE: Short
 *     val NOTATION_NODE: Short
 *     val DOCUMENT_POSITION_DISCONNECTED: Short
 *     val DOCUMENT_POSITION_PRECEDING: Short
 *     val DOCUMENT_POSITION_FOLLOWING: Short
 *     val DOCUMENT_POSITION_CONTAINS: Short
 *     val DOCUMENT_POSITION_CONTAINED_BY: Short
 *     val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 *   }
 * }
 * public external abstract class SVGMeshrowElement : SVGElement {
 *   companion object {
 *     val ELEMENT_NODE: Short
 *     val ATTRIBUTE_NODE: Short
 *     val TEXT_NODE: Short
 *     val CDATA_SECTION_NODE: Short
 *     val ENTITY_REFERENCE_NODE: Short
 *     val ENTITY_NODE: Short
 *     val PROCESSING_INSTRUCTION_NODE: Short
 *     val COMMENT_NODE: Short
 *     val DOCUMENT_NODE: Short
 *     val DOCUMENT_TYPE_NODE: Short
 *     val DOCUMENT_FRAGMENT_NODE: Short
 *     val NOTATION_NODE: Short
 *     val DOCUMENT_POSITION_DISCONNECTED: Short
 *     val DOCUMENT_POSITION_PRECEDING: Short
 *     val DOCUMENT_POSITION_FOLLOWING: Short
 *     val DOCUMENT_POSITION_CONTAINS: Short
 *     val DOCUMENT_POSITION_CONTAINED_BY: Short
 *     val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 *   }
 * }
 * public external abstract class SVGMeshpatchElement : SVGElement {
 *   companion object {
 *     val ELEMENT_NODE: Short
 *     val ATTRIBUTE_NODE: Short
 *     val TEXT_NODE: Short
 *     val CDATA_SECTION_NODE: Short
 *     val ENTITY_REFERENCE_NODE: Short
 *     val ENTITY_NODE: Short
 *     val PROCESSING_INSTRUCTION_NODE: Short
 *     val COMMENT_NODE: Short
 *     val DOCUMENT_NODE: Short
 *     val DOCUMENT_TYPE_NODE: Short
 *     val DOCUMENT_FRAGMENT_NODE: Short
 *     val NOTATION_NODE: Short
 *     val DOCUMENT_POSITION_DISCONNECTED: Short
 *     val DOCUMENT_POSITION_PRECEDING: Short
 *     val DOCUMENT_POSITION_FOLLOWING: Short
 *     val DOCUMENT_POSITION_CONTAINS: Short
 *     val DOCUMENT_POSITION_CONTAINED_BY: Short
 *     val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
 *   }
 * */

/*
 * Copyright 2010-2019
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!// See github.com/kotlin/dukat for details

package org.w3c.files

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript
 [Blob](https://developer.mozilla.org/en/docs/Web/API/Blob) to Kotlin
 */
public external open class Blob(blobParts: Array<dynamic> = definedExternally, options: BlobPropertyBag = definedExternally) :
 ImageBitmapSource {
 open val size: Number
 open val type: String
 open val isClosed: Boolean
 fun slice(start: Int = definedExternally, end: Int = definedExternally, contentType: String = definedExternally): Blob

 fun close()
}

public external interface BlobPropertyBag {
 var type: String? /* = "" */
 get() = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun BlobPropertyBag(type: String? = ": BlobPropertyBag {

 /**
 * Exposes the JavaScript
 [File](https://developer.mozilla.org/en/docs/Web/API/File) to Kotlin
 */
 public external open class File(fileBits: Array<dynamic>, fileName: String, options: FilePropertyBag = definedExternally) : Blob {
 open val name: String
 open val lastModified: Int
}

 public external interface FilePropertyBag : BlobPropertyBag {
 var lastModified: Int?
 get() = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun FilePropertyBag(lastModified: Int? = undefined, type: String? = ": FilePropertyBag {

 /**
 * Exposes the JavaScript
 [FileList](https://developer.mozilla.org/en/docs/Web/API/FileList) to Kotlin
 */
 public external abstract class FileList : ItemArrayLike<File> {
 override fun item(index: Int): File? = asDynamic()
}

 /**
 * Exposes the JavaScript
 [FileReader](https://developer.mozilla.org/en/docs/Web/API/FileReader) to Kotlin
 */
 public external open class FileReader : EventTarget {
 open val readyState: String
 open val readyStateText: String
 open val result: String
 open val error: String
 var onloadstart: ((ProgressEvent) -> dynamic)
 var onprogress: ((ProgressEvent) -> dynamic)
 var onabort: ((Event) -> dynamic)
 var onerror: ((Event) -> dynamic)
 var onloadend: ((Event) -> dynamic)
 fun readAsArrayBuffer(blob: Blob): ArrayBuffer
 fun readAsBinaryString(blob: Blob): String
 fun readAsText(blob: Blob, label: String = definedExternally): String
 fun readAsDataURL(blob: Blob): String
}

 /**
 * Exposes the JavaScript
 [FileReaderSync](https://developer.mozilla.org/en/docs/Web/API/FileReaderSync) to Kotlin
 */
 public external open class FileReaderSync {
 fun readAsArrayBuffer(blob: Blob): ArrayBuffer
 fun readAsBinaryString(blob: Blob): String
 fun readAsText(blob: Blob, label: String = definedExternally): String
 fun readAsDataURL(blob: Blob): String
}
Language contributors. * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. ** NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT**

```
// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.notifications

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.*
import org.w3c.dom.*
import org.w3c.dom.svg.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.timer.*
import org.w3c.dom.url.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript [Notification](https://developer.mozilla.org/en/docs/Web/API/Notification) to Kotlin
 */
public external open class Notification(title: String, options: NotificationOptions = definedExternally) : EventTarget {
    var onclick: ((MouseEvent) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    open val title: String
    open val dir: NotificationDirection
    open val lang: String
    open val body: String
    open val tag: String
    open val image: String
    open val icon: String
    open val badge: String
    open val sound: String
    open val vibrate: Array<out Int>
    open val timestamp: Number
    open val renotify: Boolean
    open val silent: Boolean
    open val noscreen: Boolean
    open val requireInteraction: Boolean
    open val sticky: Boolean
    open val data: Any?
    open val actions: Array<out NotificationAction>
    fun close()

    companion object {
        val permission: NotificationPermission
        val maxActions: Int
        fun requestPermission(deprecatedCallback: (NotificationPermission) -> Unit = definedExternally): Promise<NotificationPermission>
    }
}

public external interface NotificationOptions {
    var dir: NotificationDirection? /* = NotificationDirection.AUTO */
    var lang: String? /* = "" */
    var body: String? /* = "" */
    var tag: String? /* = "" */
    var image: String?
    var icon: String?
    var badge: String?
    var sound: String?
    var vibrate: dynamic
    var timestamp: Number?
    var renotify: Boolean? /* = false */
    var silent: Boolean? /* = false */
    var noscreen: Boolean? /* = false */
    var requireInteraction: Boolean? /* = false */
    var sticky: Boolean? /* = false */
    var data: Any? /* = null */
    var actions: Array<NotificationAction>? /* = arrayOf() */
}

@kotlin.internal.InlineOnly
```

Open Source Used in DNAC Platform Antelope  4144
    val o = js("({})")
    o["action"] = action
    o["title"] = title
    o["icon"] = icon
    return o
}

public external interface GetNotificationOptions {
    var tag: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun GetNotificationOptions(tag: String? = ""): GetNotificationOptions {
    val o = js("({})")
    o["tag"] = tag
    return o
}

/**
 * Exposes the JavaScript [NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin
 */
public external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) : ExtendableEvent {
    open val notification: Notification
    open val action: String

    companion object {

        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface NotificationEventInit : ExtendableEventInit {
    var notification: Notification?
        get() = definedExternally
        set(value) = definedExternally
    var action: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["notification"] = notification
    o["action"] = action
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NotificationPermission {
    companion object
}

public inline val NotificationPermission.Companion.DEFAULT: NotificationPermission get() = "default".asDynamic().unsafeCast<NotificationPermission>()

public inline val NotificationPermission.Companion.DENIED: NotificationPermission get() = "denied".asDynamic().unsafeCast<NotificationPermission>()

public inline val NotificationPermission.Companion.GRANTED: NotificationPermission get() = "granted".asDynamic().unsafeCast<NotificationPermission>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NotificationDirection {
    companion object
}

public inline val NotificationDirection.Companion.AUTO: NotificationDirection get() = "auto".asDynamic().unsafeCast<NotificationDirection>()

public inline val NotificationDirection.Companion.LTR: NotificationDirection get() = "ltr".asDynamic().unsafeCast<NotificationDirection>()

public inline val NotificationDirection.Companion.RTL: NotificationDirection get() = "rtl".asDynamic().unsafeCast<NotificationDirection>()
String
    open var onupdatefound: ((Event) -> dynamic)?
    open val APISpace: dynamic
    fun update():
    fun unregister(): Promise<Boolean>
    fun showNotification(title: String, options: NotificationOptions = definedExternally): Promise<ServiceWorkerRegistration>
    fun getRegistrations(): Promise<Array<ServiceWorkerRegistration>>
    fun methodName(): Promise<dynamic>}

/**
 * Exposes the JavaScript
[ServiceWorkerContainer](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerContainer) to Kotlin
 */
public external abstract class ServiceWorkerContainer : EventTarget {
    open val controller:
        ServiceWorker?
    open val ready: Promise<ServiceWorkerRegistration>
    open var oncontrollerchange:
        ((Event) -> dynamic)?
    fun register(scriptURL: String, options: RegistrationOptions = definedExternally): Promise<ServiceWorkerRegistration>
    fun getRegistration(clientURL: String = definedExternally): PromiseAny?
    fun startMessages():
    fun sendMessages():
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}

public external interface RegistrationOptions {
    var scope: String?
        get() = definedExternally
        set(value) = definedExternally
    var type: WorkerType? /* = WorkerType.CLASSIC */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun RegistrationOptions(scope: String? = undefined, type: WorkerType? = WorkerType.CLASSIC): RegistrationOptions {
    val o = js("({})")
    o["scope"] = scope
    o["type"] = type
    return o
}

/**
 * Exposes the JavaScript
[ServiceWorkerMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerMessageEvent) to Kotlin
 */
public external open class ServiceWorkerMessageEvent(type: String, eventInitDict: ServiceWorkerMessageEventInit = definedExternally) : Event {
    open val data: Any
    open val origin:
    open val lastEventId: String
    open val source:
    open val ports: Array<MessagePort>?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface ServiceWorkerMessageEventInit : EventInit {
    var data: Any?
        get() = definedExternally
        set(value) = definedExternally
    var origin: String?
        get() = definedExternally
        set(value) = definedExternally
    var lastEventId: String?
        get() = definedExternally
        set(value) = definedExternally
    var source: UnionMessagePortOrServiceWorker?
        get() = definedExternally
        set(value) = definedExternally
    var ports: Array<MessagePort>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["data"] = data
    o["origin"] = origin
    o["lastEventId"] = lastEventId
    o["source"] = source
    o["ports"] = ports
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
[ServiceWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerGlobalScope) to Kotlin
 */
public external abstract class ServiceWorkerGlobalScope : WorkerGlobalScope {
    open val clients:
        Clients
    open val registration: ServiceWorkerRegistration
    open var oninstall: ((Event) -> dynamic)?
    open var onactivate: ((Event) -> dynamic)?
    open var onfetch: ((FetchEvent) -> dynamic)?
    open var onforeignfetch: ((Event) -> dynamic)?
    open var onmessage: ((MessageEvent) -> dynamic)?
    open var onfunctionalevent: ((Event) -> dynamic)?
    fun skipWaiting():
    fun bridgeToWorker():
}

/**
 * Exposes the JavaScript
[Client](https://developer.mozilla.org/en/docs/Web/API/Client) to Kotlin
 */
public external abstract class Client :
    UnionClientOrMessagePortOrServiceWorker {
    open val url: String
    open val frameType: FrameType
    open val id: String
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}
ForeignFetchEventInit { val o = js("({})")\n o["request"] = request\n o["origin"] = origin\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return \n\n}"\n\n\n\n
ForeignFetchResponse {\n var response: Response?\n get() = definedExternally\n set(value) = definedExternally\n var origin: String?\n get() = definedExternally\n set(value) = definedExternally\n var headers: Array<String>?\n get() = definedExternally\n set(value) = definedExternally\n\n return\n\n}"\n\n\n
ForeignFetchEventInit {
    val o = js("({})")
    o["request"] = request
    o["origin"] = origin
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

public external interface ForeignFetchResponse {
    var response: Response?
        get() = definedExternally
        set(value) = definedExternally
    var origin: String?
        get() = definedExternally
        set(value) = definedExternally
    var headers: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")\n    o["response"] = response\n    o["origin"] = origin\n    o["headers"] = headers\n    return o
}

/**
 * Exposes the JavaScript
 [ExtendableMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableMessageEvent) to Kotlin
 */
public external open class ExtendableMessageEvent(type: String, eventInitDict: ExtendableMessageEventInit = definedExternally) : ExtendableEvent {
    open val data: Any?
    open val origin: String
    open val lastEventId: String
    open val source: UnionClientOrMessagePortOrServiceWorker?
    open val ports: Array<out MessagePort>?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface ExtendableMessageEventInit : ExtendableEventInit {
    var data: Any?
        get() = definedExternally
        set(value) = definedExternally
    var origin: String?
        get() = definedExternally
        set(value) = definedExternally
    var lastEventId: String?
        get() = definedExternally
        set(value) = definedExternally
    var source: UnionClientOrMessagePortOrServiceWorker?
        get() = definedExternally
        set(value) = definedExternally
    var ports: Array<MessagePort>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["data"] = data\n    o["origin"] = origin\n    o["lastEventId"] = lastEventId\n    o["source"] = source\n    o["ports"] = ports\n    o["bubbles"] = bubbles\n    o["cancelable"] = cancelable\n    o["composed"] = composed\n    return o
}

/**
 * Exposes the JavaScript
 [Cache](https://developer.mozilla.org/en/docs/Web/API/Cache) to Kotlin
 */
public external abstract class Cache {
    fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?>
    fun matchAll(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally): Promise<Array<Response>>
    fun add(request: dynamic): Promise<Unit>
    fun addAll(requests: Array<dynamic>): Promise<Unit>
    fun put(request: dynamic, response: Response): Promise<Unit>
    fun delete(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Boolean>
    fun keys(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally): Promise<Array<Request>>
}

public external interface CacheQueryOptions {
    var ignoreSearch: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var ignoreMethod: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var ignoreVary: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var cacheName: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["ignoreSearch"] = ignoreSearch\n    o["ignoreMethod"] = ignoreMethod\n    o["ignoreVary"] = ignoreVary\n    o["cacheName"] = cacheName
    return o
}
undefined): CacheBatchOperation {
    private final val o = js("{}")
    o["type"] = type
    o["request"] = request
    o["response"] = response
    o["options"] = options
    return o
} * Exposes the JavaScript
[CacheStorage](https://developer.mozilla.org/en/docs/Web/API/CacheStorage) to Kotlin

abstract class CacheStorage {
    fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?>
    fun has(cacheName: String): Promise<Cache>
    fun open(cacheName: String): Promise<Cache>
    fun delete(cacheName: String): Promise<Boolean>
    fun keys(): Promise<Array<String>>
}

abstract class FunctionalEvent : ExtendableEvent {
    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

interface UnionMessagePortOrServiceWorker

interface UnionClientOrMessagePortOrServiceWorker

interface ServiceWorkerState {
    companion object
}

ServiceWorkerState.Companion.INSTALLING: ServiceWorkerState get() = "installing".asDynamic().unsafeCast<ServiceWorkerState>()
ServiceWorkerState.Companion.INSTALLED: ServiceWorkerState get() = "installed".asDynamic().unsafeCast<ServiceWorkerState>()
ServiceWorkerState.Companion.ACTIVATING: ServiceWorkerState get() = "activating".asDynamic().unsafeCast<ServiceWorkerState>()
ServiceWorkerState.Companion.ACTIVATED: ServiceWorkerState get() = "activated".asDynamic().unsafeCast<ServiceWorkerState>()
ServiceWorkerState.Companion.REDUNDANT: ServiceWorkerState get() = "redundant".asDynamic().unsafeCast<ServiceWorkerState>()

interface FrameType {
    companion object
}

FrameType.Companion.AUXILIARY: FrameType get() = "auxiliary".asDynamic().unsafeCast<FrameType>()
FrameType.Companion.TOP_LEVEL: FrameType get() = "top-level".asDynamic().unsafeCast<FrameType>()
FrameType.Companion.NESTED: FrameType get() = "nested".asDynamic().unsafeCast<FrameType>()
FrameType.Companion.NONE: FrameType get() = "none".asDynamic().unsafeCast<FrameType>()

interface ClientType {
    companion object
}

ClientType.Companion.WINDOW: ClientType get() = "window".asDynamic().unsafeCast<ClientType>()
ClientType.Companion.WORKER: ClientType get() = "worker".asDynamic().unsafeCast<ClientType>()
ClientType.Companion.SHAREDWORKER: ClientType get() = "sharedworker".asDynamic().unsafeCast<ClientType>()
ClientType.Companion.ALL: ClientType get() = "all".asDynamic().unsafeCast<ClientType>()

Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.

NOT EDIT!

See github.com/kotlin/dukat for details

Open Source Used In DNAC Platform Antelope 4149
open var ontimeout: ((Event) -> dynamic)?
    open var onloadend: ((Event) -> dynamic)?
}

public external
abstract class XMLHttpRequestUpload : XMLHttpRequestEventTarget

/**
 * Exposes the JavaScript [XMLHttpRequest](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequest) to Kotlin
 */
public external open class XMLHttpRequest : XMLHttpRequestEventTarget {
    var onreadystatechange: ((Event) -> dynamic)?
    open val readyState: Short
    var timeout: Int
    var withCredentials: Boolean
    open val upload: XMLHttpRequestUpload
    open val responseURL: String
    open val status: Short
    open val statusText:
    String
    var responseType: XMLHttpRequestResponseType
    open val response: Any?
    open val responseText: String
    open val responseXML: Document?
    fun open(method: String, url: String)
    fun open(method: String, url: String, async: Boolean, username: String? = definedExternally, password: String? =
    definedExternally)
    fun setRequestHeader(name: String, value: String)
    fun send(body: dynamic =
    definedExternally)
    fun abort()
    fun getResponseHeader(name: String): String?
    fun getAllResponseHeaders(): String
    fun overrideMimeType(mime: String)

    companion object {
        val UNSENT: Short
        val OPENED: Short
        val HEADERS_RECEIVED: Short
        val LOADING: Short
        val DONE: Short
    }
}

/**
 * Exposes the JavaScript [FormData](https://developer.mozilla.org/en/docs/Web/API/FormData) to Kotlin
 */
public external open class FormData(form: HTMLFormElement = definedExternally) {
    fun append(name: String, value: String)
    fun append(name: String, value: Blob, filename: String = definedExternally)
    fun delete(name: String)
    fun get(name: String): dynamic
    fun getAll(name: String): Array<dynamic>
    fun has(name: String): Boolean
    fun set(name: String, value: String)
    fun set(name: String, value: Blob, filename: String = definedExternally)
}

/**
 * Exposes the JavaScript [ProgressEvent](https://developer.mozilla.org/en/docs/Web/API/ProgressEvent) to Kotlin
 */
public external open class ProgressEvent(type: String, eventInitDict: ProgressEventInit = definedExternally) : Event {
    open val lengthComputable: Boolean
    open val loaded: Number
    open val total: Number

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface ProgressEventInit : EventInit {
    var lengthComputable: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var loaded: Number? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var total: Number? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ProgressEventInit(lengthComputable:
    Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally)
    var loaded: Number? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var total: Number? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var bubbles: Boolean
        get() = definedExternally
        set(value) = definedExternally
    var cancelable: Boolean
        get() = definedExternally
        set(value) = definedExternally
    var composed: Boolean
        get() = definedExternally
        set(value) = definedExternally
    var returnOnUnderlyingComposable: Boolean
        get() = definedExternally
        set(value) = definedExternally
}

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface XMLHttpRequestResponseType {
    companion object
}

public inline val XMLHttpRequestResponseType.Companion.EMPTY: XMLHttpRequestResponseType get() =
"\".asDynamic().unsafeCast<XMLHttpRequestResponseType>().unsafeCast
XMLHttpRequestResponseType.Companion.EMPTY
XMLHttpRequestResponseType.Companion.ARRAYBUFFER: XMLHttpRequestResponseType get() =
"arraybuffer".asDynamic().unsafeCast<XMLHttpRequestResponseType>().unsafeCast
XMLHttpRequestResponseType.Companion.ARRAYBUFFER
XMLHttpRequestResponseType.Companion.BLOB: XMLHttpRequestResponseType get() =
"blob".asDynamic().unsafeCast<XMLHttpRequestResponseType>().unsafeCast
XMLHttpRequestResponseType.Companion.BLOB
XMLHttpRequestResponseType.Companion.DOCUMENT: XMLHttpRequestResponseType get() =
"document".asDynamic().unsafeCast<XMLHttpRequestResponseType>().unsafeCast
XMLHttpRequestResponseType.Companion.DOCUMENT
XMLHttpRequestResponseType.Companion.JSON: XMLHttpRequestResponseType get() =
"json".asDynamic().unsafeCast<XMLHttpRequestResponseType>().unsafeCast
XMLHttpRequestResponseType.Companion.JSON
XMLHttpRequestResponseType.Companion.TEXT: XMLHttpRequestResponseType get() =
"text".asDynamic().unsafeCast<XMLHttpRequestResponseType>().unsafeCast
XMLHttpRequestResponseType.Companion.TEXT

Open Source Used In DNAC Platform Antelope 4150

* Copyright 2010-2018 JetBrains s.r.o.
* and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.

```kotlin
/**
 * Signals that the annotated annotation class is a marker of an experimental API.
 * Any declaration annotated with that marker is considered an experimental declaration
 * and its call sites should accept the experimental aspect of it either by using [UseExperimental], or by being annotated with that marker themselves, effectively causing further propagation of that experimental aspect.
 * This class is experimental itself and can only be used with the compiler argument `-Xuse-experimental=kotlin.Experimental`.
 */
@Target(ANNOTATION_CLASS)
@Retention(BINARY)
@SinceKotlin("1.2")
@RequireKotlin("1.2.50 ", versionKind = RequireKotlinVersionKind.COMPILE_VERSION)
@Suppress("ANNOTATION_CLASS_MEMBER")
public annotation class Experimental(val level: Level = Level.ERROR) {
    /**
     * Severity of the diagnostic that should be reported on usages of experimental API which did not explicitly accept the experimental aspect.
     */
    public enum class Level {
        /** Specifies that a warning should be reported on incorrect usages of this experimental API. */
        WARNING,
        /** Specifies that an error should be reported on incorrect usages of this experimental API. */
        ERROR,
    }
}

/**
 * Allows to use experimental API denoted by the given markers in the annotated file, declaration, or expression.
 * If a declaration is annotated with [UseExperimental], its usages are **not** required to opt-in to that experimental API.
 *
 * This class is experimental itself and can only be used with the compiler argument `-Xuse-experimental=kotlin.Experimental`.
 */
@Target(
    CLASS,
    PROPERTY, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION,
    PROPERTY_GETTER, PROPERTY_SETTER, EXPRESSION, FILE,
    TYPEALIAS)
@Retention(SOURCE)
@SinceKotlin("1.2")
@RequireKotlin("1.2.30", versionKind = RequireKotlinVersionKind.COMPILE_VERSION)
@MustBeDocumented
public annotation class UseExperimental(
    vararg val markerClass: KClass<out Annotation>
)

@Target(CLASS, PROPERTY, CONSTRUCTOR, FUNCTION,
    TYPEALIAS)
@Retention(BINARY)
@MustBeDocumented
@ExperimentalStdlibApi
@SinceKotlin("1.3")  // TODO: Remove experimental status from itself and advance SinceKotlin to 1.4
public annotation class ExperimentalStdlibApi
```

---

Open Source Used in DNAC Platform Antelope 4151
of an annotated function parameter or its extensions for that type. The extensions must be themselves annotated with `@BuilderInference`. For example: we declare:

```kotlin
fun <T> sequence(@BuilderInference block: suspend SequenceScope<T>.() -> Unit): Sequence<T>
```

and use it like:

```kotlin
val result = sequence {
    yield("result")
}
```

Here the type argument of the resulting sequence is inferred to `String` from `block`.

Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.

*/
@Target(VALUE_PARAMETER, FUNCTION, PROPERTY)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.3")
@ExperimentalTypeInference
unpublic annotation class BuilderInference

Any usage of a declaration annotated with `@ExperimentalMultiplatform` must be accepted either by

```kotlin
@UseExperimental(ExperimentalMultiplatform::class)
```

or by using the compiler argument `-Xuse-experimental=ktor.ExperimentalMultiplatform`.

*/
@Experimental
@MustBeDocumented
@Target(CLASS, ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, TYPEALIAS)
@Retention(AnnotationRetention.BINARY)
@RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)
public annotation class ExperimentalMultiplatform

```kotlin
@Target(ANNOTATION_CLASS)
@Retention(AnnotationRetention.BINARY)
@ExperimentalMultiplatform
@RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)
public annotation class OptionalExpectation
```

The collection is covariant on its element type:

```kotlin
@UnsafeVariance E
```

The provided skeletal implementation of iterators can be simplified

```kotlin
@SinceKotlin("1.1")
private enum class State {
    Ready,
    NotReady,
    Done,
    Failed
}
```

Open Source Used In DNAC Platform Antelope 4152

| Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. */

| *package kotlin
| import kotlin.annotation.AnnotationTarget.*
| import kotlin.internal.RequireKotlin
| import kotlin.internal.RequireKotlinVersionKind

| Any usage of a declaration annotated with `@ExperimentalMultiplatform` must be accepted either by
|
| ```kotlin
| @UseExperimental(ExperimentalMultiplatform::class)
| ```
| or by using the compiler argument `-Xuse-experimental=ktor.ExperimentalMultiplatform`.

| */
| @Experimental
| @MustBeDocumented
| @Target(CLASS, ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, TYPEALIAS)
| @Retention(AnnotationRetention.BINARY)
| @RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)
| public annotation class ExperimentalMultiplatform

| * Marks an expected annotation class that it isn't required to have actual counterparts in all platforms. * This annotation is only applicable to `expect` annotation classes in multi-platform projects and marks that class as `Optional`. * Optional expected class is allowed to have no corresponding actual class on the platform. Optional annotations can only be used on * to annotate something, not as types in signatures. If an optional annotation has no corresponding actual class on a platform, * the annotation entries where it's used are simply erased when compiling code on that platform.
| *
| Note: this annotation is experimental, see [ExperimentalMultiplatform] on how to opt-in for it.

| */
| @Target(ANNOTATION_CLASS)
| @Retention(AnnotationRetention.BINARY)
| @ExperimentalMultiplatform
| @RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)
| public annotation class OptionalExpectation

| A base class to simplify implementing iterators so that
entryIterator = entries.iterator()
return object : Iterator<V> {
override fun hasNext(): Boolean = entryIterator.hasNext()
override fun next(): V = entryIterator.next().value
}

override val size: Int get() = this@AbstractMap.size
}

private fun implFindEntry(key: K): Map.Entry<K, V>? = entries.firstOrNull { it.key == key }
(v1 == v2) {\n    continue\n} else if (v1 == null || v2 == null) {\n    return false\n}\n
when {\n    v1 is Array<*> && v2 is Array<*> -> if (!v1.contentDeepEquals(v2)) return false\n    v1 is IntArray && v2 is IntArray -> if (!v1.contentEquals(v2)) return false\n    v1 is LongArray && v2 is LongArray -> if (!v1.contentEquals(v2)) return false\n    v1 is ByteArray && v2 is ByteArray -> if (!v1.contentEquals(v2)) return false\n    v1 is ShortArray && v2 is ShortArray -> if (!v1.contentEquals(v2)) return false\n    v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false\n    v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false\n    v1 is ByteArray && v2 is ByteArray -> if (!v1.contentEquals(v2)) return false\n    v1 is ShortArray && v2 is ShortArray -> if (!v1.contentEquals(v2)) return false\n    v1 is IntArray && v2 is IntArray -> if (!v1.contentEquals(v2)) return false\n    v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false\n    v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false\n    v1 is UByteArray && v2 is UByteArray -> if (!v1.contentEquals(v2)) return false\n    v1 is UShortArray && v2 is UShortArray -> if (!v1.contentEquals(v2)) return false\n    v1 is UIntArray && v2 is UIntArray -> if (!v1.contentEquals(v2)) return false\n    v1 is ULongArray && v2 is ULongArray -> if (!v1.contentEquals(v2)) return false\n    v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false\n    v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false\n
    v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false\n    v1 is IntArray && v2 is IntArray -> if (!v1.contentEquals(v2)) return false\n    v1 is LongArray && v2 is LongArray -> if (!v1.contentEquals(v2)) return false\n    v1 is FloatArray && v2 is FloatArray -> if (!v1.contentEquals(v2)) return false\n    v1 is DoubleArray && v2 is DoubleArray -> if (!v1.contentEquals(v2)) return false\n    v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false\n    v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false\n}

return true\n}

@SinceKotlin("1.3")
@PublishedApi
@kotlin.jvm.JvmName("contentDeepToString")
@kotlin.js.JsName("contentDeepToStringImpl")
internal fun <T> Array<out T>.contentDeepToStringImpl(): String {
    val length = size.coerceAtMost((Int.MAX_VALUE - 2) / 5) * 5 + 2 // in order not to overflow Int.MAX_VALUE
    return buildString(length) {
        contentDeepToStringInternal(this, mutableListOf())
    }
}

@UseExperimental(ExperimentalUnsignedTypes::class)
private fun <T> Array<out T>.contentDeepToStringInternal(result: StringBuilder, processed: MutableList<Array<*>>) {
    if (this in processed) {
        result.append("[...]")
        return
    }
    processed.add(this)
    result.append('[

    for (i in indices) {
        if (i != 0) {
            result.append(\',\n        }\n        val element = this[i]
        when (element) {
            null            -> result.append("null")
            is Array<*>     -> element.contentDeepToStringInternal(result, processed)
            is ByteArray    -> result.append(element.contentToString())
            is ShortArray   -> result.append(element.contentToString())
            is IntArray     -> result.append(element.contentToString())
            is LongArray    -> result.append(element.contentToString())
            is FloatArray   -> result.append(element.contentToString())
            is DoubleArray  -> result.append(element.contentToString())
            is CharArray    -> result.append(element.contentToString())
            is BooleanArray -> result.append(element.contentToString())
            is UByteArray   -> result.append(element.contentToString())
            is UShortArray  -> result.append(element.contentToString())
            is UIntArray    -> result.append(element.contentToString())
            is ULongArray   -> result.append(element.contentToString())
            else            -> result.append(element.toString())
        }
    }

    result.append(']

    processed.removeAt(processed.lastIndex)
}

public data class IndexedValue<out T>(public val index: Int, public val value: T)

package kotlin.collections

import kotlin.reflect.KProperty
import kotlin.internal.Exact

public inline operator fun <V, V1 : V> Map<in String, @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =
    @Suppress("UNCHECKED_CAST") (getOrImplicitDefault(property.name) as V1)
of the property for the given object from this mutable map.

* @param thisRef the object for which the value is requested (not used).
* @param property the metadata for the property, used to get the name of property and look up the value corresponding to this name in the map.
* @return the property value.
* @throws NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit default (see [withDefault]).

* @param property the metadata for the property, used to get the name of property and store the value associated with that name in the map.
* @param value the value to set.

*/
@kotlin.jvm.JvmName("getVar")
@kotlin.internal.InlineOnly
public inline operator fun <V, V1 : V> MutableMap<in String, out @Exact V>.getValue(thisRef: Any?, property: KProperty<*>) : V1 =
    @Suppress("UNCHECKED_CAST") (getOrImplicitDefault with two type parameters instead", level = DeprecationLevel.ERROR)@kotlin.jvm.JvmName("getVarContravariant")@kotlin.internal.LowPriorityInOver loadResolution@kotlin.internal.InlineOnly
public inline fun <V> MutableMap<in String, in V>.getValue(thisRef: Any?, property: KProperty<*>): V =
    @Suppress("UNCHECKED_CAST") (getOrImplicitDefault(property.name) as V)

/**
 * Stores the value of the property for the given object in this mutable map.
 * @param thisRef the object for which the value is requested (not used).
 * @param property the metadata for the property, used to get the name of property and store the value associated with that name in the map.
 * @param value the value to set.
 */
@kotlin.internal.InlineOnly
public inline operator fun <V>
    MutableMap<in String, in V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {
    this.put(property.name, value)
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
 * Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("MapsKt")

package
kotlin.collections

/**
 * Returns the value for the given key, or the implicit default value for this map.
 * By default no implicit value is provided for maps and a [NoSuchElementException] is thrown.
 * To create a map with implicit default value use [withDefault] method.
 * @throws NoSuchElementException when the map doesn't contain a value for the specified key and no implicit default was provided for that map.
 */
@kotlin.jvm.JvmName("getOrImplicitDefaultNullable")
@PublishedApi
internal fun <K, V> Map<K, V>.getOrImplicitDefault(key: K): V {
    if (this is MapWithDefault)
        return this.getOrImplicitDefault(key)

    return getOrElseNullable(key, { throw NoSuchElementException("Key $key is missing in the map.") })
}

/**
 * Returns a wrapper of this read-only map, having the implicit default value provided with the specified function [defaultValue].
 * This implicit default value is used when the original map doesn't contain a value for the key specified and a value is obtained with [Map.getValue] function, for example when properties are delegated to the map.
 * When this map already has an implicit default value provided with a former call to [withDefault], it is being replaced by this call. A map with default value provided with a former call to [withDefault] is being replaced by this call.
 */
public fun <K, V> Map<K, V>.withDefault(defaultValue: (key: K) -> V): Map<K, V> =
    when (this) {
        is MapWithDefault -> this.map.withDefault(defaultValue)
        else -> MapWithDefaultImpl(this, defaultValue)
    }

/**
 * Returns a wrapper of this mutable map, having the implicit default value provided with the specified function [defaultValue].
 */
@kotlin.jvm.JvmName("withDefaultMutable")
public fun <K, V> MutableMap<K, V>.withDefault(defaultValue: (key: K) -> V): MutableMap<K, V> =
    when (this) {
        is MutableMapWithDefault -> this.map.withDefault(defaultValue)
        else -> MutableMapWithDefaultImpl(this, defaultValue)
    }

private interface MapWithDefault<K, out V> : Map<K, V> {
    public val map: Map<K, V>
    public fun getOrImplicitDefault(key: K): V
}

private interface MutableMapWithDefault<K, V> : MutableMap<K, V>, MapWithDefault<K, V>, MapWithDefaultImpl<K, V> {
    public override val map: MutableMap<K, V>
    public override fun getOrImplicitDefault(key: K): V
}

private
class MapWithDefaultImpl<K, out V>(public override val map: Map<K, V>, private val default: (key: K) -> V) :
    MapWithDefault<K, V> {
    override fun equals(other: Any?): Boolean = map.equals(other)
    override fun hashCode(): Int = map.hashCode()
    override fun toString(): String = map.toString()
    override val size: Int get() = map.size
    override fun equals(other: Any?): Boolean = map.equals(other)
    override fun containsKey(key: K): Boolean = map.containsKey(key)
    override fun containsValue(value: V): Boolean = map.containsValue(value)
    override fun remove(key: K): V? = map.remove(key)
    override fun put(key: K, value: V): V? = map.put(key, value)
    override fun putAll(map: Map<K, V>): Mutabl...
map.containsKey(key)
    override fun containsValue(value: @UnsafeVariance V): Boolean =
    map.containsKey(value)
    override fun getOrImplicitDefault(key: K): V =
    map.getOrElseNullable(key, { default(key) })
    override fun getOrImplicitDefault(key: K, default: (key: K) -> V): V =
    map.getOrElseNullable(key, { default(key) })

private class MutableMapWithDefaultImpl<K, V>(public override val map: MutableMap<K, V>, private val default: (key: K) -> V) : MutableMapWithDefault<K, V> {
    override fun equals(other: Any?): Boolean =
    map.equals(other)
    override fun hashCode(): Int =
    map.hashCode()
    override fun toString(): String =
    map.toString()
    override val size: Int get() =
    map.size
    override fun isEmpty(): Boolean =
    map.isEmpty()
    override fun containsKey(key: K): Boolean =
    map.containsKey(key)
    override fun containsValue(value: @UnsafeVariance V): Boolean =
    map.containsValue(value)
    override fun get(key: K): V? =
    map.get(key)
    override val keys: Set<K> get() =
    map.keys
    override val values: Collection<V> get() =
    map.values
    override val entries: Set<Map.Entry<K, V>> get() =
    map.entries

    override fun put(key: K, value: V): V? =
    map.put(key, value)
    override fun remove(key: K): V? =
    map.remove(key)
    override fun putAll(from: Map<out K, V>) =
    map.putAll(from)
    override fun clear() =
    map.clear()
    override fun getOrImplicitDefault(key: K): V =
    map.getOrElseNullable(key, { default(key) })
}

import kotlin.random.Random

/**
 * Removes a single instance of the specified element
 * from this * collection, if it is present.
 * Allows to overcome type-safety restriction of `remove` that requires
 * to pass an element of type `E`: `@return `true` if the element has been successfully removed; `false` if it was
 * not present in the collection.
 */
@kotlin.internal.InlineOnly
public inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.remove(element: T): Boolean =
    @Suppress("UNCHECKED_CAST") (this as
    MutableCollection<T>).remove(element)

/**
 * Removes all of this collection's elements that are also
 * contained in the specified collection.
 * Allows to overcome type-safety restriction of `removeAll` that requires
 * to pass a collection of type `Collection<@kotlin.internal.InlineOnly@public inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.removeAll(elements: Collection<T>): Boolean =
    @Suppress("UNCHECKED_CAST") (this as
    MutableCollection<T>).removeAll(elements)

/**
 * Retains only the elements in this collection that are contained in the specified collection.
 * Allows to overcome type-safety restriction of `retainAll` that requires
 * to pass a collection of type `Collection<@kotlin.internal.InlineOnly@public inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.retainAll(elements: Collection<T>): Boolean =
    @Suppress("UNCHECKED_CAST") (this as
    MutableCollection<T>).retainAll(elements)

/**
 * Removes a single instance of the specified element
 * from this mutable collection.
 */
@kotlin.internal.InlineOnly
public inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<in T>.removeAt(index: Int): T =
    this.removeAt(index)

/**
 * Adds the specified element to this mutable collection.
 */
@kotlin.internal.InlineOnly
operator fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<in T>.plusAssign(element: T): T =
    this.add(element)

/**
 * Adds all elements of the given collection to this mutable collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<in T>.plusAssign(enumerable: Iterable<T>): T =
    this.addAll(enumerable)

/**
 * Adds all elements of the given array to this mutable collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<in T>.plusAssign(elements: Array<T>): T =
    this.addAll(elements)

/**
 * Adds all elements of the given sequence to this mutable collection.
 */
@kotlin.internal.InlineOnly
public inline operator fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<in T>.plusAssign(elements: Sequence<T>): T =
    this.addAll(elements)

/**
 * Removes a single instance of the specified element from this mutable
T>.removeAll(elements: Array<out T>): Boolean {n return elements.isNotEmpty() && 
removeAll(elements.toHashSet())}un/**n * Retains only elements of this [MutableCollection] that are contained in the given [elements] collection.

T>.retainAll(elements: Array<out T>): Boolean {n if (elements.isNotEmpty())n return 
retainAll(elements.toHashSet()) else n return retainNothing()}un/**n * Retains only elements of this [MutableCollection] that are contained in the given [elements] sequence.

/**
 * Retains only elements of this [MutableCollection] that are contained in the given [elements] collection.
 */
public fun <T> MutableCollection<in T>.retainAll(elements: Iterable<T>): Boolean {
    return retainAll(elements.toListForSetOperationWith(this))n/**n * Randomly shuffles elements in this mutable list using the specified [random] instance as the source of randomness.
}

private fun MutableList<*>.retainNothing(): Boolean {
    val result = isNotEmpty()
    clear()
    return 
}

/**
 * Randomly shuffles elements in this mutable list using the specified [random] instance as the source of randomness.
 *
 * See:
 * https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm
 */
@SinceKotlin("1.3")
public fun <T> MutableList<T>.shuffle(random: Random): Unit {
    for (i in lastIndex 
downTo 1) {
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
    }
}

/**
 * Returns a new list with the elements of this list randomly shuffled using the specified [random] instance as the source of randomness.
 */
public fun <T> List<T>.asReversed(): List<T> = ReversedListReadOnly(this)

/**
 * Returns a new list with the elements of this list randomly shuffled using the specified [random] instance as the source of randomness.
 */
@kotlin.jvm.JvmName("asReversedMutable")
public fun <T> MutableList<T>.asReversed(): MutableList<T> = ReversedList(this)
null
        private fun nextNotReady(): T
            return suspendCoroutineUninterceptedOrReturn { c ->
                var nextStep = c
                COROUTINE_SUSPENDED
            }

    private fun exceptionalState(): Throwable = when (state) {
        State_Done -> NoSuchElementException()
        State_Failed -> IllegalStateException("Iterator has failed.")
        else -> IllegalStateException("Unexpected state of the iterator: $state")
    }

    override fun resumeWith(result: Result<Unit>) {
        result.getOrThrow() // just rethrow exception if it is there
        state = State_Done
    }

    override val context: CoroutineContext
        get() = EmptyCoroutineContext
}

package kotlin.collections

internal fun checkWindowSizeStep(size: Int, step: Int) {
    require(size > 0 && step > 0) {
        if (size != step)
            "Both size $size and step $step must be greater than zero."
        else
            "size $size must be greater than zero."
    }
}

    checkWindowSizeStep(size, step)
    return Sequence {
        windowedIterator(iterator(), size, step, partialWindows, reuseBuffer) }
}

    if (!iterator.hasNext()) return EmptyIterator
    return iterator<List<T>> {
        val bufferInitialCapacity = size.coerceAtMost(1024)
        val gap = step - size
        if (gap >= 0) {
            var buffer = ArrayList<T>(bufferInitialCapacity)
            var skip = 0
            for (e in iterator) {
                if (skip > 0) {
                    skip -= 1; continue }
                buffer.add(e)
                if (buffer.size == size) {
                    yield(buffer)
                    if (reuseBuffer) buffer.clear() else buffer = ArrayList(size)
                    skip = gap
                }
            }
            if (buffer.isNotEmpty()) {
                if (partialWindows || buffer.size == size) yield(buffer)
            }
        } else {
            var buffer = RingBuffer<T>(bufferInitialCapacity)
            for (e in iterator) {
                buffer.add(e)
                if (buffer.isFull()) {
                    if (buffer.size < size) { buffer = buffer.expanded(maxCapacity = size); continue }
                    buffer.removeFirst(step)
                }
            }
            if (partialWindows) {
                while (buffer.size > step) {
                    yield(if (buffer.isEmpty()) buffer else ArrayList(buffer))
                    buffer.removeFirst(step)
                }
            }
        }
    }
}

internal class MovingSubList<out E>(private val list: List<E>) : AbstractList<E>(), RandomAccess {
    private var fromIndex: Int = 0
    private var _size: Int = 0

    fun move(fromIndex: Int, toIndex: Int) {
        checkRangeIndexes(fromIndex, toIndex, list.size)
        this.fromIndex = fromIndex
        this._size = toIndex - fromIndex
    }

    override fun get(index: Int): E {
        checkElementIndex(index, _size)

        return list[fromIndex + index]
    }

    override val size: Int get() = _size
}

/**
 * Provides ring buffer implementation.
 *
 * Buffer overflow is not allowed so [add] doesn't overwrite tail but raises an exception.
 */
private class RingBuffer<T>(private val buffer: Array<Any?>, filledSize: Int) : AbstractList<T>(), RandomAccess {
    init {
        require(filledSize >= 0) { "ring buffer filled size should not be negative but it is $filledSize" }
        require(filledSize <= buffer.size) { "ring buffer filled size: $filledSize cannot be larger than the buffer size: ${buffer.size}" }
        if (buffer.size < filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.removeFirst(step)
    }

    fun isFull() = buffer.size == filledSize

    override fun iterator(): Iterator<T> = object : AbstractIterator<T>() {
        @Suppress("UNCHECKED_CAST")
        return buffer.iterator() as T
    }

    override fun removeFirst(step: Int) {
        if (buffer.size == filledSize) {
            buffer = buffer.expanded(maxCapacity = filledSize)
        }
        buffer.removeFirst(step)
    }

    override fun insertAt(position: Int, element: T) {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.insertAt(position, element)
    }

    override fun setAt(position: Int, element: T) {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.setAt(position, element)
    }

    override fun removeAt(position: Int) {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.removeAt(position)
    }

    override fun clear() {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.clear()
    }

    override fun get(index: Int): T {
        @Suppress("UNCHECKED_CAST")
        return buffer.get(index) as T
    }

    override fun set(index: Int, element: T) {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.set(index, element)
    }

    override fun add(element: T) {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.add(element)
    }

    override fun removeAt(index: Int) {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.removeAt(index)
    }

    override fun size() = buffer.size

    override fun isEmpty() = buffer.isEmpty()

    override fun contains(element: T): Boolean {
        return buffer.contains(element)
    }

    override fun containsAll(elements: Collection<T>): Boolean {
        return buffer.containsAll(elements)
    }

    override fun iterator(): Iterator<T> = object : AbstractIterator<T>() {
        @Suppress("UNCHECKED_CAST")
        return buffer.iterator() as T
    }

    override fun removeFirst() {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.removeFirst()
    }

    override fun removeLast() {
        if (buffer.size == filledSize) { buffer = buffer.expanded(maxCapacity = filledSize); return }
        buffer.removeLast()
    }

    override fun containsAll(elements: Collection<T>): Boolean {
        return buffer.containsAll(elements)
    }

    override fun lastIndexOf(element: T): Int {
        return buffer.lastIndexOf(element)
    }

    override fun indexOf(element: T): Int {
        return buffer.indexOf(element)
    }

    override fun equals(other: Any?): Boolean {
        return buffer.equals(other)
    }

    override fun hashcode(): Int {
        return buffer.hashcode()
    }

    override fun toString(): String {
        return buffer.toString()
    }
}

Open Source Used In DNAC Platform Antelope  4162
fun quickSort(array: UIntArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1) {
        quickSort(array, left, index - 1)
    }
    if (index < right) {
        quickSort(array, index, right)
    }
}

@ExperimentalUnsignedTypes
private fun partition(array: UIntArray, left: Int, right: Int): Int {
    var index = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot) {
            i++
        }
        while (array[j] > pivot) {
            j--
        }
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun <T> quickSort(array: Array<T>, left: Int, right: Int) {
    if (array.size < this.size) array.copyOf(this.size) else array as Array<T> as Array<T>

    var index = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot) {
            i++
        }
        while (array[j] > pivot) {
            j--
        }
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return index
}

@Suppress("UNCHECKED_CAST")
@ExperimentalUnsignedTypes
private fun <T> quickSort(array: Array<T>, left: Int, right: Int): Array<T> {
    if (array.size < this.size) array.copyOf(this.size) else array as Array<T> as Array<T>

    var index = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot) {
            i++
        }
        while (array[j] > pivot) {
            j--
        }
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return index
}

@ExperimentalUnsignedTypes
private fun <T> quickSort(array: Array<T>, left: Int, right: Int): Array<T> {
    if (array.size < this.size) array.copyOf(this.size) else array as Array<T> as Array<T>

    var index = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot) {
            i++
        }
        while (array[j] > pivot) {
            j--
        }
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return index
}

@Suppress("UNCHECKED_CAST")
@ExperimentalUnsignedTypes
private fun <T> quickSort(array: Array<T>, left: Int, right: Int): Array<T> {
    if (array.size < this.size) array.copyOf(this.size) else array as Array<T> as Array<T>

    var index = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot) {
            i++
        }
        while (array[j] > pivot) {
            j--
        }
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return index
}

@Suppress("UNCHECKED_CAST")
@ExperimentalUnsignedTypes
private fun <T> quickSort(array: Array<T>, left: Int, right: Int): Array<T> {
    if (array.size < this.size) array.copyOf(this.size) else array as Array<T> as Array<T>

    var index = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot) {
            i++
        }
        while (array[j] > pivot) {
            j--
        }
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return index
}

@Suppress("UNCHECKED_CAST")
@ExperimentalUnsignedTypes
private fun <T> quickSort(array: Array<T>, left: Int, right: Int): Array<T> {
    if (array.size < this.size) array.copyOf(this.size) else array as Array<T> as Array<T>

    var index = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot) {
            i++
        }
        while (array[j] > pivot) {
            j--
        }
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return index
}

@Suppress("UNCHECKED_CAST")
@ExperimentalUnsignedTypes
private fun <T> quickSort(array: Array<T>, left: Int, right: Int): Array<T> {
    if (array.size < this.size) array.copyOf(this.size) else array as Array<T> as Array<T>

    var index = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot) {
            i++
        }
        while (array[j] > pivot) {
            j--
        }
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return index
}
private fun partition(array: ULongArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

private fun quickSort(array: ULongArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

/**
 * Sorts the given array using qsort algorithm.
 */
@ExperimentalUnsignedTypes
internal fun sortArray(array: UByteArray) = quickSort(array, 0, array.size - 1)
@ExperimentalUnsignedTypes
internal fun sortArray(array: UShortArray) = quickSort(array, 0, array.size - 1)
@ExperimentalUnsignedTypes
internal fun sortArray(array: UIntArray) = quickSort(array, 0, array.size - 1)
@ExperimentalUnsignedTypes
internal fun sortArray(array: ULongArray) = quickSort(array, 0, array.size - 1)
*\n@ContractsDsl
@ExperimentalContracts
@SinceKotlin("1.3")
public enum class InvocationKind {
    AT_MOST_ONCE, AT_LEAST_ONCE, EXACTLY_ONCE, UNKNOWN
}

/**
 * Specifies how many times a function invokes its function parameter in place.
 * @param kind InvocationKind
 */
@ContractsDsl
@ExperimentalContracts
@SinceKotlin("1.3")
public class InvocationKind {
    public enum class InvocationKind {
        AT_MOST_ONCE, AT_LEAST_ONCE, EXACTLY_ONCE, UNKNOWN
    }

    /**
     * Specifies the contract of a function.
     */
    @ContractsDsl
    @ExperimentalContracts
    @InlineOnly
    @SinceKotlin("1.3")
    public inline fun contract(builder: ContractBuilder.() -> Unit) { }

    public interface ContinuationInterceptor : CoroutineContext.Element {
        companion object Key : CoroutineContext.Key<ContinuationInterceptor>

        fun interceptContinuation(continuation: Continuation<T>): Continuation<T>

        fun releaseInterceptedContinuation(continuation: Continuation<*>) {
            /* do nothing by default */
        }
    }

        @Suppress("UNCHECKED_CAST")
        if (key === Key) this as E else null

    public override fun minusKey(key: CoroutineContext.Key<*>): CoroutineContext =
        if (key === Key) EmptyCoroutineContext else this
}

The key that defines the context interceptor.

CoroutineContext.Key<ContinuationInterceptor>:

Returns continuation that wraps the original continuation, thus intercepting all resumptions.

This function is invoked by coroutines framework when needed and the resulting continuations are cached internally per each instance of the original continuation.

This function may simply return original continuation if it does not want to intercept this particular continuation.

When the original continuation completes, coroutine framework invokes

[releaseInterceptedContinuation] with the resulting continuation if it was intercepted, that is if

`interceptContinuation` had previously returned a different continuation instance.

`public fun <T> interceptContinuation(continuation: Continuation<T>): Continuation<T>`

`@param continuation Continuation instance returned by this interceptor's `interceptContinuation` invocation.`

`public fun releaseInterceptedContinuation(continuation: Continuation<T>)`
Persistent context for the coroutine. It is an indexed set of [Element] instances. An indexed set is a mix between a set and a map.

Every element in this set has a unique [Key].

- Returns the element with the given [key] from this context or `null` if none.
- Accumulates entries of this context starting with [initial] value and applying [operation] from left to right to current accumulator value and each element of this context.
- Returns a context containing elements from this context and elements from other context. The elements from this context with the same key as in the other one are dropped.

- Returns the element with the given [key] from this context or `null`.
- Accumulates entries of this context starting with [initial] value and applying [operation] from left to right to current accumulator value and each element of this context.

The elements from this context with the same key as in the other one are dropped.

Persistent context for the coroutine. It is an indexed set of [Element] instances.

Every element in this set has a unique [Key].

- Returns the element with the given [key] from this context or `null`.
- Accumulates entries of this context starting with [initial] value and applying [operation] from left to right to current accumulator value and each element of this context.

The elements from this context with the same key as in the other one are dropped.

// this class is not exposed, but is hidden inside implementations

internal class CombinedContext(
    private val left: CoroutineContext,
    private val element: Element
) : CoroutineContext, Serializable {

    override fun <E : Element> get(key: Key<E>): E?
        if (next is CombinedContext) { return it } else { return next }<br>

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(left.fold(initial, operation), element)

    public override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }<br>
        val newLeft = left.minusKey(key)
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element
            else -> CombinedContext(newLeft, element)
        }
    }

    private fun size(): Int {
        var cur = this
        var size = 0
        while (true) {
            cur.element[key]?.let { return it }
            val next = cur.next
            if (next is CombinedContext) { return it } else { return next }<br>

            val newLeft = left.minusKey(key)
            return when {
                newLeft === left -> this
                newLeft === EmptyCoroutineContext -> element
                else -> CombinedContext(newLeft, element)
            }
        }
    }

    private fun readResolve(): Any = EmptyCoroutineContext

    public override fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)

    public override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }<br>
        val newLeft = left.minusKey(key)
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element
            else -> CombinedContext(newLeft, element)
        }
    }

    public override fun hashCode(): Int = 0

    public override fun toString(): String = "EmptyCoroutineContext"
}

public override fun plus(context: CoroutineContext): CoroutineContext =
    if (context === EmptyCoroutineContext) this else // fast path -- avoid lambda creation
    context.fold(this) { acc, element ->
        val removed = acc.minusKey(element.key)
        if (removed === EmptyCoroutineContext) element else {
            // make sure interceptor is always last in the context (and thus is fast to get when present)
            if (interceptor == null) CombinedContext(removed, element) else {
                // make sure interceptor is always last in the context (and thus is fast to get when present)
                val interceptor = removed[ContinuationInterceptor]<br>
                CombinedContext(element, interceptor) else
                CombinedContext(CombinedContext(left, element), interceptor)<br>
            }<br>
        }
    }

public override fun minusKey(key: Key<*>): CoroutineContext {
    element[key]?.let { return left }<br>
    val newLeft = left.minusKey(key)
    return when {
        newLeft === left -> this
        newLeft === EmptyCoroutineContext -> element
        else -> CombinedContext(newLeft, element)<br>
    }

    public override fun hashCode(): Int = 0

    public override fun toString(): String = "EmptyCoroutineContext"
}

--- internal impl ---

// this class is not exposed, but is hidden inside implementations

internal class CombinedContext(
    private val left: CoroutineContext,
    private val element: Element<br>
) : CoroutineContext, Serializable {

    override fun <E : Element> get(key: Key<E>): E?
        if (next is CombinedContext) { return it } else { return next }<br>

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(left.fold(initial, operation), element)<br>

    public override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }<br>
        val newLeft = left.minusKey(key)<br>
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element<br>
            else -> CombinedContext(newLeft, element)\n        }
    }

    private fun size(): Int {
        var cur = this
        var size = 0
        while (true) {
            cur.element[key]?.let { return it }
            val next = cur.next
            if (next is CombinedContext) { return it } else { return next }<br>

            val newLeft = left.minusKey(key)<br>
            return when {
                newLeft === left -> this
                newLeft === EmptyCoroutineContext -> element<br>
                else -> CombinedContext(newLeft, element)\n            }
        }
    }

    private fun readResolve(): Any = EmptyCoroutineContext

    public override fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null<br>

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)<br>

    public override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }<br>
        val newLeft = left.minusKey(key)<br>
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element<br>
            else -> CombinedContext(newLeft, element)\n        }
    }

    public override fun hashCode(): Int = 0<br>

    public override fun toString(): String = "EmptyCoroutineContext"
}

--- Base class for [CoroutineContext.Element] implementations ---

public abstract class AbstractCoroutineContextElement(public override val key: Key<*>) : Element

--- Empty coroutine context ---

public object EmptyCoroutineContext : CoroutineContext, Serializable {
    private const val serialVersionUID: Long = 0

    private fun readResolve(): Any = EmptyCoroutineContext

    public override fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)

    public override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }<br>
        val newLeft = left.minusKey(key)<br>
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element<br>
            else -> CombinedContext(newLeft, element)\n        }
    }

    public override fun hashCode(): Int = 0<br>

    public override fun toString(): String = "EmptyCoroutineContext"
}
private fun contains(element: Element): Boolean =
    get(element.key) == element

    private fun containsAll(context: CombinedContext): Boolean {
        var cur = context
        while (true) {
            if (!contains(cur.element)) return false
            val next = cur.left
            if (next is CombinedContext) {
                cur = next
            } else {
                return contains(next as Element)
            }
        }
    }

    override fun equals(other: Any?): Boolean =
        this === other || other is CombinedContext && other.size() == size() &&
        other.containsAll(this)

    override fun hashCode(): Int = left.hashCode() + element.hashCode()

    override fun toString(): String =
        "[{" + fold("") { acc, element ->
            if (acc.isEmpty()) element.toString() else
            "$acc, $element"
        } + "}"

    private fun writeReplace(): Any {
        val n = size()
        val elements =
        arrayOfNulls<CoroutineContext>(n)
        var index = 0
        fold(Unit) { _, element -> elements[index++] =
            element }
        check(index == n)
        @Suppress("UNCHECKED_CAST")
        return Serialized(elements as Array<CoroutineContext>)
    }

    private class Serialized(val elements: Array<CoroutineContext>) :
        Serializable {
        companion object {
            private const val serialVersionUID: Long = 0L
        }

        private fun readResolve(): Any =
            elements.fold(EmptyCoroutineContext, CoroutineContext::plus)
    }

    @file:kotlin.jvm.JvmName("IntrinsicsKt")
    @file:kotlin.jvm.JvmMultifileClass
    package kotlin.coroutines.intrinsics

    import kotlin.coroutines.*
    import kotlin.internal.InlineOnly

    @SinceKotlin("1.3")
    @InlineOnly
    @Suppress("UNUSED_PARAMETER", "RedundantSuspendModifier")
    public suspend inline fun <T>
        suspendCoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T =
        throw
        NotImplementedException("Implementation of suspendCoroutineUninterceptedOrReturn is intrinsic")

    @SinceKotlin("1.3")
    @PublishedApi // This class is Published API via serialized representation of SafeContinuation, don't rename/move
    @RedundantSuspendModifier
    public suspend inline fun <T>
        suspendCoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T =
        throw
        NotImplementedException("Implementation of suspendCoroutineUninterceptedOrReturn is intrinsic")

    // This class is Published API via serialized representation of SafeContinuation, don't rename/move\internal enum
class CoroutineSingletons { COROUTINE_SUSPENDED, UNDECIDED, RESUMED }

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.experimental

/** Performs a bitwise AND operation between the two values. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.and(other: Byte): Byte =
    (this.toInt() and other.toInt()).toByte()/** Performs a bitwise OR operation between the two values. */

*@
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.or(other: Byte): Byte =
    (this.toInt() or other.toInt()).toByte()/** Performs a bitwise XOR operation between the two values. */

*@
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.xor(other: Byte): Byte =
    (this.toInt() xor other.toInt()).toByte()/** Inverts the bits in this value. */

*@
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun Byte.inv(): Byte =
    (this.toInt().inv()).toByte()/** Performs a bitwise AND operation between the two values. */

*@
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Short.and(other: Short): Short =
    (this.toInt() and other.toInt()).toShort()/** Performs a bitwise OR operation between the two values. */

*@
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Short.or(other: Short): Short =
    (this.toInt() or other.toInt()).toShort()/** Performs a bitwise XOR operation between the two values. */

*@
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Short.xor(other: Short): Short =
    (this.toInt() xor other.toInt()).toShort()/** Inverts the bits in this value. */

*@
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun Short.inv(): Short =
    (this.toInt().inv()).toShort()

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.experimental

/**
 * The experimental marker for type inference augmenting annotations.
 * Any usage of a declaration annotated with @Experimental should be accepted either by
 * by using the compiler argument `-Xuse-experimental=kotlin.experimental.ExperimentalTypeInference`
 */
@Experimental(level = Experimental.Level.ERROR)
@MustBeDocumented
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.ANNOTATION_CLASS)
@SinceKotlin("1.3")
public annotation class ExperimentalTypeInference

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.internal

/**
 * Specifies that the corresponding member has the highest priority in overload resolution. Effectively this means that
 * an extension annotated with this annotation will win in overload resolution over a member with the same signature.
 */
@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY)
@Retention(AnnotationRetention.BINARY)
internal annotation class HidesMembers

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin

/**
 * Specifies that this function should not be called directly without inlining.
 */
@Target(AnnotationTarget.TYPE_PARAMETER)
@Retention(AnnotationRetention.BINARY)
internal annotation class OnlyInputTypes

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.jetbrains.kotlin

/**
 * Specifies that the constraint built for the type during type inference should be an equality one.
 */
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY)
internal annotation class Exact

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.jetbrains.kotlin

/**
 * Specifies that the corresponding type should be ignored during type inference.
 */
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.TYPE)
internal annotation class NoInfer

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.jetbrains.kotlin

/**
 * Specifies that the value of this type parameter should be mentioned in input types (argument types, receiver type or expected type).
 */
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.TYPE_PARAMETER)
internal annotation class OnlyInputTypes

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.jetbrains.kotlin

/**
 * Specifies that the constraint built for the type during type inference should be an inequality one.
 */
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY)
internal annotation class LowPriorityInOverloadResolution

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.jetbrains.kotlin

/**
 * Specifies that the constraint built for the type during type inference should be an inequality one.
 */
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY)
internal annotation class HighPriorityInOverloadResolution

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.jetbrains.kotlin

/**
 * Specifies that the constraint built for the type during type inference should be an inequality one.
 */
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY)
internal annotation class Between

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.jetbrains.kotlin

/**
 * Specifies that the constraint built for the type during type inference should be an inequality one.
 */
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY)
internal annotation class NotBetween
changed, allowing the callback to veto the modification. * @param initialValue the initial value of the property. * @param onChange the callback which is called before a change to the property value is attempted. * The value of the property hasn't been changed yet, when this callback is invoked. * If the callback returns ’true’ the value of the property is being set to the new value, * and if the callback returns ’false’ the new value is discarded and the property remains its old value. * @sample samples.properties.Delegates.vetoableDelegat

```kotlin
public inline fun <T> vetoable(initialValue: T,
        crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Boolean):
        ReadWriteProperty<Any?, T> =
        object : ObservableProperty<T>(initialValue) {
            override fun beforeChange(property: KProperty<*>, oldValue: T, newValue: T): Boolean = onChange(property, oldValue, newValue)
        }

private class NotNullVar<T : Any>() : ReadWriteProperty<Any?, T> {
    private var value: T? = null

    public override fun getValue(thisRef: Any?, property: KProperty<*>): T {
        return value
    }

    public override fun setValue(thisRef: Any?, property: KProperty<*>, value: T) {
        this.value = value
    }
}
```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 *
package kotlin.properties

import kotlin.reflect.KProperty

/**
 * Implements the core logic of a property delegate for a read/write property that calls callback functions when changed.
 * @param initialValue the initial value of the property.
 */
public abstract class ObservableProperty<T>(initialValue: T) :
    ReadWriteProperty<Any?, T> {
        // ... (remaining code)
    }
```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.random

import kotlin.math.nextDown

/**
 * An abstract class that is implemented by random number generator algorithms.
 * The companion object Random.Default is the default instance of Random.
 * @param bitCount number of bits to generate, must be in range 0..32, otherwise the behavior is unspecified.
 * @throws IllegalArgumentException if bitCount is negative or zero.
 */
public abstract class Random {
    // ... (remaining code)
}
```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.math

/**
 * Represents an `Int` whose lower [bitCount] bits are filled with random values and the remaining upper bits are zero.
 * @param bitCount number of bits to generate, must be in range 0..32, otherwise the behavior is unspecified.
 * @throws IllegalArgumentException if bitCount is negative or zero.
 */
public abstract class Random {
    // ... (remaining code)
}
```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin

/**
 * An abstract class that is implemented by random number generator algorithms.
 */
public abstract class Random {
    // ... (remaining code)
}
```

---

Open Source Used In DNAC Platform Antelope 4170
IllegalArgumentException("Cannot get random in empty range: $range")

nextLong(range.first, range.last + 1u)
range.first > Long.MIN_VALUE -> nextLong(range.first - 1u, range.last) + 1u
else -> nextLong()

fun checkRangeBounds(from: Int, until: Int) = require(until > from)
  { boundsErrorMessage(from, until) }
fun checkRangeBounds(from: Long, until: Long) = require(until > from)
  { boundsErrorMessage(from, until) }
fun checkRangeBounds(from: Double, until: Double) = require(until > from)
  { boundsErrorMessage(from, until) }

internal fun boundsErrorMessage(from: Any, until: Any) = "Random range is empty: [$from, $until)."

against ExperimentalUnsignedTypes
public fun Random.nextUInt(): UInt = nextInt().toUInt()

public fun Random.nextUInt(until: UInt): UInt = nextUInt(0u, until)

public fun Random.nextUInt(from: UInt, until: UInt): UInt {
    checkUIntRangeBounds(from, until)

    val signedFrom = from.toInt() xor Int.MIN_VALUE
    val signedUntil = until.toInt() xor Int.MIN_VALUE

    val signedResult = nextInt(signedFrom, signedUntil) xor Int.MIN_VALUE
    return signedResult.toUInt()
}

public fun Random.nextUInt(range: UIntRange): UInt = when {
    range.isEmpty() -> throw
        IllegalArgumentException("Cannot get random in empty range: $range")

    range.last < UInt.MAX_VALUE ->
        nextUInt(range.first, range.last + 1u)

    range.first > UInt.MIN_VALUE -> nextUInt(range.first - 1u, range.last) + 1u
    else -> nextUInt()

    else
        { boundsErrorMessage(range.first, range.last) }
}

public fun Random.nextULong(): ULong = nextLong().toULong()

public fun Random.nextULong(until: ULong): ULong = nextULong(0uL, until)

public fun Random.nextULong(from: ULong, until: ULong): ULong {
    checkULongRangeBounds(from, until)

    val signedFrom = from.toLong() xor Long.MIN_VALUE
    val signedUntil = until.toLong() xor Long.MIN_VALUE

    val signedResult = nextLong(signedFrom, signedUntil) xor Long.MIN_VALUE
    return signedResult.toULong()
}

public fun Random.nextULong(range: ULongRange): ULong = when {
    range.isEmpty() -> throw
        IllegalArgumentException("Cannot get random in empty range: $range")

    range.last < ULong.MAX_VALUE ->
        nextULong(range.first, range.last + 1uL)

    range.first > ULong.MIN_VALUE -> nextULong(range.first - 1uL, range.last) + 1uL
    else -> nextULong()

    else
        { boundsErrorMessage(range.first, range.last) }
}

val signedFrom = from.toULong() xor ULong.MIN_VALUE
val signedUntil = until.toULong() xor ULong.MIN_VALUE
val signedResult = nextULong(signedFrom, signedUntil) xor ULong.MIN_VALUE
return signedResult.toULong()
the next random `ULong` from the random number generator in the specified `[range]`. 

*Generates a `ULong` random value uniformly distributed in the specified `[range]`; `*` from `range.start` inclusive to `range.endInclusive` inclusive. `*` @throws IllegalArgumentException if `[range]` is empty.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextULong(range: ULongRange): ULong = when {
    range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range:
        $range")
    range.last < ULong.MAX_VALUE -> nextULong(range.first, range.last + 1u)
    else -> nextULong(range.first - 1u, range.last) + 1u
}

/**
 * Fills the specified unsigned byte [array] with random bytes and returns it. `*` @return [array] filled with random bytes.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUByteArray(array: UByteArray): UByteArray = nextBytes(array.asByteArray())

/**
 * Creates an unsigned byte array of the specified [size], filled with random bytes.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUByteArray(size: Int): UByteArray = nextBytes(size).asUByteArray()

/**
 * Fills a subrange of the specified `UByte` [array] starting from [fromIndex] inclusive and ending [toIndex] exclusive with random UBytes.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUByteArray(array: UByteArray, fromIndex: Int = 0, toIndex: Int = array.size): UByteArray {
    nextBytes(array.asByteArray(), fromIndex, toIndex)
    return array
}
```

```kotlin
@ExperimentalUnsignedTypes
private fun checkUIntRangeBounds(from: UInt, until: UInt) =
    require(until > from) { boundsErrorMessage(from, until) }
@ExperimentalUnsignedTypes
private fun checkULongRangeBounds(from: ULong, until: ULong) = require(until > from) { boundsErrorMessage(from, until) }
```

```kotlin
package kotlin.random

/**
 * Random number generator, algorithm "xorwow" from p. 5 of Marsaglia, "Xorshift RNGs". Cycles after $2^{160} \cdot (2^{32} - 1)$ repetitions. 
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors
 */
internal class XorWowRandom
internal constructor(
    private var x: Int,
    private var y: Int,
    private var z: Int,
    private var w: Int,
    private var v: Int,
    private var addend: Int
) : Random() {

    internal constructor(seed1: Int, seed2: Int) :
        this(seed1, seed2, 0, 0, seed1.inv(), (seed1 shl 10) xor (seed2 ushr 4))

    init {
        require((x or y or z or w or v) != 0) { "Initial state must have at least one non-zero element." }

        // some trivial seeds can produce several values with zeroes in upper bits, so we discard first 64
        repeat(64) { nextInt() }
    }

    override fun nextInt(): Int {
        var t = x
        t = t xor (t ushr 2)
        x = y
        y = z
        z = w
        val v0 = v
        w = v0
        t = (t xor (t shr 1)) xor v0 xor (v0 shr 4)
        v = t
        val addend = 362437
        return t + addend
    }

    override fun nextBits(bitCount: Int): Int =
        nextInt().takeUpperBits(bitCount)
}
```

```kotlin
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("RangesKt")

package kotlin.ranges

/**
 * Represents a range of floating point numbers.
 * Extends `ClosedRange` interface providing custom operation `lessThanOrEquals` for comparing values of range domain type.
 */
@SinceKotlin("1.1")
public interface ClosedFloatingPointRange<T : Comparable<T>> : ClosedRange<T> {
    override fun contains(value: T): Boolean = lessThanOrEquals(start, value) && lessThanOrEquals(value, endInclusive)
    override fun isEmpty(): Boolean = !lessThanOrEquals(start, endInclusive)

    /**
     * Compares two values of range domain type and returns true if first is less than or equal to second.
     */
    fun lessThanOrEquals(a: T, b: T): Boolean
}
```

```kotlin
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("RangesKt")

package kotlin.ranges

/**
 * Represents a range of `Comparable` values.
 */
private open class ComparableRange<T : Comparable<T>>(
    override val start: T,
    override val endInclusive: T
) : ClosedRange<T> {
override fun toString(): String = "$start..$endInclusive"
}

/**
 * A closed range of values of type `Double`.
 *
 * Numbers are compared
with the ends of this range according to IEEE-754.
 */
private class ClosedDoubleRange(
    start: Double,
    endInclusive: Double
) : ClosedFloatingPointRange<Double> {
    private val _start = start
    private val _endInclusive = endInclusive
    override val start: Double get() = _start
    override val endInclusive: Double get() = _endInclusive
    override fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b
    override fun contains(value: Double): Boolean = value >= _start && value <= _endInclusive
    override fun isEmpty(): Boolean = !(_start <= _endInclusive)
    override fun equals(other: Any?): Boolean {
        return other is ClosedDoubleRange && (isEmpty() && other.isEmpty() ||
            _start == other._start && _endInclusive == other._endInclusive)
    }
    override fun hashCode(): Int {
        return if (isEmpty()) -1 else 31 * _start.hashCode() + _endInclusive.hashCode()
    }
    override fun toString(): String = "$_start..$_endInclusive"
}

/**
 * Creates a range from this `[Comparable]` value to the specified `[that]` value.
 *
 * Numbers are compared with the ends of this range according to IEEE-
 * 754.
 * @sample samples.ranges.Ranges.rangeFromComparable
 */
public operator fun <T : Comparable<T>> T.rangeTo(that: T): ClosedRange<T> = ComparableRange(this, that)

/**
 * Creates a range from this `Double` value to the specified `[that]` value.
 *
 * Numbers are compared with the ends of this range according to IEEE-
 * 754.
 * @sample samples.ranges.Ranges.rangeFromDouble
 */
@SinceKotlin("1.1")
public operator fun Double.rangeTo(that: Double): ClosedFloatingPointRange<Double> = ClosedDoubleRange(this, that)

/**
 * Returns `true` if this iterable range contains the specified `[element]`.
 *
 * Always returns `false` if the `[element]` is `null`.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
    element != null && contains(element)

internal fun checkStepIsPositive(isPositive: Boolean, step: Number) {
    if (!isPositive)
        throw IllegalArgumentException("Step must be positive, was: $step.")
}

/*
 * Copyright 2010-2019
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.reflect

/**
 * Returns a runtime representation of the given reified type `[T]` as an instance of `[KType]`.
 */
@Suppress("unused") // KT-12448
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <reified T> typeOf(): KType =
    throw UnsupportedOperationException("This function is implemented as an intrinsic on all supported platforms.")

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CharsKt")

package kotlin.text

/**
 * Concatenates this Char and a String.
 */
@kotlin.internal.InlineOnly
public inline operator fun Char.plus(other: String): String = this.toString() + other

/**
 * Returns `true` if this character is equal to the `[other]` character, optionally ignoring character case.
 *
 * Two characters are considered the same ignoring case if at least one of the following is `true`:
 *   - The two characters
are the same (as compared by the == operator)
 *   - Applying the method `toUpperCase` to each character produces the same result
 *     * @param ignoreCase `true` to ignore character case when comparing characters. By default `false`
 *     * @param ignoreCase `true` to ignore character case when comparing characters. By default `false`
 *   - The two characters are the same (as compared by the `==` operator)
 * @param ignoreCase `true` to ignore character case when comparing characters. By default `false`
 * @param ignoreCase `true` to ignore character case when comparing characters. By default `false`
 */
public fun Char.equals(other: Char, ignoreCase: Boolean = false): Boolean {
    if (this == other) return true
    if (!ignoreCase) return false
    if (this.toUpperCase() == other.toUpperCase()) return true
    if (this.toLowerCase() == other.toLowerCase()) return true
    return false
}

/**
 * Returns `true` if this character is a Unicode surrogate code unit.
 */
public fun Char.isSurrogate(): Boolean = this in Char.MIN_SURROGATE..Char.MAX_SURROGATE

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("DNACPlatformAntelope")

open class DNACPlatformAntelope
string and removes the first and the last lines if they are blank (notice difference blank vs empty).

* Doesn’t affect a line if it doesn’t contain [marginPrefix] except the first and the last blank lines.*
* Doesn’t preserve the original line endings.*

* @param marginPrefix non-blank string, which is used as a margin delimiter. Default is `|` (pipe character).
* @sample samples.text.Strings.trimMargin

```kotlin
public fun String.trimMargin(marginPrefix: String = "|'): String =
    replaceIndentByMargin("", marginPrefix)
```

* Detects indent by [marginPrefix] as it does [trimMargin] and replace it with [newIndent].

* Doesn’t preserve the original line endings.*

* @param marginPrefix non-blank string, which is used as a margin delimiter.
* @sample samples.text.Strings.trimIndent

```kotlin
public fun String.trimIndent(): String = replaceIndent("")
```

* Detects a common minimal indent of all the input lines, removes it from every line and also preserves the first and the last line if lines are blank (notice difference blank vs empty).*

* Doesn’t preserve the original line endings.*

* @sample samples.text.Strings.trimIndent

```kotlin
private fun String.indentWidth(): Int = indexOfFirst { !it.isWhitespace() }.let { if (it == -1) length else it }
```

* @see trimIndent

```kotlin
private inline fun List<String>.reindent(
    resultSizeEstimate: Int,
    indentAddFunction: (String) -> String,
    indentCutFunction: (String) -> String?
): String {
    val lastIndex = lastIndex
    return
        mapIndexedNotNull { index, value ->
            if ((index == 0 || index == lastIndex) && value.isBlank())
                null
            else
                indentCutFunction(value)?.let(indentAddFunction) ?: value
        }
        .joinTo(StringBuilder(resultSizeEstimate), "\n")
        .toString()
}
```

```kotlin
private fun getIndentFunction(indent: String) = when {
    indent.isEmpty() -> { line: String -> line }
    else -> { line: String -
        indent + line }
}
```

```kotlin
/**
 * Defines names for Unicode symbols used in proper Typography.
 */
public object Typography {
    /** The character “” quotation mark */
    public const val quote: Char = '\u0022'
    /** The character $ dollar sign */
    public const val dollar: Char = '\u0024'
    /** The character & ampersand */
    public const val amp: Char = '\u0026'
    /** The character < less-than sign */
    public const val less: Char = '\u003C'
    /** The character > greater-than sign */
    public const val greater: Char = '\u003E'
    /** The non-breaking space character */
    public const val nbsp: Char = '\u00A0'
    /** The character × times */
    public const val times: Char = '\u00D7'
    /** The character ¢ cent */
    public const val cent: Char = '\u00A2'
    /** The character £ pound */
    public const val pound: Char = '\u00A3'
    /** The character § section */
    public const val section: Char = '\u00A7'
    /** The character © copyright */
    public const val copyright: Char = '\u00A9'
    /** The character « leftGuillemete */
    public const val leftGuillemete: Char = '\u00AB'
    /** The character » rightGuillemete */
    public const val rightGuillemete: Char = '\u00BB'
    /** The
character &\#xAE; \*\n    public const val registered: Char = '\u00AE'
    /** The character &\#xB0; */
    public const val public const val degree: Char = '\u00B0'
    /** The character &\#xB7; */
    public const val middleDot: Char = '\u00B7'
    /** The character &\#xBD; */
    public const val half: Char = '\u00BD'
    /** The character – */
    public const val ndash: Char = '\u2013'
    /** The character — */
    public const val mdash: Char = '\u2014'
    /** The character ‘ */
    public const val leftSingleQuote: Char = '\u2018'
    /** The character ' */
    public const val rightSingleQuote: Char = '\u2019'
    /** The character ' */
    public const val lowSingleQuote: Char = '\u201A'
    /** The character “ */
    public const val leftDoubleQuote: Char = '\u201C'
    /** The character ” */
    public const val rightDoubleQuote: Char = '\u201D'
    /** The character ” */
    public const val lowDoubleQuote: Char = '\u201E'
    /** The character † */
    public const val dagger: Char = '\u2020'
    /** The character ‡ */
    public const val doubleDagger: Char = '\u2021'
    /** The character • */
    public const val bullet: Char = '\u2022'
    /** The character … */
    public const val ellipsis: Char = '\u2026'
    /** The character ′ */
    public const val prime: Char = '\u2032'
    /** The character ″ */
    public const val doublePrime: Char = '\u2033'
    /** The character € */
    public const val euro: Char = '\u20AC'
    /** The character ™ */
    public const val tm: Char = '\u2122'
    /** The character ≈ */
    public const val almostEqual: Char = '\u2248'
    /** The character ≠ */
    public const val notEqual: Char = '\u2260'
    /** The character ≤ */
    public const val lessOrEqual: Char = '\u2264'
    /** The character ≥ */
    public const val greaterOrEqual: Char = '\u2265'
} /*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.text

/**
 * Represents a collection of captured groups in a single match of a regular expression.
 */
public interface MatchGroupCollection : Collection<MatchGroup?> {

    /** Returns a group with the specified [index].
     *
     * @return An instance of [MatchGroup] if the group with the specified [index] was matched or `null` otherwise.
     * Groups are indexed from 1 to the count of groups in the regular expression.
     * @sample samples.text.Regexp.matchDestructuringToGroupValues
     */
    public operator fun get(index: Int): MatchGroup?
}

/**
 * Extends [MatchGroupCollection] by introducing a way to get matched groups by name, when regex supports it.
 */
@SinceKotlin("1.1")
public interface MatchNamedGroupCollection : MatchGroupCollection {
    /**
     * Returns a named group with the specified [name].
     * @return An instance of [MatchGroup] if the group with the specified [name] was matched or `null` otherwise.
     * @throws IllegalArgumentException if there is no group with the specified [name] defined in the regex pattern.
     * @throws UnsupportedOperationException if getting named groups isn't supported on the current platform.
     */
    public operator fun get(name: String): MatchGroup?
}

/**
 * Represents the results from a single regular expression match.
 */
public interface MatchResult {
    /** The range of indices in the original string where match was captured. */
    public val range: IntRange
    /** The substring from the input string captured by this match. */
    public val value: String
    /** A collection of groups matched by the regular expression. */
    public val groups: MatchGroupCollection
    /** A list of matched indexed group values. */
    public val groupValues: List<String>

    @SinceKotlin("1.1")
    vital for understanding how to use regular expressions in Kotlin. The MatchResult interface provides a convenient way to access the results of a match, including the range of the match, the matched substring, and a collection of MatchGroup objects representing the captured groups. The MatchGroupCollection allows accessing groups by their index, while MatchNamedGroupCollection extends this functionality by allowing access to groups by name. This is particularly useful when dealing with complex regular expressions that define named groups. The MatchResult interface is a cornerstone of Kotlin's powerful regex capabilities, enabling developers to efficiently process and analyze text data.
/**
 * An instance of [MatchResult.Destructured] wrapper providing components for destructuring assignment
 * of group values.
 * component1 corresponds to the value of the first group, component2 of the second, and so on.
 * public val destructured: Destructured get() = Destructured(this)
 * Returns a new [MatchResult] with the results for the next match, starting at the position
 * at which the last match ended (at the character after the last
 * matched character).
 * public fun next(): MatchResult?
 * Provides components for destructuring assignment of group values.
 * [component1] corresponds to the value of the first group, [component2] of the second, and so on.
 * If the group in the regular expression is optional and there were no match captured by that group,
 * corresponding component value is an empty string.
 * *
 * @sample samples.text.Regexps.matchDestructuringToGroupValues
 */

public class Destructured internal constructor(public val match: MatchResult) {
    // ...
subsequent invocations. * If the clock is monotonic, it can change only from 'true' to 'false', namely, when the clock mark becomes behind the current point of the clock. *

/*
 * public fun hasNotPassedNow(): Boolean =
          elapsedNow().isNegative()?
          @ExperimentalTime
          @SinceKotlin("1.3")
          @kotlin.internal.InlineOnly
          @Deprecated("Subtracting one ClockMark from another is not a well defined operation because these clock marks could have been obtained from the different clocks.", level = DeprecationLevel.ERROR)
          @kotlin.internal.InlineOnly
          public inline operator fun ClockMark.minus(other: ClockMark): Duration = throw Error("Operation is disallowed.")
          @ExperimentalTime
          @SinceKotlin("1.3")
          @kotlin.internal.InlineOnly
          @Deprecated("Comparing one ClockMark to another is not a well defined operation because these clock marks could have been obtained from the different clocks.", level = DeprecationLevel.ERROR)
          @kotlin.internal.InlineOnly
          public inline operator fun ClockMark.compareTo(other: ClockMark): Int = throw Error("Operation is disallowed.")
          @ExperimentalTime
          @SinceKotlin("1.3")
          @kotlin.internal.InlineOnly
          @Deprecated("Abstract class used to implement clocks that return their readings as [Long] values in the specified [unit].
          * @property unit The unit in which this clock readings are expressed.
          * @SinceKotlin("1.3")
          @ExperimentalTime
          @public abstract class AbstractLongClock(protected val unit: DurationUnit) : Clock {
          /**
           * This protected method should be overridden to return the current reading of the clock expressed as a [Long] number in the unit specified by the [unit] property.
           * @protected abstract fun read(): Long
           * @SinceKotlin("1.3")
           @ExperimentalTime
           @public abstract class TestClock : AbstractLongClock(unit = DurationUnit.NANOSECONDS) {
          private var reading: Long = 0L
          override fun read(): Long = reading
          /**
           * Advances the current reading value of this clock by the specified [duration].
           * @protected fun markNow(): ClockMark =
           * @throws IllegalStateException

Open Source Used In DNAC Platform Antelope
when the reading value overflows as the result of this operation.

    public operator fun plusAssign(duration: Duration)
    { val delta = duration.toDouble(unit) val longDelta = delta.toDouble() reading = if
(longDelta != Long.MIN_VALUE && longDelta != Long.MAX_VALUE) { when delta fits in long, add
    it as long
    val newReading = reading + longDelta if (reading xor longDelta >= 0 && reading xor
newReading < 0) overflow(duration) newReading
    } else { when delta is greater than long, add it as double
    val newReading = reading + delta if (newReading > Long.MAX_VALUE ||
newReading < Long.MIN_VALUE) overflow(duration) newReading toLong() } }

private fun overflow(duration: Duration) { throw IllegalArgumentException("TestClock will overflow if its reading
$\{reading\} ns is advanced by $\{duration\}.") }

/*
 * Copyright 2010-2019 JetBrains
 * Use of this source code is governed by the Apache 2.0
license that can be found in the LICENSE.txt file.
 */

package kotlin.time

import kotlin.math.abs

@UseExperimental(ExperimentalTime::class)
private inline val storageUnit get() = DurationUnit.NANOSECONDS

/**
 * Represents the amount of time one instant of time is away from another
 * instant. A negative duration is possible in a situation when the second instant is earlier than the first one.
 * An infinite duration value [Duration.INFINITY] can be used to represent infinite timeouts.
 * To construct a duration use either the extension function [toDuration],
 * or the extension properties [hours], [minutes], [seconds],
 * and so on. * available on [Int], [Long], and [Double] numeric types.
 * To get the value of this duration
 * expressed in a particular [duration units] [DurationUnit] * use the functions [toInt], [toLong], and [toDouble] * or
 * the properties [inHours], [inMinutes], [inSeconds], [inNanoseconds], and so on.
 * An infinite duration value [Duration.INFINITY] can be used to represent infinite timeouts.
 * To construct a duration use either the extension function [toDuration],
 * or the extension properties [hours], [minutes], [seconds],
 * and so on. * available on [Int], [Long], and [Double] numeric types.
 * To get the value of this duration
 * expressed in a particular [duration units] [DurationUnit] * use the functions [toInt], [toLong], and [toDouble] * or
 * the properties [inHours], [inMinutes], [inSeconds], [inNanoseconds], and so on.
 */
@SinceKotlin("1.3")
@ExperimentalTime
@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_O
F_INLINE_CLASS")
public inline class Duration internal constructor(internal val value: Double) :
Comparable<Duration> {
    // arithmetic operators

    /** Returns the negative of this value. */
    public operator fun unaryMinus(): Duration = Duration(-value)

    /** Returns a duration whose value is the sum of this and [other] duration values. */
    public operator fun plus(other: Duration): Duration = Duration(value + other.value)

    /** Returns a duration whose value is the difference between this and [other] duration values. */
    public operator fun minus(other: Duration): Duration = Duration(value - other.value)

    /** Returns a number that is the ratio of this and [other] duration values. */
    public operator fun div(other: Duration): Duration = Duration(value / other.value)

    /** Returns a duration whose value is infinite. */
    public fun isInfinite(): Boolean = value.isInfinite()

    /** Returns true, if the duration value is finite. */
    public fun isFinite(): Boolean = !isInfinite()
fun isFinite(): Boolean = value.isFinite()// ** Returns the absolute value of this value. The returned value is always non-negative. */
public val absoluteValue: Duration get() = if (isNegative()) -this else this
override
fun compareTo(other: Duration): Int = this.value.compareTo(other.value)
// splitting to components
/**
 * Splits this duration into days, hours, minutes, seconds, and nanoseconds and executes the given [action] with these components.
 * The result of [action] is returned as the result of this function.
 * `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;
 * `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;
 * `hours` represents the whole number of hours in this duration, and its absolute value is less than 24;
 * `days` represents the whole number of days in this duration.
 * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
 * it is coerced into that range.
 */
public inline fun <T> toComponents(action: (days: Int, hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T =
    action(inDays.toLong(), hoursComponent, minutesComponent, secondsComponent, nanosecondsComponent)
/**
 * Splits this duration into hours, minutes, and nanoseconds and executes the given [action] with these components.
 * The result of [action] is returned as the result of this function.
 * `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;
 * `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;
 * `hours` represents the whole number of hours in this duration.
 * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
 * it is coerced into that range.
 */
public inline fun <T> toComponents(action: (hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T =
    action(inHours.toInt(), minutesComponent, secondsComponent, nanosecondsComponent)
/**
 * Splits this duration into minutes, seconds, and nanoseconds and executes the given [action] with these components.
 * The result of [action] is returned as the result of this function.
 * `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 1_000_000_000;
 * `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;
 * `hours` represents the whole number of hours in this duration.
 * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
 * it is coerced into that range.
 */
public inline fun <T> toComponents(action: (minutes: Int, seconds: Int, nanoseconds: Int) -> T): T =
    action(inMinutes.toInt(), secondsComponent, nanosecondsComponent)
/**
 * Splits this duration into seconds, and nanoseconds and executes the given [action] with these components.
 * The result of [action] is returned as the result of this function.
 * `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 1_000_000_000;
 * `minutes` represents the whole number of minutes in this duration.
 * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
 * it is coerced into that range.
 */
public inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Int) -> T): T =
    action(inSeconds.toLong(), nanosecondsComponent)
/**
 * Splits this duration into milliseconds, and nanoseconds and executes the given [action] with these components.
 * The result of [action] is returned as the result of this function.
 * `milliseconds` represents the whole number of milliseconds in this duration, and its absolute value is less than 1_000;
 * `seconds` represents the whole number of seconds in this duration.
 * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
 * it is coerced into that range.
 */
public inline fun <T> toComponents(action: (milliseconds: Long, seconds: Int) -> T): T =
    action(inMilliseconds.toLong(), secondsComponent)
/**
 * Splits this duration into microseconds, and nanoseconds and executes the given [action] with these components.
 * The result of [action] is returned as the result of this function.
 * `microseconds` represents the whole number of microseconds in this duration, and its absolute value is less than 1_000_000;
 * `seconds` represents the whole number of seconds in this duration.
 * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
 * it is coerced into that range.
 */
public inline fun <T> toComponents(action: (microseconds: Long, seconds: Int) -> T): T =
    action(inMicroseconds.toLong(), secondsComponent)
/**
 * Splits this duration into nanoseconds and executes the given [action] with these components.
 * The result of [action] is returned as the result of this function.
 * `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000;
 * `seconds` represents the whole number of seconds in this duration.
 * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
 * it is coerced into that range.
 */
public inline fun <T> toComponents(action: (nanoseconds: Long, seconds: Int) -> T): T =
    action(inNanoseconds.toLong(), secondsComponent)
public val days: Double get() = toDouble(DurationUnit.DAYS)  /** The value of this duration expressed as a [Double] number of days. */
public val hours: Double get() = toDouble(DurationUnit.HOURS)  /** The value of this duration expressed as a [Double] number of hours. */
public val minutes: Double get() = toDouble(DurationUnit.MINUTES)  /** The value of this duration expressed as a [Double] number of minutes. */
public val seconds: Double get() = toDouble(DurationUnit.SECONDS)  /** The value of this duration expressed as a [Double] number of milliseconds. */
public val nanoseconds: Double get() = toDouble(DurationUnit.NANOSECONDS)  // shortcuts

public fun toLongNanoseconds(): Long = toLong(DurationUnit.NANOSECONDS)  /** Returns the value of this duration expressed as a [Long] number of nanoseconds. */

public fun toLongMilliseconds(): Long = toLong(DurationUnit.MILLISECONDS)  /** Returns the value of this duration expressed as a [Long] number of milliseconds. */

override fun toString(): String = when {
    isInfinite() -> value.toString()
    value == 0.0 -> "0s"
    else -> {
        val absNs = absoluteValue.inNanoseconds
        var scientific = false
        var maxDecimals = 0
        val unit = when {
            absNs < 1e-6 -> DurationUnit.SECONDS.also { scientific = true }
            absNs < 1 -> DurationUnit.NANOSECONDS
            absNs < 1e3 -> DurationUnit.NANOSECONDS
            absNs < 1e6 -> DurationUnit.MICROSECONDS
            absNs < 1000e9 -> DurationUnit.SECONDS
            absNs < 60_000e9 -> DurationUnit.MINUTES
            absNs < 3600_000e9 -> DurationUnit.HOURS
            absNs < 86400e9 * 1e7 -> DurationUnit.DAYS
            else -> DurationUnit.DAYS.also { scientific = true }
        }
        val value = toDouble(unit)
        when {
            scientific -> formatScientific(value)
            maxDecimals > 0 -> formatUpToDecimals(value, maxDecimals)
            else -> formatToExactDecimals(value, precision(abs(value))) + unit.shortName()
        }
    }
}

private fun precision(value: Double): Int = when {
    value < 1 -> 3
    value < 10 -> 2
    value < 100 -> 1
    else -> 0
}

public fun toString(unit: DurationUnit, decimals: Int = 0): String {
    require(decimals >= 0) { "decimals must be not negative, but was $decimals" }
    if (isInfinite()) return value.toString()
    val absNs = absoluteValue.inNanoseconds
    val value = Double(absNs)
    return when {
        absNs < 1e6 -> formatToExactDecimals(value, decimals.coerceAtMost(12)) + unit.shortName()
        else -> formatScientific(value)
    }
}

public fun toStringDefault(): String = when {
    isInfinite() -> "Infinity"
    value == 0.0 -> "0s"
    else -> {
        val absNs = absoluteValue.inNanoseconds
        var scientific = false
        var maxDecimals = 0
        val unit = when {
            absNs < 1e-6 -> DurationUnit.SECONDS.also { scientific = true }
            absNs < 1 -> DurationUnit.NANOSECONDS
            absNs < 1e3 -> DurationUnit.NANOSECONDS
            absNs < 1e6 -> DurationUnit.MICROSECONDS
            absNs < 1000e9 -> DurationUnit.SECONDS
            absNs < 60_000e9 -> DurationUnit.MINUTES
            absNs < 3600_000e9 -> DurationUnit.HOURS
            absNs < 86400e9 * 1e7 -> DurationUnit.DAYS
            else -> DurationUnit.DAYS.also { scientific = true }
        }
        val value = toDouble(unit)
        when {
            scientific -> formatScientific(value)
            maxDecimals > 0 -> formatUpToDecimals(value, maxDecimals)
            else -> formatToExactDecimals(value, precision(abs(value))) + unit.shortName()
        }
    }
}

fun Duration.toStringDefault(): String = when {
    isInfinite() -> "Infinity"
    value == 0.0 -> "0s"
    else -> {
        val absNs = absoluteValue.inNanoseconds
        var scientific = false
        var maxDecimals = 0
        val unit = when {
            absNs < 1e-6 -> DurationUnit.SECONDS.also { scientific = true }
            absNs < 1 -> DurationUnit.NANOSECONDS
            absNs < 1e3 -> DurationUnit.NANOSECONDS
            absNs < 1e6 -> DurationUnit.MICROSECONDS
            absNs < 1000e9 -> DurationUnit.SECONDS
            absNs < 60_000e9 -> DurationUnit.MINUTES
            absNs < 3600_000e9 -> DurationUnit.HOURS
            absNs < 86400e9 * 1e7 -> DurationUnit.DAYS
            else -> DurationUnit.DAYS.also { scientific = true }
        }
        val value = toDouble(unit)
        when {
            scientific -> formatScientific(value)
            maxDecimals > 0 -> formatUpToDecimals(value, maxDecimals)
            else -> formatToExactDecimals(value, precision(abs(value))) + unit.shortName()
        }
    }
}

val abs value

of this duration (see [toComponents])\n* and \(^{\prime}\) is a fractional part of second. Depending on the roundness of the value the fractional part can be formatted with either\n* 0, 3, 6, or 9 decimal digits.\n* If the hours component absolute value of this duration is greater than [Int.MAX_VALUE], it is replaced with [Int.MAX_VALUE].\n* Negative durations are indicated with the sign \(^{-}\) in the beginning of the returned string, for example, \"^{-PT3M30S}\".\n* @sample samples.time.Durations.toString.\n* public fun toString(): String = buildString {\n  if (isNegative()) append("^{-}\n  append("PT")\n  absoluteValue.toComponents { hours, minutes, seconds, nanoseconds ->\n    if (hasHours) {\n      append(hours).append('H')\n    }\n    if (hasMinutes) {\n      append(minutes).append('M')\n    }\n    if (hasSeconds || (!hasHours && !hasMinutes)) {\n      append(seconds)\n      if (nanoseconds != 0) {\n        append('.')\n        val nss = nanoseconds.toString().padStart(9, '0')\n        when {\n          nanoseconds % 1_000_000 == 0 -> append(nss, 0, 3)\n          nanoseconds % 1_000 == 0 -> append(nss, 0, 6)\n          else -> append(nss)\n        }\n      } else -> append('S')\n    }\n  }\n}\n* @sample samples.time.Durations.toIsoString.\n* public fun toIsoString(): String = buildString {\n  if (isNegative()) append('-')\n  append("PT")\n  absoluteValue.toComponents { hours, minutes, seconds, nanoseconds, nanoseconds ->\n    if (hasHours) {\n      append(hours).append('H')\n    }\n    if (hasMinutes) {\n      append(minutes).append('M')\n    }\n    if (hasSeconds || (!hasHours && !hasMinutes)) {\n      append(seconds)\n      if (nanoseconds != 0) {\n        append('.')\n        val nss = nanoseconds.toString().padStart(9, '0')\n        when {\n          nanoseconds % 1_000_000 == 0 -> append(nss, 0, 3)\n          nanoseconds % 1_000 == 0 -> append(nss, 0, 6)\n          else -> append(nss)\n        }\n      } else -> append('S')\n    }\n  }\n}\n* // constructing from number of units\n* // extension functions\n* // Returns a [Duration] equal to this [Int] number of the specified [unit].\n* @sample samples.time.Durations.toInt\n* fun Int.toInt(unit: DurationUnit): Duration = toDouble().toInt(unit)\n* @sample samples.time.Durations.toLong\n* public fun toLong(unit: DurationUnit): Long = toDouble().toLong(unit)\n* @sample samples.time.Durations.toDouble\n* fun Double.toDouble(unit: DurationUnit): Double = Duration(convertDurationUnit(this, unit, storageUnit))\n* // constructing from number of units\n* // extension properties\n* // Returns a [Duration] equal to this [Int] number of nanoseconds.\n* @sample samples.time.Durations.nanoseconds\n* val Int.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n* // Returns a [Duration] equal to this [Long] number of nanoseconds.\n* @sample samples.time.Durations.nanoSeconds\n* val Long.nanoSeconds get() = toDuration(DurationUnit.NANOSECONDS)\n* // Returns a [Duration] equal to this [Double] number of nanoseconds.\n* @sample samples.time.Durations.nanoSeconds\n* val Double.nanoSeconds get() = toDuration(DurationUnit.NANOSECONDS)\n* // Returns a [Duration] equal to this [Int] number of microseconds.\n* @sample samples.time.Durations.microseconds\n* val Int.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n* // Returns a [Duration] equal to this [Long] number of microseconds.\n* @sample samples.time.Durations.microseconds\n* val Long.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n* // Returns a [Duration] equal to this [Double] number of microseconds.\n* @sample samples.time.Durations.microseconds\n* val Double.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n* // Returns a [Duration] equal to this [Int] number of milliseconds.\n* @sample samples.time.Durations.milliseconds\n* val Int.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n* // Returns a [Duration] equal to this [Long] number of milliseconds.\n* @sample samples.time.Durations.milliseconds\n* val Long.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n* // Returns a [Duration] equal to this [Double] number of milliseconds.\n* @sample samples.time.Durations.milliseconds\n* val Double.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n* // Returns a [Duration] equal to this [Int] number of seconds.\n* @sample samples.time.Durations.seconds\n* val Int.seconds get() = toDuration(DurationUnit.SECONDS)\n* // Returns a [Duration] equal to this [Long] number of seconds.\n* @sample samples.time.Durations.seconds\n* val Long.seconds get() = toDuration(DurationUnit.SECONDS)\n* // Returns a [Duration] equal to this [Double] number of seconds.\n* @sample samples.time.Durations.seconds\n* val Double.seconds get() = toDuration(DurationUnit.SECONDS)\n* // Returns a [Duration] equal to this [Int] number of minutes.\n* @sample samples.time.Durations.minutes\n* val Int.minutes get() = toDuration(DurationUnit.MINUTES)\n* // Returns a [Duration] equal to this [Long] number of minutes.\n* @sample samples.time.Durations.minutes\n* val Long.minutes get() = toDuration(DurationUnit.MINUTES)
@SinceKotlin("1.3")
@ExperimentalTime
public val Long.minutes get() = toDuration(DurationUnit.MINUTES)

/** Returns a [Duration] equal to this [Double] number of minutes. */
@SinceKotlin("1.3")
@ExperimentalTime
public val Double.minutes get() = toDuration(DurationUnit.MINUTES)

/** Returns a [Duration] equal to this [Int] number of hours. */
@SinceKotlin("1.3")
@ExperimentalTime
public val Int.hours get() = toDuration(DurationUnit.HOURS)

/** Returns a [Duration] equal to this [Long] number of hours. */
@SinceKotlin("1.3")
@ExperimentalTime
public val Long.hours get() = toDuration(DurationUnit.HOURS)

/** Returns a [Duration] equal to this [Double] number of hours. */
@SinceKotlin("1.3")
@ExperimentalTime
public val Double.hours get() = toDuration(DurationUnit.DAYS)

/** Returns a [Duration] equal to this [Int] number of days. */
@SinceKotlin("1.3")
@ExperimentalTime
public val Int.days get() = toDuration(DurationUnit.DAYS)

/** Returns a [Duration] equal to this [Long] number of days. */
@SinceKotlin("1.3")
@ExperimentalTime
public val Long.days get() = toDuration(DurationUnit.DAYS)

/** Returns a duration whose value is the specified [duration] value multiplied by this number. */
@SinceKotlin("1.3")
@ExperimentalTime
@kotlin.internal.InlineOnly
public inline operator fun Int.times(duration: Duration): Duration = duration * this

/** Returns a duration whose value is the specified [duration] value multiplied by this number. */
@SinceKotlin("1.3")
@ExperimentalTime
@kotlin.internal.InlineOnly
public inline operator fun Double.times(duration: Duration): Duration = duration * this

internal expect fun formatToExactDecimals(value: Double, decimals: Int): String
internal expect fun formatUpToDecimals(value: Double): String
internal expect fun formatScientific(value: Double): String

import kotlin.jvm.JvmMultifileClass
import kotlin.jvm.JvmName

package kotlin.time

/**
 * The list of possible time measurement units, in which a duration can be expressed.
 * The smallest time unit is [NANOSECONDS] and the largest is [DAYS], which corresponds to exactly 24 [HOURS].
 * Time unit representing one nanosecond, which is 1/1000 of a microsecond.
 * Time unit representing one microsecond, which is 1/1000 of a millisecond.
 * Time unit representing one millisecond, which is 1/1000 of a second.
 * Time unit representing one second.
 * Time unit representing one minute.
 * Time unit representing one hour.
 * Time unit representing one day, which is always equal to 24 hours.
 * Converts the given time duration [value] expressed in the specified [sourceUnit] into the specified [targetUnit].
 */
@SinceKotlin("1.3")
@ExperimentalTime
internal fun DurationUnit.shortName(): String = when (this) {
    DurationUnit.NANOSECONDS -> "ns"
    DurationUnit.MICROSECONDS -> "us"
    DurationUnit.MILLISECONDS -> "ms"
    DurationUnit.SECONDS -> "s"
    DurationUnit.MINUTES -> "m"
    DurationUnit.HOURS -> "h"
    DurationUnit.DAYS -> "d"
}
annotated with `@ExperimentalTime` must be accepted either by
* annotating that usage with the
[UseExperimental] annotation, e.g. `@UseExperimental(ExperimentalTime::class)`,
* or by using the compiler argument `-Xuse-experimental=kotlin.time.ExperimentalTime`
*
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*
* The elapsed time is measured with [MonoClock].
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
* The elapsed time is measured with the specified `this` [Clock] instance.
*
override fun toString(): String = "$major.$minor.$patch"

    override fun equals(other: Any?): Boolean {
        if (this === other) return true
        val otherVersion = (other as? KotlinVersion) ?: return false
        return this.version == otherVersion.version
    }

    override fun compareTo(other: KotlinVersion): Int = version - other.version

    /**
     * Returns `true` if this version is not less than the version specified
     * with the provided [major] and [minor] components.
     */
    public fun isAtLeast(major: Int, minor: Int): Boolean = // this.version >= versionOf(major, minor, 0)
        this.major > major || (this.major == major &&
                this.minor >= minor)

    /**
     * Returns `true` if this version is not less than the version specified
     * with the provided [major], [minor] and [patch] components.
     */
    public fun isAtLeast(major: Int, minor: Int, patch: Int): Boolean = // this.version >= versionOf(major, minor,
        this.major > major || (this.major == major &&
                (this.minor > minor || this.minor == minor &&
                        this.patch >= patch))

    companion object {
        /**
         * Maximum value a version component can have, a constant value 255.
         */
        // NOTE: Must be placed before CURRENT because its
        public const val MAX_COMPONENT_VALUE = 255

        /**
         * Returns the current version of the Kotlin standard library.
         */
        @kotlin.jvm.JvmField
        public val CURRENT: KotlinVersion = KotlinVersion(1, 3, 61) // value is written here automatically during build
    }
}

@file:kotlin.jvm.JvmName("LateinitKt")
@file:Suppress("unused")

package kotlin

import kotlin.internal.InlineOnly
import kotlin.internal.AccessibleLateinitPropertyLiteral
import kotlin.reflect.KProperty0

/**
 * Returns `true` if this lateinit property has been assigned a value, and `false`
 * otherwise.
 * Cannot be used in an inline function, to avoid binary compatibility issues.
 */
@SinceKotlin("1.2")
@InlineOnly
inline val @receiver:AccessibleLateinitPropertyLiteral
        KProperty0<*>.isInitialized: Boolean
    get() = throw NotImplementedError("Implementation is

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
 * Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in
 * the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmName("LazyKt")
@file:kotlin.jvm.JvmMultifileClass

package kotlin

import kotlin.reflect.KProperty

/**
 * Represents a value with lazy initialization.
 * To create an instance of
 * [Lazy] use the [lazy] function.
 */
public interface Lazy<out T> {
    /**
     * Gets the lazily initialized value of
     * the current Lazy instance.
     * Once the value was initialized it must not change during the rest of lifetime of this
     * Lazy instance.
     */
    public val value: T

    /**
     * Returns `true` if a value for this Lazy instance has been
     * already initialized, and `false`
     * otherwise.
     * Once this function has returned `true` it stays `true` for the rest of
     * lifetime of this Lazy instance.
     */
    public fun isInitialized(): Boolean = // this.isInitialized()

    /**
     * Specifies how a [Lazy] instance synchronizes initialization
     * among multiple threads.
     */
    public enum class LazyThreadSafetyMode {

        /**
         * Locks are used to ensure
         * that only a single thread can initialize the [Lazy] instance.
         */
        SYNCHRONIZED,

        /**
         * An extension to delegate a read-only property of type [T] to an instance of
         * [Lazy]. This extension allows to use instances of Lazy for property delegation.
         */
        NONE,

        /**
         * This mode should not be used unless the [Lazy] instance
         * is guaranteed never to be initialized from more than one thread
         */
        PUBLICATION,

        /**
         * No locks are used to synchronize an access to the [Lazy] instance value; if
         * the instance is accessed from multiple threads, its behavior is undefined.
         */
        UNINITIALIZED_VALUE,

        /**
         * Specifies how a [Lazy] instance synchronizes initialization
         * among multiple threads.
         */
        UNINITIALIZED_VALUE
    }

    @receiver KProperty0<*>.getValue(thisRef: Any?, property: KProperty): T

    @receiver KProperty0<*>.getDelegate(thisRef: Any?, property: KProperty): LazyThreadSafetyMode

    companion object {

        @receiver KProperty0<*>::getDelegate(thisRef: Any?, property: KProperty): LazyThreadSafetyMode
        // Implementation is

    }
}

// internal to be called from lazy in JS
internal class UnsafeLazyImpl<out T>(initializer: () -> T) : Lazy<T> {
    private var initializer: (() -> T)? = initializer
    private var _value: Any? = UNINITIALIZED_VALUE

    override fun isInitialized(): Boolean = // this.isInitialized()

    override fun getValue(thisRef: Any?, property: KProperty): T
        get() = throw NotInitializedException("Implementation is

    // Implementation is

    companion object {

        @receiver KProperty0<*>::getDelegate(thisRef: Any?, property: KProperty): LazyThreadSafetyMode
        // Implementation is

    }
}

@file:kotlin.jvm.JvmName("LateinitKt")
@file:Suppress("unused")

package kotlin

import kotlin.internal.InlineOnly
import kotlin.internal.AccessibleLateinitPropertyLiteral
import kotlin.reflect.KProperty0

/**
 * Returns `true` if this lateinit property has been assigned a value, and `false`
 * otherwise.
 * Cannot be used in an inline function, to avoid binary compatibility issues.
 */
@SinceKotlin("1.2")
@InlineOnly
inline val @receiver:AccessibleLateinitPropertyLiteral
        KProperty0<*>.isInitialized: Boolean
    get() = throw NotImplementedError("Implementation is

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
 * Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in
 * the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmName("LazyKt")
@file:kotlin.jvm.JvmMultifileClass

package kotlin

import kotlin.reflect.KProperty

/**
 * Represents a value with lazy initialization.
 * To create an instance of
 * [Lazy] use the [lazy] function.
 */
public interface Lazy<out T> {
    /**
     * Gets the lazily initialized value of
     * the current Lazy instance.
     * Once the value was initialized it must not change during the rest of lifetime of this
     * Lazy instance.
     */
    public val value: T

    /**
     * Returns `true` if a value for this Lazy instance has been
     * already initialized, and `false`
     * otherwise.
     * Once this function has returned `true` it stays `true` for the rest of
     * lifetime of this Lazy instance.
     */
    public fun isInitialized(): Boolean = // this.isInitialized()

    /**
     * Specifies how a [Lazy] instance synchronizes initialization
     * among multiple threads.
     */
    public enum class LazyThreadSafetyMode {

        /**
         * Locks are used to ensure
         * that only a single thread can initialize the [Lazy] instance.
         */
        SYNCHRONIZED,

        /**
         * An extension to delegate a read-only property of type [T] to an instance of
         * [Lazy]. This extension allows to use instances of Lazy for property delegation.
         */
        NONE,

        /**
         * This mode should not be used unless the [Lazy] instance
         * is guaranteed never to be initialized from more than one thread
         */
        PUBLICATION,

        /**
         * No locks are used to synchronize an access to the [Lazy] instance value; if
         * the instance is accessed from multiple threads, its behavior is undefined.
         */
        UNINITIALIZED_VALUE,

        /**
         * Specifies how a [Lazy] instance synchronizes initialization
         * among multiple threads.
         */
        UNINITIALIZED_VALUE
    }

    @receiver KProperty0<*>.getValue(thisRef: Any?, property: KProperty): T

    @receiver KProperty0<*>.getDelegate(thisRef: Any?, property: KProperty): LazyThreadSafetyMode

    companion object {

        @receiver KProperty0<*>::getDelegate(thisRef: Any?, property: KProperty): LazyThreadSafetyMode
        // Implementation is

    }
}

// internal to be called from lazy in JS
internal class UnsafeLazyImpl<out T>(initializer: () -> T) : Lazy<T> {
    private var initializer: (() -> T)? = initializer
    private var _value: Any? = UNINITIALIZED_VALUE

    override fun isInitialized(): Boolean = // this.isInitialized()

    override fun getValue(thisRef: Any?, property: KProperty): T
        get() = throw NotInitializedException("Implementation is

    // Implementation is

    companion object {

        @receiver KProperty0<*>::getDelegate(thisRef: Any?, property: KProperty): LazyThreadSafetyMode
        // Implementation is

    }
}
@Suppress("UNCHECKED_CAST")
            return _value as T
        }

    override fun isInitialized():
        = _value !== UNINITIALIZED_VALUE

    override fun toString(): String = if (isInitialized())
        value.toString() else "Lazy value not initialized yet.";

    private fun writeReplace(): Any =
        InitializedLazyImpl(value)
}

internal class InitializedLazyImpl<out T>(override val value: T) : Lazy<T>,
    Serializable {

    override fun isInitialized(): Boolean = true

    override fun toString(): String =
        value.toString()

}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("NumbersKt")
package kotlin

/**
 * Counts the number of set bits in the binary representation of this [Int] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Int.countOneBits(): Int

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Int.countLeadingZeroBits(): Int

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Int] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Int.countTrailingZeroBits(): Int

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [Int] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Int.takeHighestOneBit(): Int

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [Int] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Int.takeLowestOneBit(): Int

/**
 * Rotates the binary representation of this [Int] number left by the specified [bitCount] number of bits.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Int.rotateLeft(bitCount: Int): Int

/**
 * Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Int.rotateRight(bitCount: Int): Int

/**
 * Counts the number of set bits in the binary representation of this [Long] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Long.countOneBits(): Int

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Long.countLeadingZeroBits(): Int

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Long.countTrailingZeroBits(): Int

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [Long] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Long.takeHighestOneBit(): Long

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [Long] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Long.takeLowestOneBit(): Long

/**
 * Rotates the binary representation of this [Long] number left by the specified [bitCount] number of bits.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Long.rotateLeft(bitCount: Int): Long

/**
 * Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun Long.rotateRight(bitCount: Int): Long
multiple of \([\text{Long.SIZE_BITS}]\) (64) returns the same number, or more generally:
\[ \text{number.rotateLeft(n)} == \text{number.rotateLeft(n \% 64)} \]
\[ *n\text{number.rotateRight(n)} == \text{number.rotateRight(n \% 64)} \]
\[ \text{number.rotateLeft(n)} == \text{number.rotateRight(n)} \]

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun
Long.rotateLeft(bitCount: Int): Long

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun
Long.rotateRight(bitCount: Int): Long
```

```kotlin
/*
 * Rotates the binary representation of this [Long] number right by the
 * specified [bitCount] number of bits.
 * The least significant bits pushed out from the right side reenter the number
 * as the most significant bits on the left side.
 * `number.rotateRight(n) == number.rotateLeft(n \% 64)`
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun
Long.rotateRight(bitCount: Int): Long
```

```
```
having a single bit set in the position of the most significant set bit of this [Short] number.\n* or zero, if this number is zero.\n*/
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline fun

```
Short.takeHighestOneBit(): Short = (toInt() and 0xFFFF).takeHighestOneBit().toShort()\n
```
public override fun toString(): String = \"(\$first, \$second, \$third)\"}

/**
 * Converts this triple into a list.
 * @sample samples.misc.Tuples.tripleToList
 */
public fun <T> Triple<T, T, T>.toList(): List<T> =
    listOf(first, second, third)

/**
 * A range of values of type `UInt`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public class UIntRange(start: UInt, endInclusive: UInt) : UIntProgression(start, endInclusive, 1), ClosedRange<UInt> {
    override val start: UInt get() = first
    override val endInclusive: UInt get() = last

    override fun contains(value: UInt): Boolean = first <= value && value <= last

    override fun isEmpty(): Boolean = if (step > 0) first > last else first < last

    override fun equals(other: Any?): Boolean =
        other is UIntRange && (isEmpty() && other.isEmpty() ||
                first == other.first && last == other.last)

    override fun hashCode(): Int =
        if (isEmpty()) -1 else (31 * first.toInt() + last.toInt())

    override fun toString(): String = "$first..$last"

    companion object {
        /**
         * An empty range of values of type UInt.
         */
        public val EMPTY: UIntRange = UIntRange(UInt.MAX_VALUE, UInt.MIN_VALUE)
    }
}

/**
 * A progression of values of type `UInt`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public open class UIntProgression
internal constructor(
    start: UInt,
    endInclusive: UInt,
    step: Int
) : Iterable<UInt> {
    init {
        if (step == 0.toInt()) throw kotlin.IllegalArgumentException("Step must be non-zero.")
        if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to avoid overflow on negation.")
    }

    /**
     * The first element in the progression.
     */
    public val first: UInt = start

    /**
     * The last element in the progression.
     */
    public val last: UInt = getProgressionLastElement(start, endInclusive, step)

    /**
     * The step of the progression.
     */
    public val step: Int = step

    override fun iterator(): UIntIterator = UIntProgressionIterator(first, last, step)

    /**
     * Checks if the progression is empty.
     */
    public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last

    override fun equals(other: Any?): Boolean =
        other is UIntProgression && (isEmpty() && other.isEmpty() ||
                first == other.first && last == other.last && step == other.step)

    override fun hashCode(): Int =
        if (isEmpty()) -1 else (31 * (31 * first.toInt() + last.toInt()) + step.toInt())

    override fun toString(): String = if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"

    companion object {
        /**
         * Creates UIntProgression within the specified bounds of a closed range.
         */
        public fun fromClosedRange(rangeStart: UInt, rangeEnd: UInt, step: Int): UIntProgression = UIntProgression(rangeStart, rangeEnd, step)
    }
}

/**
 * An iterator over a progression of values of type `UInt`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UIntIterator : Iterator<UInt> {
    override final fun next() = nextUByte()

    /**
     * Returns the next value in the sequence without boxing.
     */
    public abstract fun nextUByte(): UInt
}

/**
 * An iterator over a sequence of values of type `UByte`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UByteIterator : Iterator<UByte> {
    override final fun next() = nextUByte()

    /**
     * Returns the next value in the sequence without boxing.
     */
    public abstract fun nextUByte(): UByte
}

/**
 * An iterator over a sequence of values of type `UShort`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UShortIterator : Iterator<UShort> {
    override final fun next() = nextUShort()

    /**
     * Returns the next value in the sequence without boxing.
     */
    public abstract fun nextUShort(): UShort
}
public abstract fun nextUShort(): UShort

/** An iterator over a sequence of values of type `UInt`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UIntIterator : Iterator<UInt> {
    override final fun next() = nextUInt()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextUInt(): UInt
}

/** An iterator over a sequence of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class ULongIterator : Iterator<ULong> {
    override final fun next() = nextULong()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextULong(): ULong
}

/** A range of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public class ULongRange(start: ULong, endInclusive: ULong) : ULongProgression(start, endInclusive, 1), ClosedRange<ULong> {
    override val start: ULong get() = first
    override val endInclusive: ULong get() = last

    override fun contains(value: ULong): Boolean = first <= value && value <= last

    override fun isEmpty(): Boolean = first > last

    override fun equals(other: Any?): Boolean =
        other is ULongRange && (isEmpty() && other.isEmpty() ||
                first == other.first && last == other.last)

    override fun hashCode(): Int =
        if (isEmpty()) -1 else (31 * (first xor (first shr 32)).toInt() +
        last xor (last shr 32)).toInt()

    override fun toString(): String = "$first..$last"

    companion object {
        /** An empty range of values of type ULong. */
        public val EMPTY: ULongRange =
            ULongRange(ULong.MAX_VALUE, ULong.MIN_VALUE)
    }
}

/** A progression of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public open class ULongProgression
    internal constructor(
            start: ULong,
            endInclusive: ULong,
            step: Long
    ) : Iterable<ULong> {
    init {
        if (step == 0.toLong()) throw kotlin.IllegalArgumentException("Step must be non-zero.")
        if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Long.MIN_VALUE to avoid overflow on negation."
        )
    }

    /** The first element in the progression. */
    public val first: ULong = start

    /** The last element in the progression. */
    public val last: ULong = getProgressionLastElement(ULongRange(ULong.MAX_VALUE, ULong.MIN_VALUE), 0)

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean =
        if (step > 0) first > last else first < last

    override fun equals(other: Any?): Boolean =
        other is ULongProgression &&
        (isEmpty() && other.isEmpty() ||
                first == other.first && last ==
        other.last && step == other.step)

    override fun hashCode(): Int =
        if (isEmpty()) -1 else (31 * (first xor (first shr 32)).toInt() +
        last xor (last shr 32)).toInt() + step xor (step ushr 32))

    override fun toString(): String = if (step > 0) "$first..$last step $step" else "$first downTo $last step-${step}

    companion object {
        /** Creates ULongProgression within the specified bounds of a closed range. */
        public fun fromClosedRange(rangeStart: ULong, rangeEnd: ULong, step: Long): ULongProgression =
            ULongProgression(rangeStart, rangeEnd, step)
    }
}
Returns the smaller of two values.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun min(a: UInt, b: UInt): UInt {
    return minOf(a, b)
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun min(a: ULong, b: ULong): ULong {
    return minOf(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun max(a: UInt, b: UInt): UInt {
    return maxOf(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun max(a: ULong, b: ULong): ULong {
    return maxOf(a, b)
}
```

@file:kotlin.jvm.JvmName("UNumbersKt")

package kotlin

/**
 * Counts the number of set bits in the binary representation of this [UInt] number.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.countOneBits(): Int = toInt().countOneBits()

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UInt] number.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.countLeadingZeroBits(): Int = toInt().countLeadingZeroBits()

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UInt] number.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.countTrailingZeroBits(): Int = toInt().countTrailingZeroBits()

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [UInt] number.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.takeHighestOneBit(): UInt = toInt().takeHighestOneBit().toUInt()

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [UInt] number.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.takeLowestOneBit(): UInt = toInt().takeLowestOneBit().toUInt()

/**
 * Rotates the binary representation of this [UInt] number left by the specified [bitCount] number of bits.
 * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.rotateLeft(bitCount: Int): UInt = toInt().rotateLeft(bitCount).toUInt()

/**
 * Rotates the binary representation of this [UInt] number right by the specified [bitCount] number of bits.
 * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.rotateRight(bitCount: Int): UInt = toInt().rotateRight(bitCount).toUInt()

/**
 * Counts the number of set bits in the binary representation of this [ULong] number.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULong.countOneBits(): Int = toLong().countOneBits()

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [ULong] number.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULong.countLeadingZeroBits(): Int = toLong().countLeadingZeroBits()

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [ULong] number.

 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULong.countTrailingZeroBits(): Int = toLong().countTrailingZeroBits()
number of consecutive least significant bits that are zero in the binary representation of this [ULong] number.

```kotlin
*\n@SinceKotlin("1.3.1")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl\nypublic inline fun ULong.countTrailingZeroBits(): Int = toLong().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [ULong] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.3.1")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl\nypublic inline fun ULong.takeHighestOneBit(): ULong = toLong().takeHighestOneBit().toULong()\n\n/**\n * Rotates the binary representation of this [ULong] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n *\n * number.rotateLeft(-n) == number.rotateRight(n)\n * Rotating by a multiple of [ULong.SIZE_BITS] (64) returns the same number, or more generally\n *\n * number.rotateLeft(n % 64)\n
*\n@SinceKotlin("1.3.1")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl\nypublic inline fun ULong.rotateLeft(bitCount: Int): ULong = toLong().rotateLeft(bitCount).toULong()\n\n/**\n * Rotates the binary representation of this [ULong] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the right side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n *\n * number.rotateRight(-n) == number.rotateLeft(n)\n * Rotating by a multiple of [ULong.SIZE_BITS] (64) returns the same number, or more generally\n *\n * number.rotateRight(n % 64)\n
*\n@SinceKotlin("1.3.1")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl\nypublic inline fun UByte.countOneBits(): Int = toUInt().countOneBits()\n\n/**\n * Counts the number of set bits in the binary representation of this [UByte] number.\n
*\n@SinceKotlin("1.3.1")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl\nypublic inline fun UByte.countLeadingZeroBits(): Int = toByte().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UByte] number.\n
*\n@SinceKotlin("1.3.1")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl\nypublic inline fun UByte.countTrailingZeroBits(): Int = toByte().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [UByte] number,\n * or zero, if this number is zero.\n
*\n@SinceKotlin("1.3.1")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl\nypublic inline fun UByte.takeHighestOneBit(): UByte = toInt().takeHighestOneBit().toUByte()\n\n/**\n * Rotates the binary representation of this [UByte] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n *\n * number.rotateLeft(-n) == number.rotateRight(n)\n * Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally\n *\n * number.rotateLeft(n % 8)\n
Open Source Used In DNAC Platform Antelope 4193
```
public inline fun UByte.rotateLeft(bitCount: Int): UByte = toByte().rotateLeft(bitCount).toUByte()

/**
 * Rotates the number right by a negative bit count is the same as rotating it left by the negated bit count:
 * `number.rotateRight(-n) == number.rotateLeft(n)`
 * Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally
 * `number.rotateRight(n) == number.rotateRight(n % 8)`
 */

public inline fun UByte.rotateRight(bitCount: Int): UByte = toByte().rotateRight(bitCount).toUByte()

/**
 * Counts the number of set bits in the binary representation of this [UByte] number.
 *
 */

public inline fun UShort.countOneBits(): Int = toUInt().countOneBits()

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [UShort] number.
 *
 */

public inline fun UShort.countLeadingZeroBits(): Int = toShort().countLeadingZeroBits()

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UShort] number.
 *
 */

public inline fun UShort.countTrailingZeroBits(): Int = toShort().countTrailingZeroBits()

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [UShort] number, or zero, if this number is zero.
 */

public inline fun UShort.takeHighestOneBit(): UShort = toInt().takeHighestOneBit().toUShort()

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [UShort] number, or zero, if this number is zero.
 */

public inline fun UShort.takeLowestOneBit(): UShort = toInt().takeLowestOneBit().toUShort()

/**
 * Rotates the binary representation of this [UShort] number left by the specified [bitCount] number of bits.
 * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.
 * `number.rotateLeft(-n) == number.rotateRight(n)`
 * Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally
 * `number.rotateLeft(n) == number.rotateLeft(n % 16)`
 */

public inline fun UShort.rotateLeft(bitCount: Int): UShort = toShort().rotateLeft(bitCount).toUShort()

/**
 * Rotates the binary representation of this [UShort] number right by the specified [bitCount] number of bits.
 * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.
 * `number.rotateRight(-n) == number.rotateLeft(n)`
 * Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally
 * `number.rotateRight(n) == number.rotateRight(n % 16)`
 */

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

@file:UseExperimental(ExperimentalUnsignedTypes::class)
package kotlin.internal

internal fun differenceModulo(a: UInt, b: UInt, c: UInt): UInt {
    val ac = a % c
    val bc = b % c
    return if (ac >= bc) ac - bc else ac - bc + c
}

internal fun differenceModulo(a: ULong, b: ULong, c: ULong): ULong {
    val ac = a % c
    val bc = b % c
    return if (ac >= bc) ac - bc else ac - bc + c
}

/**
 * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range
 * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative [step].
 * No validation on passed parameters is performed. The given parameters should satisfy the condition:
 */
`step > 0` and `start <= end`, or `step < 0` and `start >= end`.

* @param start first element of the progression
* @param end ending bound for the progression
* @param step step increment, or difference of successive elements in the progression
* @return the final element of the progression
* @suppress

@PublishedApi
@SinceKotlin("1.3")
internal fun getProgressionLastElement(start: UInt, end: UInt, step: Int): UInt = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toUInt())
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toUInt())
    else -> throw kotlin.IllegalArgumentException("Step is zero.")
}

/**
 * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative [step].
 * No validation on passed parameters is performed. The given parameters should satisfy the condition:
 * either `step > 0` and `start <= end`, or `step < 0` and `start >= end`.
 * @param start first element of the progression
 * @param end ending bound for the progression
 * @param step increment, or difference of successive elements in the progression
 * @return the final element of the progression
 * @suppress

@PublishedApi
@SinceKotlin("1.3")
internal fun getProgressionLastElement(start: ULong, end: ULong, step: Long): ULong = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toULong())
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toULong())
    else -> throw kotlin.IllegalArgumentException("Step is zero.")
}

@file:kotlin.jvm.JvmName("UStringsKt")  // string representation of unsigned numbers
package kotlin.text

/**
 * Returns a string representation of this [Byte] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
//@kotlin.internal.InlineOnly
public /*inline*/ fun UByte.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Parses the string as a signed [UByte] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUByte(): UByte = toUByteOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a signed [UByte] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUByte(radix: Int): UByte = toUByteOrNull(radix) ?: numberFormatError(this)

/*
 * Copyright 2010-2018 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
toUShortOrNull(radix) ?: numberFormatError(this)\n
/**
 * Parses the string as an \[UByte\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUByteOrNull(): UByte? =
    toUByteOrNull(radix = 10)\n
/**
 * Parses the string as a \[UShort\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUShortOrNull(): UShort? =
    toUShortOrNull(radix = 10)\n
/**
 * Parses the string as a \[UInt\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUIntOrNull(): UInt? =
    toUIntOrNull(radix = 10)\n
/**
 * Contains utility functions to parse strings as numbers.\n
/**
 * Parses the string as an \[ULong\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULongOrNull(): ULong? =
    toULongOrNull(radix = 10)\n
/**
 * Parses the string as a \[UByte\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUByte(): UByte.

/**
 * Parses the string as an \[UShort\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUShort(): UShort.

/**
 * Parses the string as a \[UInt\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUInt(): UInt.

/**
 * Parses the string as an \[ULong\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULong(): ULong.

/**
 * Parses the string as a \[UByte\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUByte(radix: Int): UByte.

/**
 * Parses the string as an \[UShort\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as a \[UInt\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUInt(radix: Int): UInt.

/**
 * Parses the string as an \[ULong\] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULong(radix: Int): ULong.

/**
 * Parses the string as a \[UByte\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUByteOrNull(): UByte? =
    toUByteOrNull(radix = 10)\n
/**
 * Parses the string as an \[UShort\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUShortOrNull(): UShort? =
    toUShortOrNull(radix = 10)\n
/**
 * Parses the string as a \[UInt\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUIntOrNull(): UInt? =
    toUIntOrNull(radex = 10)\n
/**
 * Parses the string as an \[ULong\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULongOrNull(): ULong? =
    toULongOrNull(radix = 10)\n
/**
 * Parses the string as a \[UByte\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUByte(radix: Int): UByte.

/**
 * Parses the string as an \[UShort\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as a \[UInt\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUInt(radex: Int): UInt.

/**
 * Parses the string as an \[ULong\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULong(radex: Int): ULong.

/**
 * Parses the string as a \[UByte\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUByte(radix: Int): UByte.

/**
 * Parses the string as an \[UShort\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as a \[UInt\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUInt(radex: Int): UInt.

/**
 * Parses the string as an \[ULong\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULong(radex: Int): ULong.

/**
 * Parses the string as a \[UByte\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as an \[UShort\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as a \[UInt\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUInt(radex: Int): UInt.

/**
 * Parses the string as an \[ULong\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULong(radex: Int): ULong.

/**
 * Parses the string as a \[UByte\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as an \[UShort\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as a \[UInt\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUInt(radex: Int): UInt.

/**
 * Parses the string as an \[ULong\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULong(radex: Int): ULong.

/**
 * Parses the string as a \[UByte\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as an \[UShort\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as a \[UInt\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toUInt(radex: Int): UInt.

/**
 * Parses the string as an \[ULong\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULong(radex: Int): ULong.

/**
 * Parses the string as a \[UByte\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes

/**
 * Parses the string as an \[UShort\] number and returns the result
 * or `null` if the string is not a valid representation of a number.

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun String.toULongOrNull(radix: Int): ULong? {
    val length = this.length
    if (length == 0) return null
    val limit = ULong.MAX_VALUE
    val start: Int
    val firstChar = this[0]
    if (firstChar == '+') {
        start = 1
    } else {
        start = 0
    }
    var result = 0uL
    for (i in start until length) {
        val digit = digitOf(this[i], radix)
        if (digit < 0) return null
        if (result > limit / uradix) {
            if (limit / uradix == limit / 36) {
                limitBeforeMul = limit / uradix
                if (result > limitBeforeMul) {
                    return null
                }
            } else {
                return null
            }
        }
        result *= uradix
        result += digit.toUInt()
        if (result < beforeAdding) return null // overflow has happened
    }
    return result
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("MathKt")

package kotlin.math

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:Suppress("INVISIBLE_REFERENCE", "INVISIBLE_MEMBER")
package kotlin

import kotlin.annotation.AnnotationTarget.*
import kotlin.internal.RequireKotlin
import kotlin.internal.RequireKotlinVersionKind

/**
 * Computes the sine of the angle [x] given in radians.
 * Special cases:
 * - sin(NaN) is NaN
 * - sin(x) is NaN when x is NaN
 */
@SinceKotlin("1.2")
public expect fun sin(x: Double): Double

/**
 * Computes the cosine of the angle [x] given in radians.
 * Special cases:
 * - cos(NaN) is NaN
 * - cos(x) is NaN when x is NaN
 */
@SinceKotlin("1.2")
public expect fun cos(x: Double): Double

/**
 * Computes the tangent of the angle [x] given in radians.
 * Special cases:
 * - tan(NaN) is NaN
 * - tan(x) is NaN when x is NaN
 */
@SinceKotlin("1.2")
public expect fun tan(x: Double): Double

/**
 * Computes the arc sine of the value [x] in the range from `-PI/2` to `PI/2` radians.
 * Special cases:
 * - asin(0) is 0
 * - asin(x) is NaN when x is outside the range from -1 to 1
 */
@SinceKotlin("1.2")
public expect fun asin(x: Double): Double

/**
 * Computes the arc cosine of the value [x] in the range from 0 to `PI` radians.
 * Special cases:
 * - acos(0) is PI/2
 * - acos(x) is NaN when x is outside the range from -1 to 1
 */
@SinceKotlin("1.2")
public expect fun acos(x: Double): Double

/**
 * Computes the arc tangent of the value [x] in the range from `-PI/2` to `PI/2` radians.
 * Special cases:
 * - atan(0) is 0
 * - atan(x) is NaN when x is NaN
 */
@SinceKotlin("1.2")
public expect fun atan(x: Double): Double

/**
 * Computes the arc cotangent of the value [x] in the range from 0 to `PI` radians.
 * Special cases:
 * - cotan(0) is 0
 * - cotan(x) is NaN when x is NaN
 */
@SinceKotlin("1.2")
public expect fun cotan(x: Double): Double

/**
 * Computes the arc secant of the value [x] in the range from 1 to `PI` radians.
 * Special cases:
 * - sec(x) is NaN when x is NaN
 */
@SinceKotlin("1.2")
public expect fun sec(x: Double): Double

/**
 * Computes the arc cosecant of the value [x] in the range from 1 to `PI` radians.
 * Special cases:
 * - csc(x) is NaN when x is NaN
 */
@SinceKotlin("1.2")
public expect fun csc(x: Double): Double

/**
 * Computes the polar coordinates (r, theta) from rectangular coordinates (x, y) by computing the arc tangent of the value [y] / [x] in the range from `-PI` to `PI` radians.
 * Special cases:
 * - atan2(0,0) is 0
 * - atan2(x,0) is 0 for x > 0
 * - atan2(0,x) is 0 for y > 0
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toULongOrNull(radix: Int): ULong? {
    checkRadix(radix)
    val length = this.length
    if (length == 0) return null
    val limit = ULong.MAX_VALUE
    val start: Int
    val firstChar = this[0]
    if (firstChar < '0') {
        if (length == 1 ||
            firstChar == '+') return null
        start = 1
    } else {
        start = 0
    }
    val limitBeforeMul = limit / uradix
    var result = 0uL
    for (i in start until length) {
        val digit = digitOf(this[i], radix)
        if (digit < 0) return null
        if (result > limitBeforeMul) {
            if (limitBeforeMul == limit / 36) {
                limitBeforeMul = limit / uradix
                if (result > limitBeforeMul) {
                    return null
                }
            } else {
                return null
            }
        }
        result *= uradix
        result += digit.toUInt()
        if (result < beforeAdding) return null // overflow has happened
    }
    return result
}
for `x < 0`\n - `atan2(-0.0, x)` is `-0.0` for `x > 0` and `-PI` for `x < 0`\n - `atan2(y, +Inf)` is `PI` for `y < +Inf` and `-PI` for `-Inf` for `y < 0`\n - `atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`\n - `atan2(+Inf, x)` is `PI/2` for finite `x`\n - `atan2(-Inf, x)` is `-PI/2` for finite `x`\n
```java
@SinceKotlin("1.2")
public expect fun atan2(y: Double, x: Double): Double

/**
 * Computes the hyperbolic sine of the value [x].\n * Special cases:\n * - `sinh(NaN)` is `NaN`\n * - `sinh(+Inf)` is `+Inf`\n * - `sinh(-Inf)` is `-Inf`\n * @see [sin\n * @SinceKotlin("1.2")
public expect fun sinh(x: Double): Double

/**
 * Computes the hyperbolic cosine of the value \n * Special cases:\n * - `cosh(NaN)` is `NaN`\n * - `cosh(+Inf)` is `+Inf`\n * - `cosh(-Inf)` is `-Inf`\n * @see [cos\n * @SinceKotlin("1.2")
public expect fun cosh(x: Double): Double

/**
 * Computes the hyperbolic tangent of the value \n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is `+Inf`\n * - `tanh(-Inf)` is `-Inf`\n * @see [tan\n * @SinceKotlin("1.2")
public expect fun tanh(x: Double): Double

/**
 * Computes the inverse hyperbolic sine of the value [x].\n * Special cases:\n * - `asinh(NaN)` is `NaN`\n * - `asinh(+Inf)` is `+Inf`\n * - `asinh(-Inf)` is `-Inf`\n * @SinceKotlin("1.2")
public expect fun asinh(x: Double): Double

/**
 * Computes the inverse hyperbolic cosine of the value \n * Special cases:\n * - `acosh(NaN)` is `NaN`\n * - `acosh(+Inf)` is `+Inf`\n * - `acosh(-Inf)` is `-Inf`\n * @SinceKotlin("1.2")
public expect fun acosh(x: Double): Double

/**
 * Computes the inverse hyperbolic tangent of the value \n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(1.0)` is `-Inf`\n * - `tanh(-1.0)` is `-Inf`\n * @SinceKotlin("1.2")
public expect fun atanh(x: Double): Double

/**
 * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.\n * Special cases:\n * - returns `-Inf` if any of arguments is infinite\n * - returns `NaN` if any of arguments is `NaN` and the other is not infinite\n * @SinceKotlin("1.2")
public expect fun hypot(x: Double, y: Double): Double

/**
 * Computes Euler's number `e` raised to the power of the value \n * Special cases:\n * - `exp(NaN)` is `NaN`\n * - `exp(+Inf)` is `+Inf`\n * - `exp(-Inf)` is `0.0`\n * @SinceKotlin("1.2")
public expect fun exp(x: Double): Double

/**
 * Computes `exp(x) - 1`\n * This function can be implemented to produce more precise result for \n * Special cases:\n * - `expm1(NaN)` is `NaN`\n * - `expm1(+Inf)` is `+Inf`\n * - `expm1(-Inf)` is `-Inf`\n * @SinceKotlin("1.2")
public expect fun expm1(x: Double): Double

/**
 * Computes the natural logarithm (base `E`) of the value \n * Special cases:\n * - `ln(NaN)` is `NaN`\n * - `ln(x)` is `NaN` when `x < 0.0`\n * @SinceKotlin("1.2")
public expect fun ln(x: Double): Double

/**
 * Computes the common logarithm (base 10) of the value \n * @see [ln\n * @SinceKotlin("1.2")
public expect fun log10(x: Double): Double

/**
 * Computes the binary logarithm (base 2) of the value \n * @SinceKotlin("1.2")
public expect fun log2(x: Double): Double

/**
 * Computes `ln1p(x)`\n * This function can be implemented to produce more precise result for \n * @SinceKotlin("1.2")
public expect fun ln1p(x: Double): Double

/**
 * Rounds the given value \n * Special cases:\n * - `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n * @SinceKotlin("1.2")
public expect fun ceil(x: Double): Double
```
integer towards negative infinity.

@return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.

Special cases:
- 'floor(x)' is 'x' where 'x' is 'NaN' or '+Inf' or '-Inf' or already a mathematical integer.
- Returns the value of the given value [x] if there is no special case.

@sinceKotlin("1.2")
public expect fun floor(x: Double): Double

/**
 * Rounds the given value [x] to the closest integer with ties rounded towards even integer.

Special cases:
- 'round(x)' is 'x' where 'x' is 'NaN' or '+Inf' or '-Inf' or already a mathematical integer.
- Returns the sign of the given value [x] if the value is negative.
- If the value is zero, then the result is 'NaN'.
- Returns the value of the given value [x] if there is no special case.

@sinceKotlin("1.2")
public expect fun round(x: Double): Double

/**
 * Returns the absolute value of the given value [x].

Special cases:
- 'abs(NaN)' is 'NaN'.
- Returns the absolute value of the given value [x] if there is no special case.

@sinceKotlin("1.2")
public expect val abs(x: Double): Double

/**
 * Returns the sign of the given value [x]:

- `-1.0` if the value is negative,
- `0.0` if the value is zero,
- `1.0` if the value is positive.

Special case:
- 'sign(NaN)' is `NaN'.
- Returns the sign of the given value [x] if there is no special case.

@sinceKotlin("1.2")
public expect fun sign(x: Double): Double

/**
 * Returns the smaller of two values.

If either value is 'NaN', then the result is 'NaN'.
- Returns the value of the given value [x] if there is no special case.

@sinceKotlin("1.2")
public expect fun min(a: Double, b: Double): Double

/**
 * Returns the greater of two values.

If either value is 'NaN', then the result is 'NaN'.
- Returns the value of the given value [x] if there is no special case.

@sinceKotlin("1.2")
public expect fun max(a: Double, b: Double): Double

/**
 * Raises this value to the power [x].

Special cases:
- 'b.pow(0.0)' is '1.0'.
- 'b.pow(1.0)' is 'b'.
- 'b.pow(NaN)' is 'NaN'.
- 'b.pow(x)' is 'NaN' for 'x' != 0.0.
- 'b.pow(x)' is 'NaN' for 'x > 0' and 'x' is finite and not an integer.
- Returns the value of the given value [x] if there is no special case.

@sinceKotlin("1.2")
public expect fun Double.pow(x: Double): Double

/**
 * Raises this value to the integer power [n].

Special cases:
- 'b.pow(0)' is '1'.
- 'b.pow(1)' is 'b'.
- 'b.pow(NaN)' is 'NaN'.
- 'b.pow(x)' is 'NaN' for 'b < 0' and 'x' is finite.
- Returns the value of the given value [x] if there is no special case.

@sinceKotlin("1.2")
public expect fun Double.pow(n: Int): Double

/**
 * Returns the absolute value of this value.

Special case:
- Returns the absolute value of this value if there is no special case.

@sinceKotlin("1.2")
public expect val Double.absoluteValue: Double

/**
 * Returns the sign of this value:

- `-1.0` if the value is negative,
- `0.0` if the value is zero,
- `1.0` if the value is positive.

Special case:
- 'NaN.sign' is `NaN`. 
- Returns the sign of this value if there is no special case.

@sinceKotlin("1.2")
public expect val Double.sign: Double

/**
 * Returns this value with the sign bit same as of the [sign] value.

If [sign] is 'NaN' the sign of the result is undefined.

@sinceKotlin("1.2")
public expect fun Double.withSign(sign: Double): Double

/**
 * Returns this value with the sign bit same as of the [sign] value.

@sinceKotlin("1.2")
public expect fun Double.withSign(sign: Int): Double

/**
 * Returns the ulp (unit in the last place) of this value.

A ulp is a positive distance between this value and the next nearest [Double] value larger in magnitude.
- 'NaN.ulp' is 'NaN'.
- 'Inf.ulp' is 'Inf'.
- 'x.ulp' is '+Inf' when 'x' is '+Inf' or '-Inf'.
- '0.0.ulp' is 'Double.MIN_VALUE'.

@sinceKotlin("1.2")
public expect val Double.ulp: Double

/**
 * Returns the value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.

@sinceKotlin("1.2")
public expect fun Double.nextDown(): Double

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.

@sinceKotlin("1.2")
public expect fun Double.nextUp(): Double
Double.roundToLong(): Long

// endregion


// region ================ Float Math

========================================

/** Computes the sine of the angle [x] given in radians.
 * Special cases:
 * - `sin(NaN)` is `NaN`
 * - `sin(±Inf)` is `±Inf`
 */
@SinceKotlin("1.2")
public expect fun sin(x: Float): Float

/** Computes the cosine of the angle [x] given in radians.
 * Special cases:
 * - `cos(NaN)` is `NaN`
 * - `cos(±Inf)` is `±1`
 */
@SinceKotlin("1.2")
public expect fun cos(x: Float): Float

/** Computes the tangent of the angle [x] given in radians.
 * Special cases:
 * - `tan(NaN)` is `NaN`
 * - `tan(±Inf)` is `±Inf`
 */
@SinceKotlin("1.2")
public expect fun tan(x: Float): Float

/** Computes the arcsine of the value [x] in the range from `-PI/2` to `PI/2` radians.
 * Special cases:
 * - `asin(NaN)` is `NaN`
 * - `asin(±Inf)` is `±Inf`
 * - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun asin(x: Float): Float

/** Computes the arccosine of the value [x] in the range from `0` to `PI` radians.
 * Special cases:
 * - `acos(NaN)` is `NaN`
 * - `acos(±Inf)` is `±Inf`
 * - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun acos(x: Float): Float

/** Computes the arctangent of the value [x] in the range from `-PI/2` to `PI/2` radians.
 * Special cases:
 * - `atan(NaN)` is `NaN`
 * - `atan(±Inf)` is `±Inf`
 * - `atan(x)` is `NaN`, when `abs(x)` > 1 or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun atan(x: Float): Float

/** Computes the arccotangent of the value [x] in the range from `0` to `PI` radians.
 * Special cases:
 * - `acot(NaN)` is `NaN`
 * - `acot(±Inf)` is `±Inf`
 * - `acot(x)` is `NaN`, when `abs(x)` > 1 or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun acot(x: Float): Float

/** Computes the arcsecant of the value [x] in the range from `-PI/2` to `PI/2` radians.
 * Special cases:
 * - `asec(NaN)` is `NaN`
 * - `asec(±Inf)` is `±Inf`
 * - `asec(x)` is `NaN`, when `abs(x)` > 1 or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun asec(x: Float): Float

/** Computes the arccosecant of the value [x] in the range from `-PI/2` to `PI/2` radians.
 * Special cases:
 * - `acsc(NaN)` is `NaN`
 * - `acsc(±Inf)` is `±Inf`
 * - `acsc(x)` is `NaN`, when `abs(x)` > 1 or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun acsc(x: Float): Float

/** Computes the hyperbolic sine of the value [x].
 * Special cases:
 * - `sinh(NaN)` is `NaN`
 * - `sinh(±Inf)` is `±Inf`
 * - `sinh(x)` is `NaN`, when `abs(x)` > 1 or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun sinh(x: Float): Float

/** Computes the hyperbolic cosine of the value [x].
 * Special cases:
 * - `cosh(NaN)` is `NaN`
 * - `cosh(±Inf)` is `±Inf`
 * - `cosh(x)` is `NaN`, when `abs(x)` > 1 or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun cosh(x: Float): Float

/** Computes the hyperbolic tangent of the value [x].
 * Special cases:
 * - `tanh(NaN)` is `NaN`
 * - `tanh(±Inf)` is `±1`
 * - `tanh(x)` is `NaN`, when `abs(x)` > 1 or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun tanh(x: Float): Float

/** Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.
 * Special cases:
 * - `sqrt(±Inf)` is `±Inf`
 * - `sqrt(x)` is `NaN`, when `abs(x)` > 1 or x is `NaN`
 */
@SinceKotlin("1.2")
public expect fun hypot(x: Float, y: Float): Float

/** Computes the positive square root of the value [x].
 * Special cases:
 * - `sqrt(x)` is `NaN`, when `x` < 0 or `x` is `NaN`
 */
@SinceKotlin("1.2")
public expect fun sqrt(x: Float): Float

/** Computes Euler's number `e` raised to the power of the value [x].
 * Special cases:
 * - `exp(NaN)` is `NaN`
 * - `exp(±Inf)` is `±Inf`
 * - `exp(-Inf)` is `0.0`
 */
@SinceKotlin("1.2")
public expect fun exp(x: Float): Float

/** Computes `expm1(x)`.
 * Special cases:
 * - `expm1(NaN)` is `NaN`
 * - `expm1(±Inf)` is `±Inf`
 * - `expm1(-Inf)` is `-1.0`
 */
@SinceKotlin("1.2")
public expect fun expm1(x: Float): Float

/** Computes the logarithm of the value [x] to the given [base].
 * Special cases:
 * - `log(0, x)` is `NaN` if either `x` or `b` are
 */
@SinceKotlin("1.2")
public expect fun log(x: Float, b: Float): Float
'NaN' \* - \log(x, b)' is 'NaN' when x < 0 or b <= 0 or b == 1.0 \* - \log(+Inf, +Inf)' is 'NaN' \* - \log(+Inf, b)' is '+Inf' for b > 1 and '-Inf' for b < 1 \* - \log(0.0, b)' is '-Inf' for b > 1 and '+Inf' for b > 1 \* \* \* \* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

\*@SinceKotlin("1.2")\*public expect fun \*log(x: Float, base: Float)\* Computes the natural logarithm (base 'E') of the value x: \*\* Special cases: x: \* - \ln(NaN)' is 'NaN' \* - \ln(x)' is 'NaN' when x < 0.0 \* - \ln(+Inf)' is '+Inf' \* - \ln(0.0)' is '-Inf'. \*@SinceKotlin("1.2")\*public expect fun ln(x: Float): \*\* Computes the binary logarithm (base 2) of the value x: \*\* \* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

\*@SinceKotlin("1.2")\*public expect fun ln1p(x: Float): \*\* Returns the smaller of two values.

\*@SinceKotlin("1.2")\*public expect fun abs(x: Float): \*\* Returns the sign of the given value:

\*@SinceKotlin("1.2")\*public expect fun sign(x: Float): \*\* Returns the absolute value of the given value:

\*@SinceKotlin("1.2")\*public expect fun truncate(x: Float): \*\* Returns the absolute value of this value.

\*@SinceKotlin("1.2")\*public expect fun withSign(sign: Int): Float\* withSign(sign: Float): Float\* Returns this value with the sign bit same as of the [sign] value.

\*@SinceKotlin("1.2")\*public expect func Float\* **abs**\* \*\* Returns the absolute value of this value.

\*@SinceKotlin("1.2")\*public expect func Float\* **max**\* Returns the sign of this value:

\*@SinceKotlin("1.2")\*public expect func Float\* **min**\* Returns the absolute value of this value:

\*@SinceKotlin("1.2")\*public expect func Float\* **pow**\* Returns this value with the sign bit same as of the [sign] value.

\*@SinceKotlin("1.2")\*public expect func Float\* **sign**\* Returns this value with the sign bit same as of the [sign] value.

\*@SinceKotlin("1.2")\*public expect func Float\* **round**\* Returns the sign of this value:

\*@SinceKotlin("1.2")\*public expect func Float\* **roundToInt**\* Returns the sign of this value:

\*@SinceKotlin("1.2")\*public expect func Float\* **sign**\* Returns the sign of this value:

\*@SinceKotlin("1.2")\*public expect func Float\* **withSign**\* Returns this value with the sign bit same as of the [sign] value.
Float.roundToInt(): `Int.minValue` * Rounds this [Float] value to the nearest integer and converts the result to [Long]. * Ties are rounded towards positive infinity. * Special cases:* - `x.roundToInt() == Int.minValue` when `x < Int.minValue` * - `x.roundToInt() == Long.MAX_VALUE` when `x > Long.MAX_VALUE` * Returns the absolute value of the given value [n]. * Special cases:* - `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow

```java
/**
 * Rounds this [Int] value to the nearest integer and converts the result to [Long].
 * Ties are rounded towards positive infinity.
 * Special cases:
 *   - `Int.minValue` is `Int.minValue` due to an overflow
 *   - `Int.minValue` is `Int.minValue` due to an overflow
 * Returns the absolute value of this value.
 */
@SinceKotlin("1.2")
public expect fun Int.minValue(): Long

/**
 * Rounds this [Int] value to the nearest integer and converts the result to [Long].
 * Ties are rounded towards positive infinity.
 * Special cases:
 *   - `Int.minValue` is `Int.minValue` due to an overflow
 *   - `Int.minValue` is `Int.minValue` due to an overflow
 * Returns the absolute value of this value.
 */
@SinceKotlin("1.2")
public expect fun Int.roundToInt(): Int
```
Open Source Used In DNAC Platform Antelope  4214
Open Source Used In DNAC Platform Antelope
Open Source Used In DNAC Platform Antelope 4262

AA,2B;QAaHb,OAaGb,cAAhB,c;UAaGb,y;UBaCZ,UAaBiD,WA+BvC,CAAY,OAAZ,C;UZj9BP,UA;UDPP,YY
m9Be,WZn9BH,WM99BW,B;GZn9Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc
,GZ9hBc,EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WEGyCd,cIAeCt,dCAAe,OAaF,
C,AeAJ,C;;QAqCT,OAMoC,W;OKA7CX,C;0GAQA,a;QAQI,b;QA5eB,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,
UAaAU,YAAY,OAAZ,C;UZj7BP,UA;UDPP,YYb9Be,WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaC,
C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;
UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,
OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,WZ99BH,WM99BW,"
B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,EAAS,MAAT,C;YAC
A,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;KKAIx,C;0FAkA,YA
AY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,
WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,
EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;
KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,
UA;UDPP,YYb9Be,WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay
9g9Bc,GZ9hBc,EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QA
Oe,OAaO,W;O;KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,
OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;
YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,
C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;
UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,
C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;
UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,
OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,WZ99BH,WM99BW,"
B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,EAAS,MAAT,C;YAC
A,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;KKAIx,C;0FAkA,YA
AY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,
WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,
EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;
KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,
UA;UDPP,YYb9Be,WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay
9g9Bc,GZ9hBc,EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;Q
OAe,OAaO,W;O;KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,
OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;
YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,
C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;
UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,
C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;
UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,
OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,WZ99BH,WM99BW,"
B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,EAAS,MAAT,C;YAC
A,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;KKAIx,C;0FAkA,YA
AY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,UA;UDPP,YYb9Be,
WZ99BH,WM99BW,B;GZ99Bx,C;UAeCl,IAAIaAaCJ,C;YACCh,aY9BuC,b;YAA5B,WZ99Bx,Ay9g9Bc,GZ9hBc,
EAAS,MAAT,C;YACA,e,;;;;YAEA,c;;UY68BA,iB;UAeCA,IAAKe,WAGeAe,C;OAaF,CAAJ,C;;QAOe,OAaO,W;O;
KKAIx,C;0FAkA,YAAY,C;QACV,2B;QAaHb,OAaGb,cAAhB,C;UAaGb,y;UBaCZ,UAaAU,YAAY,OAAZ,C;UZj9BP,
Open Source Used In DNAC Platform Antelope 4298
Open Source Used In DNAC Platform Antelope
Open Source Used In DNAC Platform Antelope 4347
Open Source Used In DNAC Platform

Antelope 4358
AAS,QAAS,WAAC,MAAC2,C,0B;MACvC,eAAAo,B,C;MACpB,MBAA4B,I;MAC5B,sBAAYC, IC;MACCZC,gBAACo,C;I;gDAEc,P,Y;MACI,0AAO,IAAP,C;QACI,QAAm,YAAN,C;eACl,C;YACI,IAAI,kCAAC,eUAAn,B;C;cACl,eAAQC,c;ACR,OAACO,I;.;;eAEp,sBAACe,I;.;;YAVB,K;eAOA,C;YAAAC,OAACO, K;eACBi,C;eAAA,C;YAACG,QAOAO,I;kBCAC,Y;BAAM,Y;B;QAGiB,eAAQC;QACR,WAAW,4B;QAXC,gBAW I;QAXC,IAAK,cAAAA,IAAP,C;K;6C4AB,Y;MACU,IAAC,e;I;MATcB,QAAm,YAAC;C:aACl,C:aAAA,C;UAAAC C;OAACo,qB;RBAC7C,C;UAACL,eAAQC;UACR,OAACO,kCAAC,e;O;aAE1B,C;UAACL,eAAQC;UACR,aACl,iF;UACh, mBAAY;I;UAACZ,OAACO,M;gBAEHC,MAAM,Y;B;K;uDAItB,Y;MACI,IAAI,CAAC,CAAL,C;QAAgB,MAAM,6B; QAABO,OAACO,W;K;2DAcD,Y;MAAC4,QAAm,YAAC;C:UACx,C;UADwC,OACI,1B6;8AcCd,C;UAFwC,OA ExB,2BAAsB,BAAC,B;C;gBAFwB,OAghC,2BAAsB,uCAoc,Y,YAA1D,C;.;;K;IAowB,4E;MAAA,eB;QAC5B,wC AAW,C;QAXC,OAACO,M;B;O;K;+DALR,+B;MACI,MBAY,Y;MACZ,eAAQC;MACR,ODPhKc,kCCoHF,0CDp HE,CAAAC,c;K;IC8H0C,+E;MAAA,eB;QAC5B,wCAAW,C;QAXC,OAACO,MB;O;K;yDANR,kC;MACI,IAAI,CA AC,QAAm,UAAD,C;QAAyB,M;MACzB,sBAEe,Q;MACf,eAAQC;MACR,OD9HkC,kCC8HF,6CD9HE,CAAAC,c ;K;qDCoIA,iB;MAEI,eAAQC;K;eEAGZ,qB;MACI,MAAM,S;K;8EAIN,Y;MAAQ,0C;K;+D9I0B,0D;MAAA,eB; QAUU,qBAMAM,IAAK,OAAC,C;O;K;1ArBPd,uD;MAcBi,OAAcs,CAAcN;C;K;8GARBJ,Y;BAAMQc,W;D;QAA AuB;UAUU,qBAAM,IAAK,OAAC,C;O;K;EAcBpD,aC;QAAyB,LMBAACs,CAAcN,6B;QAAy,B,Y;K;KARBJ,C;IA uBA,eB;MAU,MAAM,wBAAo,B,sEAcP,B,C;K;2HAVV,Y;BAAM,GE;MAAM,eB;QAUI,MAAM,6BAAo,B,sEA ApB,C;O;KAVV,C;mHAYA,yB;MAAN,GE;MAAM,eB;QAMI,MAAM,6BAAo,B,4CAAP,B,C;O;KANV;C;..............:..............................................................:..................................................:gBPR8B,S;cACF,S;OBEGiB,sC;qBMwC,CH;yBACl,C;sBACh,C;kBACJ,C;IBACD,C;MBACE, C;.;;:;C9N3EvC,E;")

Found in path(s):
* /opt/cola/permits/1177762212_1625439481.48/0/kotlin-1-3-61-tgz/package/kotlin.js.map

1.1148 babel-plugin-syntax-nullish-coalescing-operator 7.8.3
1.1148.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

Open Source Used In DNAC Platform Antelope 4372
1.1149 referrer-policy 1.0.0

1.1149.1 Available under license:

   The MIT License (MIT)

Copyright (c) 2016 Evan Hahn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1150 batik-slideshow 1.13

1.1150.1 Available under license:

   No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146690685_1616597796.72/0/batik-slideshow-1-13-sources-1-
jar/org/apache/batik/apps/slideshow/Main.java
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE
2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1146690685_1616597796.72/0/batik-slideshow-1-13-sources-1-jar/META-
INF/maven/org.apache.xmlgraphics/batik-slideshow/pom.xml

1.1151 engine.io-parser 1.3.2

1.1151.1 Available under license:
(The MIT License)

Copyright (c) 2016 Guillermo Rauch (@rauchg)

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1152 mime 2.4.4

1.1152.1 Available under license:

The MIT License (MIT)

Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1153 yargs-parser 18.1.3

1.1153.1 Available under license:

Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1154 avatica 1.17.0
1.1154.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Apache Calcite -- Avatica
Copyright 2012-2020 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.1155 gson-fire 1.5.1
1.1155.1 Available under license :
Copyright 2015 Julio Viera

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1156 cliui 6.0.0
1.1156.1 Available under license :
Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1157 dashify 0.2.2
1.1157.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1158 @babel/generator 7.11.5
1.1158.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1159 isomorphic-fetch 2.2.1 2.2.1

1.1159.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Matt Andrews

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1160 json-schema-traverse 0.3.1 0.3.1

1.1160.1 Available under license:

MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

Open Source Used In DNAC Platform Antelope 4382
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1161 lodash-es 4.17.10 4.17.10

1.1161.1 Available under license:

Copyright JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/
Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1162 tpch-sample-data 1.19.0

1.1162.1 Available under license:

Drill : Contrib : Data : TPCH Sample
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1163 @babel/traverse 7.13.13

1.1163.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1164 bson 3.12.4
1.1164.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright 2018 Cezary Bartosiak
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):

* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/jsr310/LocalDateCodec.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/jsr310/InstantCodec.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/jsr310/LocalTimeCodec.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/jsr310/Jsr310CodecProvider.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/jsr310/ LocalDateTimeCodec.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/BsonCodecProvider.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/BsonCodec.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/pojo/IdPropertyModelHolder.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/pojo/IdGenerator.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/pojo/IdGenerators.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/pojo/ConventionObjectIdGeneratorsImpl.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2008-present MongoDB, Inc.
* Copyright 2010 The Guava Authors
* Copyright 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/internal/UnsignedLongs.java

No license file was found, but licenses were detected in source scan.
/*
 * Copyright 2008-present MongoDB, Inc.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/OverridableUuidRepresentationCodec.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/json/JsonParseException.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/json/RelaxedExtendedJsonDoubleConverter.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/BsonMinKey.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/pojo/annotations/BsonDiscriminator.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/BsonWriter.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/jsr310/package-info.java

Open Source Used In DNAC Platform Antelope 4391
Open Source Used In DNAC Platform Antelope 4392
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright (c) 2008-2014 Atlassian Pty Ltd
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/util/CopyOnWriteMap.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/util/AbstractCopyOnWriteMap.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/BSO encoder.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/types/ObjectId.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/json/JsonScanner.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/io/ByteBufferBsonInput.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/ValueCodecProvider.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/BsonMaxKey.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/pojo/LazyMissingCodec.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/BsonSymbolCodec.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/configuration/CodecProvider.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/BSO N.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/ internal/UuidHelper.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/json/StrictCharacterStreamJsonWriter.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/BsonBinarySubType.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/UuidCodec.java
* /opt/cola/permits/1136447680_1613767071.02/0/bson-3-12-4-sources-2-jar/org/bson/codecs/UuidCodec.java
1.1165 js-yaml 3.14.0
1.1165.1 Available under license:
(The MIT License)

Copyright (C) 2011-2015 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1166 has-bigints 1.0.1
1.1166.1 Available under license:
MIT License

Copyright (c) 2019 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1167 is-yarn-global 0.3.0

1.1167.1 Available under license:
MIT License

Copyright (c) 2018 LitoMore

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1168 jsreport-core 2.10.1

1.1168.1 Available under license:
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.
As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.1169 json-schema-traverse 0.3.1
1.1169.1 Available under license :
MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1170 babel-plugin-transform-es2015-for-of
6.23.0 6.23.0
1.1170.1 Available under license :
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

1.1171 logback-core 1.2.3
1.1171.1 Available under license :

Found license 'Eclipse Public License 1.0' in '* Copyright (C) 1999-2015, QOS.ch. All rights reserved. * This program and the accompanying materials are dual-licensed under * either the terms of the Eclipse Public License v1.0 as published by * under the terms of the GNU Lesser General Public License version 2.1 * as published by the Free Software Foundation.'

Found license 'GNU Lesser General Public License' in '* Copyright (C) 1999-2015, QOS.ch. All rights reserved. * This program and the accompanying materials are dual-licensed under * either the terms of the Eclipse Public License v1.0 as published by * under the terms of the GNU Lesser General Public License version 2.1 * as published by the Free Software Foundation.'

1.1172 orc-core 1.5.6
1.1172.1 Available under license :

ORC Core
Copyright 2013-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertaining to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
**1.1173 woodstox-core 5.0.3**

**1.1173.1 Available under license:**

No license file was found, but licenses were detected in source scan.

```xml
<name>The Apache License, Version 2.0</name>
```

Found in path(s):

* /opt/cola/permits/1135953508_1613640887.59/0/woodstox-core-5-0-3-3-jar/META-INF/maven/com.fasterxml.woodstox/woodstox-core/pom.xml

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1472011116283
Build-Jdk: 1.7.0_79
Built-By: tatu
Bundle-Description: Woodstox is a high-performance XML processor that implements Stax (JSR-173), SAX2 and Stax2 APIs
Bundle-DocURL: https://github.com/FasterXML/woodstox
Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.txt
Bundle-ManifestVersion: 2
Bundle-Name: Woodstox
Bundle-SymbolicName: com.fastern.xml.woodstox.woodstox-core
Bundle-Vendor: FasterXML
Bundle-Version: 5.0.3
Created-By: Apache Maven Bundle Plugin
Export-Package: com.ctc.wstx.api;version="5.0.3",com.ctc.wstx.cfg;version="5.0.3",com.ctc.wstx.dom;version="5.0.3",com.ctc.wstx.dtd;version="5.0.3",com.ctc.wstx.ent;version="5.0.3",com.ctc.wstx.evt;version="5.0.3",com.ctc.wstx.exc;version="5.0.3",com.ctc.wstx.io;version="5.0.3",com.ctc.wstx.msv;version="5.0.3",com.ctc.wstx.osgi;version="5.0.3",com.ctc.wstx.sax;version="5.0.3",com.ctc.wstx.sr;version="5.0.3",com.ctc.wstx.stax;version="5.0.3",com.ctc.wstx.sw;version="5.0.3",com.ctc.wstx.util;version="5.0.3"
Implementation-Build-Date: 2016-08-24 03:58:19+0000
Implementation-Title: Woodstox
Implementation-Vendor: FasterXML
Implementation-Vendor-Id: com.fastern.xml.woodstox
Implementation-Version: 5.0.3
1.1174 is-number 2.1.0

1.1174.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.1175 extract-zip 1.7.0

1.1175.1 Available under license:

Copyright (c) 2014 Max Ogden and other contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1176 tunnel-agent 0.6.0

1.1176.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section)
patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1177 type-is 1.6.18

1.1177.1 Available under license :
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1178 babel-helper-optimise-call-expression
6.24.1

1.1178.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The "Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1179 clean-stack 2.2.0
1.1179.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1180 hive-llap-common 3.1.2
1.1180.1 Available under license:

Hive Llap Common
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.1181 bytes 2.3.0 2.3.0

1.1181.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Jed Watson <jed.watson@me.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1182 app-root-path 2.0.1

1.1182.1 Available under license:
(The MIT License)

Copyright (c) 2014 Chris Morrell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1183 p-limit 2.3.0
1.1183.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1184 kafka-schema-registry-client 6.1.1
1.1184.1 Available under license:
Confluent Community License Agreement
Version 1.0

This Confluent Community License Agreement Version 1.0 (the Agreement) sets forth the terms on which Confluent, Inc. (Confluent) makes available certain software made available by Confluent under this Agreement (the Software). BY INSTALLING, DOWNLOADING, ACCESSING, USING OR DISTRIBUTING ANY OF THE SOFTWARE, YOU AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO SUCH TERMS AND CONDITIONS, YOU MUST NOT USE THE SOFTWARE. IF YOU ARE RECEIVING THE SOFTWARE ON BEHALF OF A LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE THE ACTUAL AUTHORITY TO AGREE TO THE TERMS AND CONDITIONS OF THIS
AGREEMENT ON BEHALF OF SUCH ENTITY. Licensee means you, an individual, or the entity on whose behalf you are receiving the Software.

1. LICENSE GRANT AND CONDITIONS.

1.1 License. Subject to the terms and conditions of this Agreement, Confluent hereby grants to Licensee a non-exclusive, royalty-free, worldwide, non-transferable, non-sublicenseable license during the term of this Agreement to: (a) use the Software; (b) prepare modifications and derivative works of the Software; (c) distribute the Software (including without limitation in source code or object code form); and (d) reproduce copies of the Software (the License). Licensee is not granted the right to, and Licensee shall not, exercise the License for an Excluded Purpose. For purposes of this Agreement, Excluded Purpose means making available any software-as-a-service, platform-as-a-service, infrastructure-as-a-service or other similar online service that competes with Confluent products or services that provide the Software.

1.2 Conditions. In consideration of the License, Licensees distribution of the Software is subject to the following conditions:

(a) Licensee must cause any Software modified by Licensee to carry prominent notices stating that Licensee modified the Software.

(b) On each Software copy, Licensee shall reproduce and not remove or alter all Confluent or third party copyright or other proprietary notices contained in the Software, and Licensee must provide the notice below with each copy.

This software is made available by Confluent, Inc., under the terms of the Confluent Community License Agreement, Version 1.0 located at http://www.confluent.io/confluent-community-license. BY INSTALLING, DOWNLOADING, ACCESSING, USING OR DISTRIBUTING ANY OF THE SOFTWARE, YOU AGREE TO THE TERMS OF SUCH LICENSE AGREEMENT.

1.3 Licensee Modifications. Licensee may add its own copyright notices to modifications made by Licensee and may provide additional or different license terms and conditions for use, reproduction, or distribution of Licensees modifications. While redistributing the Software or modifications thereof, Licensee may choose to offer, for a fee or free of charge, support, warranty, indemnity, or other obligations. Licensee, and not Confluent, will be responsible for any such obligations.

1.4 No Sublicensing. The License does not include the right to sublicense the Software, however, each recipient to which Licensee provides the Software may exercise the Licensesc so long as such recipient agrees to the terms and conditions of this Agreement.
2. TERM AND TERMINATION. This Agreement will continue unless and until earlier terminated as set forth herein. If Licensee breaches any of its conditions or obligations under this Agreement, this Agreement will terminate automatically and the License will terminate automatically and permanently.

3. INTELLECTUAL PROPERTY. As between the parties, Confluent will retain all right, title, and interest in the Software, and all intellectual property rights therein. Confluent hereby reserves all rights not expressly granted to Licensee in this Agreement. Confluent hereby reserves all rights in its trademarks and service marks, and no licenses therein are granted in this Agreement.

4. DISCLAIMER. CONFLUENT HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE.

5. LIMITATION OF LIABILITY. CONFLUENT WILL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, LOST PROFITS OR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, INDIRECT, OR DIRECT DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING OUT OF THIS AGREEMENT. THE FOREGOING SHALL APPLY TO THE EXTENT PERMITTED BY APPLICABLE LAW.

6. GENERAL.

6.1 Governing Law. This Agreement will be governed by and interpreted in accordance with the laws of the state of California, without reference to its conflict of laws principles. If Licensee is located within the United States, all disputes arising out of this Agreement are subject to the exclusive jurisdiction of courts located in Santa Clara County, California, USA. If Licensee is located outside of the United States, any dispute, controversy or claim arising out of or relating to this Agreement will be referred to and finally determined by arbitration in accordance with the JAMS International Arbitration Rules. The tribunal will consist of one arbitrator. The place of arbitration will be Palo Alto, California. The language to be used in the arbitral proceedings will be English. Judgment upon the award rendered by the arbitrator may be entered in any court having jurisdiction thereof.

6.2 Assignment. Licensee is not authorized to assign its rights under this Agreement to any third party. Confluent may freely assign its rights under this Agreement to any third party.

6.3 Other. This Agreement is the entire agreement between the parties regarding the subject matter hereof. No amendment or modification of this Agreement will be valid or binding upon the parties unless made in writing and signed by the duly authorized representatives of both
parties. In the event that any provision, including without limitation
any condition, of this Agreement is held to be unenforceable, this
Agreement and all licenses and rights granted hereunder will immediately
terminate. Waiver by Confluent of a breach of any provision of this
Agreement or the failure by Confluent to exercise any right hereunder
will not be construed as a waiver of any subsequent breach of that right
or as a waiver of any other right.

Apache Kafka
Copyright 2016 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This distribution has a binary dependency on jersey, which is available under the CDDL
License. The source code of jersey can be found at https://github.com/jersey/jersey/.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
Java ClassMate library was originally written by Tatu Saloranta (tatu.saloranta@iki.fi)

Other developers who have contributed code are:

* Brian Langel
This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
Apache License, Version 2.0

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="Home page of The Apache Software Foundation">
<link rel="apple-touch-icon" sizes="57x57" href="/favicons/apple-touch-icon-57x57.png">
<link rel="apple-touch-icon" sizes="60x60" href="/favicons/apple-touch-icon-60x60.png">
<link rel="apple-touch-icon" sizes="72x72" href="/favicons/apple-touch-icon-72x72.png">
<link rel="apple-touch-icon" sizes="76x76" href="/favicons/apple-touch-icon-76x76.png">
<link rel="apple-touch-icon" sizes="114x114" href="/favicons/apple-touch-icon-114x114.png">
<link rel="apple-touch-icon" sizes="120x120" href="/favicons/apple-touch-icon-120x120.png">
<link rel="apple-touch-icon" sizes="144x144" href="/favicons/apple-touch-icon-144x144.png">
<link rel="apple-touch-icon" sizes="152x152" href="/favicons/apple-touch-icon-152x152.png">
<link rel="apple-touch-icon" sizes="180x180" href="/favicons/apple-touch-icon-180x180.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicons/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="194x194" href="/favicons/favicon-194x194.png">
<link rel="icon" type="image/png" sizes="96x96" href="/favicons/favicon-96x96.png">
<link rel="icon" type="image/png" sizes="192x192" href="/favicons/android-chrome-192x192.png">
<link rel="manifest" href="/favicons/manifest.json">
<link rel="shortcut icon" href="/favicons/favicon.ico" content="#603cba">
<meta name="msapplication-TileColor" content="#603cba">
<meta name="msapplication-TileImage" content="/favicons/mstile-144x144.png">
<meta name="msapplication-config" content="/favicons/browserconfig.xml">
<meta name="theme-color" content="#303284">
<title>Apache License, Version 2.0</title>
<link href='https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700%7cDroid+Serif:400,700' rel='stylesheet' type='text/css'>
<link href="/css/min.bootstrap.css" rel="stylesheet">
<link href="/css/styles.css" rel="stylesheet">
</head>
<body>
/* The following code is added by mdx_elementid.py */

/*
 * Hide class="elementid-permalink", except when an enclosing heading
 * has the :hover property.
 */

.headerlink, .elementid-permalink {
    visibility: hidden;
}

h2:hover > .headerlink, h3:hover > .headerlink, h1:hover > .headerlink, h6:hover > .headerlink, h4:hover > 
.headerlink, h5:hover > .headerlink, dt:hover > .elementid-permalink { visibility: visible }


<p>&lt;p&gt;TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION&lt;/p&gt;
&lt;p&gt;&lt;strong&gt;&lt;a name="definitions">1. Definitions&lt;/a&gt;&lt;/strong&gt;.&lt;/p&gt;
&lt;p&gt;"License" shall mean the terms and conditions for use, reproduction, and
distribution as defined by Sections 1 through 9 of this document.&lt;/p&gt;
&lt;p&gt;"Licensor" shall mean the copyright owner or entity authorized by the
copyright owner that is granting the License.&lt;/p&gt;
&lt;p&gt;"Legal Entity" shall mean the union of the acting entity and all other
entities that control, are controlled by, or are under common control with
that entity. For the purposes of this definition, "control" means (i) the
power, direct or indirect, to cause the direction or management of such
entity, whether by contract or otherwise, or (ii) ownership of fifty
percent (50%) or more of the outstanding shares, or (iii) beneficial
ownership of such entity.&lt;/p&gt;
&lt;p&gt;"You" (or "Your") shall mean an individual or Legal Entity exercising
permissions granted by this License.&lt;/p&gt;
&lt;p&gt;"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source,
and configuration files.&lt;/p&gt;
&lt;p&gt;"Object" form shall mean any form resulting from mechanical transformation
or translation of a Source form, including but not limited to compiled
object code, generated documentation, and conversions to other media types.&lt;/p&gt;
&lt;p&gt;"Work" shall mean the work of authorship, whether in Source or Object form,
made available under the License, as indicated by a copyright notice that
is included in or attached to the work (an example is provided in the
Appendix below).&lt;/p&gt;
&lt;p&gt;"Derivative Works" shall mean any work, whether in Source or Object form,
that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as
a whole, an original work of authorship. For the purposes of this License,
Derivative Works shall not include works that remain separable from, or
merely link (or bind by name) to the interfaces of, the Work and Derivative
Works thereof.&lt;/p&gt;
&lt;p&gt;"Contribution" shall mean any work of authorship, including the original
version of the Work and any modifications or additions to that Work or
Derivative Works thereof, that is intentionally submitted to Licensor for
inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution.”</p>
<p>“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.</p>
<p>Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.</p>
<p>Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.</p>
<p>You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:</p>
<ol style="list-style: lower-latin;">
<li>You must give any other recipients of the Work or Derivative Works a copy of this License; and</li>
<li>You must cause any modified files to carry prominent notices stating that You changed the files; and</li>
<li>You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and</li>
</ol>
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and
under no legal theory, whether in tort (including negligence), contract, or
evertheless, unless required by applicable law (such as deliberate and
grossly negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result
of this License or out of the use or inability to use the Work (including
but not limited to damages for loss of goodwill, work stoppage, computer
failure or malfunction, or any and all other commercial damages or losses),
even if such Contributor has been advised of the possibility of such
damages.\p>
\p><strong>9. Accepting Warranty or Additional Liability</strong>. While redistributing the Work or Derivative Works thereof, You may choose
to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License.
However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor,
and only if You agree to indemnify, defend, and hold each Contributor
harmless for any liability incurred by, or claims asserted against, such
Contributor by reason of your accepting any such warranty or additional
liability.\p>
\p>END OF TERMS AND CONDITIONS</p>

**APPENDIX: How to apply the Apache License to your work**

To apply the Apache License to your work, attach the following boilerplate
notice, with the fields enclosed by brackets "[]" replaced with your own
identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also
recommend that a file or class name and description of purpose be included
on the same "printed page" as the copyright notice for easier
identification within third-party archives.\p>

```
Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

<!- Footer -->

<footer class="bg-primary">
<div class="container">
  <div class="row">
    <br />
    <div class="col-sm-1">
    </div>
    <div class="col-sm-2">
      <h5 class="white">Community</h5>
      <ul class="list-unstyled white role="menu">
        <li><a href="http://community.apache.org/">Overview</a></li>
        <li><a href="/foundation/conferences.html">Conferences</a></li>
        <li><a href="http://community.apache.org/newcomers/">Getting Started</a></li>
        <li><a href="/foundation/how-it-works.html">The Apache Way</a></li>
        <li><a href="/travel/">Travel Assistance</a></li>
        <li><a href="/foundation/getinvolved.html">Get Involved</a></li>
        <li><a href="http://community.apache.org/newbiefaq.html">Community FAQ</a></li>
      </ul>
    </div>
    <div class="col-sm-2">
      <h5 class="white">Innovation</h5>
      <ul class="list-unstyled white role="menu">
        <li><a href="http://incubator.apache.org/">Incubator</a></li>
        <li><a href="http://labs.apache.org/">Labs</a></li>
        <li><a href="/licenses/">Licensing</a></li>
        <li><a href="/foundation/license-faq.html">Licensing FAQ</a></li>
        <li><a href="/foundation/marks/">Trademark Policy</a></li>
        <li><a href="/foundation/contact.html">Contacts</a></li>
      </ul>
    </div>
    <div class="col-sm-2">
      <h5 class="white">Tech Operations</h5>
      <ul class="list-unstyled white role="menu">
        <li><a href="/dev/">Developer Information</a></li>
        <li><a href="/dev/infrastructure.html">Infrastructure</a></li>
        <li><a href="/security/">Security</a></li>
        <li><a href="http://status.apache.org">Status</a></li>
        <li><a href="/foundation/contact.html">Contacts</a></li>
      </ul>
    </div>
    <div class="col-sm-2">
      <h5 class="white">Press</h5>
      <ul class="list-unstyled white role="menu">
        <li><a href="/press/">Overview</a></li>
      </ul>
    </div>
  </div>
</div>
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
This copy of Jackson JSON processor annotations is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://opensource.org/licenses/mit-license.php">here</a>.</p>
</body></html>
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: schema-registry
Source: https://github.com/confluentinc/schema-registry

Files: *
Copyright: 2018 Confluent, Inc.
License: Apache-2

Files: core/*
Copyright: 2015 Confluent, Inc.
License: Confluent Community License

License: Confluent Community License
Licensed under the Confluent Community License; you may not use this file
except in compliance with the License. You may obtain a copy of the License at

http://www.confluent.io/confluent-community-license

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OF ANY KIND, either express or implied. See the License for the
specific language governing permissions and limitations under the License.

License: Apache-2
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

On Debian systems, the Apache 2.0 license can be found in /usr/share/common-licenses/Apache-2.0.

__Apache License__

Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This distribution has a binary dependency on jersey, which is available under the CDDL License as described below.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL - Version 1.1)
1. Definitions.
1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.
2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.
3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the
name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.
COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.
6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S
NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California. The following commands were used to generate license and notice files. Replace <VERSION> with the Schema Registry version, <SRC_PATH> with the path to the Schema Registry source directory, and <LICENSE_TOOL_PATH> with the path of the license tool.

```
cd <SRC_PATH>
mvn package -DskipTests
mkdir /tmp/jars
mkdir /tmp/overrides
```
Apache Avro
Copyright 2009-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en-US">
<head>
<title>GlassFish Dual License Header and License Notice GPL v2 and CDDL</title>
<style type="text/css">
li {display: block;}
</style>
</head>
<body leftmargin="0" topmargin="0" marginheight="0" marginwidth="0" bgcolor="#ffffff">
<h3>COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0</h3>

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. “Modifications” means the Source Code and Executable form of any of the following:

- A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
- B. Any new file that contains any part of the Original Software or previous Modification; or
- C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. “Original Software” means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. “Patent Claims” means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
<p>(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
</p>
<p>(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
</p>
<p>(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.
</p>

### 2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an
unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
</p>
</li>
<li>
<p><b>(b)</b> under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1)&nbsp;Modifications made by that Contributor (or portions thereof); and (2)&nbsp;the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
</p>
</li>
<li>
<p><b>(c)</b> The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
</p>
</li>
<li>
<p><b>(d)</b> Notwithstanding Section 2.2(b) above, no patent license is granted: (1)&nbsp;for any code that Contributor has deleted from the Contributor Version; (2)&nbsp;for infringements caused by: (i)&nbsp;third party modifications of Contributor Version, or (ii)&nbsp;the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3)&nbsp;under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.
</p>
</li>
</ul>
</li>
</ul>
</li>
</ul>
</li>
</ul>

</p>

3. Distribution Obligations.</p>
3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so
only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

</p>
</li>
<li>
<p><b>3.5. Distribution of Executable Versions.</b></p>
<p>You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.
</p>
</li>
<li>
<p><b>3.6. Larger Works.</b></p>
<p>You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.
</p>
</li>
<li>
<br/>
<ul><li><p><b>4. Versions of the License.</b></p>
</li></ul>
</li>
</ul>
Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.
COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.
6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software and commercial computer software documentation; as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This
U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.
</p>
</li>

<li>
<p><b>9. MISCELLANEOUS.</b></p>
<p>This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.</p>
</li>

<li>
<p><b>10. RESPONSIBILITY FOR CLAIMS.</b></p>
<p>As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.</p>
</li>
NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT 
AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws 
of the State of California (excluding conflict-of-law provisions). 
Any litigation relating to this License shall be subject to the 
jurisdiction of the Federal Courts of the Northern District of 
California and the state courts of the State of California, with 
venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies 
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your 
freedom to share and change it. By contrast, the GNU General Public 
License is intended to guarantee your freedom to share and change 
free software--to make sure the software is free for all its users. 
This General Public License applies to most of the Free Software 
Foundation's software and to any other program whose authors commit 
to using it. (Some other Free Software Foundation software is covered 
by the GNU Library General Public License instead.) You can apply it 
to your programs, too.

When we speak of free software, we are referring to freedom, not 
price. Our General Public Licenses are designed to make sure that 
you have the freedom to distribute copies of free software (and 
charge for this service if you wish), that you receive source code 
or can get it if you want it, that you can change the software or
use pieces of it in new free programs; and that you know you can
do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the
rights. These restrictions translate to certain responsibilities
for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights
that you have. You must make sure that they, too, receive or can get
the source code. And you must show them these terms so they know
their rights.

We protect your rights with two steps: (1) copyright the software,
and (2) offer you this license which gives you legal permission to
copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on,
we want its recipients to know that what they have is not the original,
so that any problems introduced by others will not reflect on the
original authors' reputations.

Finally, any free program is threatened constantly by software patents.
We wish to avoid the danger that redistributors of a free program will
individually obtain patent licenses, in effect making the program
proprietary. To prevent this, we have made it clear that any patent
must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

This License applies to any program or other work which
contains a notice placed by the copyright holder saying it may be
distributed under the terms of this General Public License. The
"Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law; that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".
</p>
<p>
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
</p>
<p>
<b>1.</b> You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
</p>
<p>
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
</p>
<p>
<b>2.</b> You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
</p>
<ul>
<li>
<p>
a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
</p>
</li>
</ul>
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

p>
</li>
</ul>

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

p>
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

p>
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

li>
</p>
</li>

li>
</p>

<b>3. </b>You may copy and distribute the Program (or a work based
source code, even though third parties are not compelled to copy the 
source along with the object code.

<p></p>

<li>
<p>
<b>4.</b> You may not copy, modify, sublicense, or distribute the 
Program except as expressly provided under this License. Any attempt 
otherwise to copy, modify, sublicense or distribute the Program is void, 
and will automatically terminate your rights under this License. However, 
parties who have received copies, or rights, from you under this License 
will not have their licenses terminated so long as such parties remain 
in full compliance.
</p>
</li>

<li>
<p>
<b>5.</b> You are not required to accept this License, since you have 
not signed it. However, nothing else grants you permission to modify or 
distribute the Program or its derivative works. These actions are prohibited 
by law if you do not accept this License. Therefore, by modifying or 
distributing the Program (or any work based on the Program), you indicate 
your acceptance of this License to do so, and all its terms and conditions 
for copying, distributing or modifying the Program or works based on it.
</p>
</li>

<li>
<p>
<b>6.</b> Each time you redistribute the Program (or any work based on the 
Program), the recipient automatically receives a license from the original 
licensor to copy, distribute or modify the Program subject to these terms 
and conditions. You may not impose any further restrictions on the recipients' 
exercise of the rights granted herein. You are not responsible for enforcing 
compliance by third parties to this License.
</p>
</li>

<li>
<p>
<b>7.</b> If, as a consequence of a court judgment or allegation of patent 
infringement or for any other reason (not limited to patent issues), 
conditions are imposed on you (whether by court order, agreement or otherwise) 
that contradict the conditions of this License, they do not excuse you from 
the conditions of this License. If you cannot distribute so as to satisfy 
simultaneously your obligations under this License and any other pertinent
obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

</p>
</li>

<li>
<p>END OF TERMS AND CONDITIONS</p>
</li>

<li>
<p><b>How to Apply These Terms to Your New Programs</b></p>
</li>

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

<p>To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

</p>
Open Source Used In DNAC Platform Antelope 4476

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words: "Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.
<strong>COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1</strong>

1. Definitions.

1.1. &ldquo;Contributor&rdquo; means each individual or entity that creates or contributes to the creation of Modifications.

1.2. &ldquo;Contributor Version&rdquo; means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. &ldquo;Covered Software&rdquo; means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. &ldquo;Executable&rdquo; means the Covered Software in any form other than Source Code.

1.5. &ldquo;Initial Developer&rdquo; means the individual or entity that first makes Original Software available under this License.

1.6. &ldquo;Larger Work&rdquo; means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. &ldquo;License&rdquo; means this document.

1.8. &ldquo;Licensable&rdquo; means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. &ldquo;Modifications&rdquo; means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. &ldquo;Original Software&rdquo; means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. &ldquo;Patent Claims&rdquo; means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. &ldquo;Source Code&rdquo; means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. &ldquo;You&rdquo; (or &ldquo;Your&rdquo;) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, &ldquo;You&rdquo; includes...
any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

<p>2. License Grants.</p>

<blockquote>2.1. The Initial Developer Grant.</blockquote>

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

<blockquote>(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

<blockquote>(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on
or through a medium customarily used for software exchange.</blockquote>

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any
references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S
NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.</p>

8. U.S. GOVERNMENT END USERS.</blockquote>

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation"; as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.</blockquote>

9. MISCELLANEOUS.</blockquote>

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.</blockquote>

10. RESPONSIBILITY FOR CLAIMS.</blockquote>

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.</blockquote>

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

THE GNU General Public License (GPL) Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.
of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" AND/OR WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) &lt;year&gt; &lt;name of author&gt;

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public
License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items—whatever suits your program.</p>
<p>You should also get your employer (if you work as a programmer) or your school, if any, to sign a &quot;copyright disclaimer&quot; for the program, if necessary. Here is a sample; alter the names: </p><p>Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.</p><p>Ty Coon, President of Vice</p><p>This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.</p><li style="background-color:yellow;">
<p><strong>"CLASSPATH" EXCEPTION TO THE GPL VERSION 2</strong></p><p>Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."</p><br />
<p>Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.</p><br />
<p>As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.</p></li>
<p></p><p>This copy of Jackson JSON processor is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.</p><p>You may obtain a copy of the License at:</p><p>http://www.apache.org/licenses/</p><p>A copy is also included with both the downloadable source code package</p>
and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/asl"; and in jar "META-INF/"
The project is licensed under the Confluent Community License, except for client libs, which is under the Apache 2.0 license.

See LICENSE file in each subfolder for detailed license agreement.
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.
It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="Home page of The Apache Software Foundation">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="Home page of The Apache Software Foundation">

<link rel="apple-touch-icon" sizes="57x57" href="/favicons/apple-touch-icon-57x57.png">
<link rel="apple-touch-icon" sizes="60x60" href="/favicons/apple-touch-icon-60x60.png">
<link rel="apple-touch-icon" sizes="72x72" href="/favicons/apple-touch-icon-72x72.png">
<link rel="apple-touch-icon" sizes="76x76" href="/favicons/apple-touch-icon-76x76.png">
<link rel="apple-touch-icon" sizes="114x114" href="/favicons/apple-touch-icon-114x114.png">
<link rel="apple-touch-icon" sizes="120x120" href="/favicons/apple-touch-icon-120x120.png">
<link rel="apple-touch-icon" sizes="144x144" href="/favicons/apple-touch-icon-144x144.png">
<link rel="apple-touch-icon" sizes="152x152" href="/favicons/apple-touch-icon-152x152.png">
Open Source Used In DNAC Platform Antelope 4489

<!-- Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"; you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. -->

</head>

<body>

<!-- Navigation -->
<header>
<nav class="navbar navbar-default navbar-fixed-top">
<div class="container">
<div class="navbar-header">
<button class="navbar-toggle" type="button" data-toggle="collapse" data-target="#mainnav-collapse">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a href="#" class="navbar-brand"><span class="glyphicon glyphicon-home"></span></a>
</div>
<div class="collapse navbar-collapse" id="mainnav-collapse">
<div style="line-height:20px; padding-top:5px; float:left"><a href="/">Home</a> &raquo; <a href="/licenses/">Licenses</a></div>
</div>
</nav>
</header>

</body>
The Apache Software Foundation uses various licenses to distribute software and documentation, to accept regular contributions from individuals and corporations, and to accept larger grants of existing software products. These licenses help us achieve our goal of providing reliable and
long-lived software products through collaborative open source software
development. In all cases, contributors retain full rights to use their
original contributions for any other purpose outside of Apache while
providing the ASF and its projects the right to distribute and build upon
their work within Apache.

<h1 id="distributions">Licensing of Distributions</h1>
<p>All software produced by The Apache Software Foundation or any of its
projects or subjects is licensed according to the terms of the documents
listed below.</p>

<h3 id="2.0">Apache License, Version 2.0 (current)</h3>
<p><a href="LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a> (<a href="LICENSE-2.0.txt">TXT</a> or <a href="LICENSE-2.0.html">HTML</a>)</p>
<p>The 2.0 version of the Apache License was approved by the ASF in 2004. The
goals of this license revision have been to reduce the number of frequently
asked questions, to allow the license to be reusable without modification
by any project (including non-ASF projects), to allow the license to be
included by reference instead of listed in every file, to clarify the
license on submission of contributions, to require a patent license on
contributions that necessarily infringe the contributor's own patents, and
to move comments regarding Apache and other inherited attribution notices
to a location outside the license terms (the <a href="example-NOTICE.txt">NOTICE
file</a> ).</p>
<p>The result is a license that is supposed to be compatible with other open
source licenses, while remaining true to the original goals of the Apache
Group and supportive of collaborative development across both nonprofit and
commercial organizations. The Apache Software Foundation is still trying to
determine if this version of the Apache License is <a href="GPL-compatibility.html">compatible with the
GPL</a>.</p>
<p>All packages produced by the ASF are implicitly licensed under the Apache
License, Version 2.0, unless otherwise explicitly stated. More developer
documentation on how to apply the Apache License to your work can be found
in * <a href="../dev/apply-license.html">Applying the Apache License, Version 2.0</a> *</p>

<h3 id="1.1">Apache License, Version 1.1 (historic)</h3>
<p><a href="LICENSE-1.1">http://www.apache.org/licenses/LICENSE-1.1</a></p>
<p>The 1.1 version of the Apache License was approved by the ASF in 2000. The
primary change from the 1.0 license is in the 'advertising clause' (section
3 of the 1.0 license); derived products are no longer required to include
attribution in their advertising materials, only in their documentation.</p>
<p>Individual packages licensed under the 1.1 version may have used different
wording due to varying requirements for attribution or mark identification,
but the binding terms were all the same.</p>

<h3 id="1.0">Apache License, Version 1.0 (historic)</h3>
<p><a href="LICENSE-1.0">http://www.apache.org/licenses/LICENSE-1.0</a></p>
This is the original Apache License which applies only to older versions of Apache packages (such as version 1.2 of the Web server).

The ASF desires that all contributors of ideas, code, or documentation to any Apache projects complete, sign, and submit (via fax or email) an 'Individual Contributor License Agreement' (ICLA). The purpose of this agreement is to clearly define the terms under which intellectual property has been contributed to the ASF and thereby allow us to defend the project should there be a legal dispute regarding the software at some future time. A signed ICLA is required to be on file before an individual is given commit rights to an ASF project.

For a corporation that has assigned employees to work on an Apache project, a 'Corporate CLA' (CCLA) is available for contributing intellectual property via the corporation, that may have been assigned as part of an employment agreement. Note that a Corporate CLA does not remove the need for every developer to sign their own ICLA as an individual, to cover any of their contributions which are not owned by the corporation signing the CCLA.

The ICLA is not tied to any employer you may have, so it is recommended to use one's personal email address in the contact details, rather than an @work address.

Your Full name will be published unless you provide an alternative Public name. For example if your full name is Andrew Bernard Charles Dickens, but you wish to be known as Andrew Dickens, please enter the latter as your Public name.

The email address and other contact details are not published.

When an individual or corporation decides to donate a body of existing software or documentation to one of the Apache projects, they need to execute a formal 'Software Grant Agreement' (SGA) with the ASF. Typically, this is done after negotiating approval with the ASF or one of the PMCs, since the ASF will not accept software unless there is a viable community available to support a collaborative project.

Documents may be submitted by fax or email. If submitting by fax, please print, sign, and fax all pages of the document to +1-919-573-9199. Please send documents right-side-up, first page first; and send only one document per fax.

If submitting by email, please fill the form with a pdf viewer, print, sign, scan all pages into a single pdf file, and email the pdf file as an attachment to secretary@apache.org. If possible, send the attachment from the email address in the document. Please send only one document per email.

If you prefer to sign electronically, please fill the form, save it locally (e.g. icla.pdf), and sign the file by preparing a detached PGP signature. For example,
The above will create a file icla.pdf.asc. Send both the file and signature as attachments in the same email to secretary@apache.org. Please send only one document (file plus signature) per email. Please do not submit your public key to Apache. Instead, please upload your public key to pgpkeys.mit.edu.

The files typically are named icla.pdf and icla.pdf.asc for individual agreements; ccla.pdf and ccla.pdf.asc for corporate agreements; software-grant.pdf and software-grant.pdf.asc for grants.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.

For export restriction information, please consult our ASF Export Classifications page.

For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.

For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
either as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product currently only contains code developed by authors of specific components, as identified by the source code files; if such notes are missing files have been created by Tatu Saloranta.

For additional credits (generally to people who reported problems) see CREDITS file.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1185 @babel/plugin-proposal-private-methods 7.13.0

1.1185.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1186 libxext 1.3.3-1

1.1186.1 Available under license:


Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be
used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization from The Open Group.

Copyright (c) 1996 Digital Equipment Corporation, Maynard, Massachusetts.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software.

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
DIGITAL EQUIPMENT CORPORATION BE LIABLE FOR ANY CLAIM, DAMAGES, INCLUDING,
BUT NOT LIMITED TO CONSEQUENTIAL OR INCIDENTAL DAMAGES, OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR
IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Digital Equipment Corporation
shall not be used in advertising or otherwise to promote the sale, use or other
dealings in this Software without prior written authorization from Digital
Equipment Corporation.

Copyright (c) 1997 by Silicon Graphics Computer Systems, Inc.
Permission to use, copy, modify, and distribute this
software and its documentation for any purpose and without
fee is hereby granted, provided that the above copyright
notice appear in all copies and that both that copyright
notice and this permission notice appear in supporting
documentation, and that the name of Silicon Graphics not be
used in advertising or publicity pertaining to distribution
of the software without specific prior written permission.
Silicon Graphics makes no representation about the suitability
of this software for any purpose. It is provided "as is"
without any express or implied warranty.
SILICON GRAPHICS DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS
SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SILICON
GRAPHICS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1992 Network Computing Devices

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of NCD not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. NCD makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

NCD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NCD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.


All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Digital or Olivetti not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL AND OLIVETTI DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THEY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1986, 1987, 1988 by Hewlett-Packard Corporation

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without
fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

Hewlett-Packard makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Copyright (c) 1994, 1995 Hewlett-Packard Company

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL HEWLETT-PACKARD COMPANY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the Hewlett-Packard Company shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the Hewlett-Packard Company.

Copyright Digital Equipment Corporation, 1996

Permission to use, copy, modify, distribute, and sell this documentation for any purpose is hereby granted without fee, provided that the above copyright notice and this permission notice appear in all copies. Digital Equipment Corporation makes no representations about the suitability for any purpose
of the information in this document. This documentation is provided "as is" without express or implied warranty.

Copyright (c) 1999, 2005, 2006, 2013, Oracle and/or its affiliates. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 1989 X Consortium, Inc. and Digital Equipment Corporation.
Copyright (c) 1992 X Consortium, Inc. and Intergraph Corporation.
Copyright (c) 1993 X Consortium, Inc. and Silicon Graphics, Inc.
Copyright (c) 1994, 1995 X Consortium, Inc. and Hewlett-Packard Company.

Permission to use, copy, modify, and distribute this documentation for any purpose and without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

Digital Equipment Corporation, Intergraph Corporation, Silicon Graphics, Hewlett-Packard, and the X Consortium make no representations about the suitability for any purpose of the information in this document. This documentation is provided "as is" without express or implied warranty.

1.1187 proto-google-common-protos 1.17.0
1.1187.1 Available under license:

No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
Found in path(s):
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/google/api/launch_stage.proto
No license file was found, but licenses were detected in source scan.

/**
 * <pre>
 * Early Access features are limited to a closed group of testers. To use
 * these features, you must sign up in advance and sign a Trusted Tester
 * agreement (which includes confidentiality provisions). These features may
 * be unstable, changed in backward-incompatible ways, and are not
 * guaranteed to be released.
 * </pre>
 * *
 * <code>EARLY_ACCESS = 1;</code>
 */

Found in path(s):
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/com/google/api/LaunchStage.java
No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//     http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/google/api/monitoring.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/google/api/control.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/google/api/documentation.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/google/api/resource.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/google/type/latlng.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/google/type/dayofweek.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
    jar/google/type/fraction.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/logging/type/log_severity.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/system_parameter.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/label.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/monitored_resource.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/endpoint.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/usage.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/type/color.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/client.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/longrunning/operations.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/billing.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/metric.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/config_change.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/type/timeofday.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/annotations.proto
* /opt/cola/permits/1199767240_1630583079.56/0/proto-google-common-protos-1-17-0-sources-1-
jar/google/api/service.proto

1.1188 bluebird 3.7.2 3.7.2

1.1188.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013-2018 Petka Antonov

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1189 klaw 1.3.1
1.1189.1 Available under license :
(The MIT License)

Copyright (c) 2015-2016 JP Richardson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1190 commons-cli 1.4
1.1190.1 Available under license :

Apache Commons CLI
Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions.
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1191 jsreport-authentication 2.6.1

1.1191.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

a) under this License, provided that you make a good faith effort to
ensure that, in the event an Application does not supply the
function or data, the facility still operates, and performs
whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of
this License applicable to that copy.


The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the
Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license
document.


You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:
a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.1192 jsreport-public-templates 2.2.0
1.1192.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.
0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.1193 y18n 4.0.3 4.0.3
1.1193.1 Available under license:
Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1194 generate-function 2.0.0 2.0.0
1.1194.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
1.1195 babel-plugin-transform-es2015-modules-systemjs 6.24.1
1.1195.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1196 paranamer 2.5.6

1.1197 re2j 1.1

1.1198 send 0.13.2

1.1198.1 Available under license :

(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
1.1199 types-color-name 1.1.1

Available under license:
MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1200 babel-helper-replace-supers 6.24.1

Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
1.1201 memory 0.9.0

1.1201.1 Available under license:

Arrow Memory
Copyright 2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1202 babel-helper-explode-assignable-expression 6.24.1 6.24.1

1.1202.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1203 component-emitter 1.3.0 1.3.0
1.1203.1 Available under license:
(The MIT License)

Copyright (c) 2014 Component contributors <dev@component.io>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1204 node-fetch 1.7.3 1.7.3
1.1204.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 David Frank

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1205 generate-object-property 1.2.0 1.2.0

1.1205.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1206 define-property 0.2.5

1.1206.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1207 uzaygezen-core 0.2

1.1207.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 Google Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-jar/com/google/uzaygezen/core/BitSetMath.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-jar/com/google/uzaygezen/core/BitSetBackedBitVector.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-jar/com/google/uzaygezen/core/FilteredIndexRange.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-jar/com/google/uzaygezen/core/IndexCalculator.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-jar/com/google/uzaygezen/core/SpaceVisitor.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/MapRegionInspector.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/Query.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/SimpleRegionInspector.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/MultiDimensionalSpec.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/RegionInspector.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/LongBitVector.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/PlainFilterCombiner.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/QueryFactory.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/CompactHilbertCurve.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/HilbertIndexMasks.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/TestUtils.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/ZoomingNavigator.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/LinkedNodeList.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/BoundedRollup.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/SelectiveFilter.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/FilterCombiner.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/ZoomingSpaceVisitorAdapter.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/LongArrayBitVector.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/BitVectorMath.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/ranges/LongRange.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/RangeListFilter.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/NodeList.java
* /opt/ cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/SpatialRelation.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/Pow2LengthBitSetRange.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/BitVector.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/QueryBuilder.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/ListConcatCombiner.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/Pow2LengthBitSetRangeFactory.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/BitVectorFactories.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/MathUtils.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/Assessment.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/BitSetComparator.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/SpaceFillingCurve.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/AdditiveValue.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/BacktrackingQueryBuilder.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/MapNode.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/StreamingRollup.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/package-info.java
* /opt/cola/permits/1202446435_1631568163.64/0/uzaygezen-core-0-2-sources-
jar/com/google/uzaygezen/core/NodeValue.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 Daniel Aioanei.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
Copyright (c) 2015 James Messinger

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1209 es6-map 0.1.5

1.1209.1 Available under license:
Copyright (C) 2013 Mariusz Nowak (www.medikoo.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1210 accepts 1.3.3 1.3.3

1.1210.1 Available under license:
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1211 negotiator 0.6.2

1.1211.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schlueter
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1212 private 0.1.8 0.1.8

1.1212.1 Available under license:

Copyright (c) 2014 Ben Newman <bn@cs.stanford.edu>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
Included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1213.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1214 pseudomap 1.0.2
1.1214.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1215 ms 2.1.1 2.1.1
1.1215.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Zeit, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1216 js-tokens 4.0.0 4.0.0
1.1216.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014, 2015, 2016, 2017, 2018 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1217 revalidator 0.1.8
1.1217.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
1.1218 liftoff 3.0.0

1.1218.1 Available under license:

Copyright (c) 2014 Tyler Kellen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1219 form-data 2.5.1

1.1219.1 Available under license:

Copyright (c) 2012 Felix Geisendrfer (felix@debuggable.com) and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1220 lodash-pickby 4.6.0

1.1220.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.
Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1221 package-json 6.5.0

1.1221.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1222 cairo 1.14.6 1

1.1222.1 Available under license:

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

Cairo is free software.

cairo-trace is released under the terms of the GNU General Public License
/GPL) version 3. Please see COPYING-GPL-3 for the precise terms and
conditions.
Cairo is free software.
These tests are mainly available under a liberal MIT license to simplify any use of the code for reference purposes. Please check the opening comment of each file for copyright and licensing information.

The test suite also bundles some fonts for use by the test suite. The fonts included, their licenses, and why we use them in the test suite are as follows:

<table>
<thead>
<tr>
<th>Font</th>
<th>License</th>
<th>Distinguishing feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>6x13.pcf</td>
<td>Public Domain</td>
<td>Bitmap font</td>
</tr>
</tbody>
</table>

The test suite also bundles some images for use by the test suite. The images included, their licenses, and why we use them in the test suite are as follows:

<table>
<thead>
<tr>
<th>Image</th>
<th>License</th>
<th>Distinguishing feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>romedalen.jpg</td>
<td>Public Domain</td>
<td>Bitmap image (image/jpeg)</td>
</tr>
<tr>
<td>romedalen.png</td>
<td>Public Domain</td>
<td>Bitmap image (image/png)</td>
</tr>
</tbody>
</table>

The kind contributors of the bundled files are (in alphabetical order):

yvind Kols <pippin@freedesktop.org> Author of the original romedalen shot.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS
0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no
more than your reasonable cost of physically performing this
conveying of source, or (2) access to copy the
Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the
written offer to provide the Corresponding Source. This
alternative is allowed only occasionally and noncommercially, and
only if you received the object code with such an offer, in accord
with subsection 6b.

d) Convey the object code by offering access from a designated
place (gratis or for a charge), and offer equivalent access to the
Corresponding Source in the same way through the same place at no
further charge. You need not require recipients to copy the
Corresponding Source along with the object code. If the place to
copy the object code is a network server, the Corresponding Source
may be on a different server (operated by you or a third party)
that supports equivalent copying facilities, provided you maintain
clear directions next to the object code saying where to find the
Corresponding Source. Regardless of what server hosts the
Corresponding Source, you remain obligated to ensure that it is
available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided
you inform other peers where the object code and Corresponding
Source of the work are being offered to the general public at no
charge under subsection 6d.

A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling. In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage. For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product. A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

"Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.
Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.
You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that
You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.
Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.
IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.
Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```plaintext
<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Cairo is free software.

These tests are mainly available under a liberal MIT license to simplify any use of the code for reference purposes. Please check the opening comment of each file for copyright and licensing information.

Cairo is free software.

Every source file in the implementation of cairo is available to be redistributed and/or modified under the terms of either the GNU Lesser General Public License (LGPL) version 2.1 or the Mozilla Public License (MPL) version 1.1. Some files are available under more liberal terms, but we believe that in all cases, each file may be used under either the LGPL or the MPL.

See the following files in this directory for the precise terms and conditions of either license:

COPYING-LGPL-2.1
COPYING-MPL-1.1

Please see each file in the implementation for copyright and licensing information, (in the opening comment of each file).
Cairo is free software.

These utilities are all free software, please see the licensing conditions in the opening comments of each file.
Cairo is free software.

Every source file in the implementation[*] of cairo is available to be redistributed and/or modified under the terms of either the GNU Lesser General Public License (LGPL) version 2.1 or the Mozilla Public License (MPL) version 1.1. Some files are available under more liberal terms, but we believe that in all cases, each file may be used under either the LGPL or the MPL.

See the following files in this directory for the precise terms and conditions of either license:

COPYING-LGPL-2.1
COPYING-MPL-1.1

Please see each file in the implementation for copyright and licensing information, (in the opening comment of each file).

[*] The implementation of cairo is contained entirely within the "src" directory of the cairo source distribution. There are other components of the cairo source distribution (such as the "test", "util", and "perf") that are auxiliary to the library itself. None of the source code in these directories contributes to a build of the cairo library itself, (libcairo.so or cairo.dll or similar).

These auxiliary components are also free software, but may be under different license terms than cairo itself. For example, most of the test cases in the perf and test directories are made available under an MIT license to simplify any use of this code for reference purposes in using cairo itself. Other files might be available under the GNU General Public License (GPL), for example. Again, please see the COPYING file under each directory and the opening comment of each file for copyright and licensing information.

This is the Debian package of the Cairo multi-platform 2D graphics library

Packaged by Dave Beckett <dajobe@debian.org>

It was downloaded from http://cairographics.org/snapshots/

-----------------------------------------------------------------------

Copyright 1999 Tom Tromey
Copyright 2002, 2003 University of Southern California, Information Sciences Institute (ISI)
Copyright 2004 Calum Robinson
Copyright 2004 Richard D. Worth
Copyright 2004, 2005 Red Hat, Inc.

Copyright 2004 David Reveman
Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of David Reveman not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. David Reveman makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

DAVID REVEMAN DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DAVID REVEMAN BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Author: David Reveman <davidr@novell.com>

------------------------------------------------------------------------

Cairo is free software.

Every source file in the implementation of cairo is available to be redistributed and/or modified under the terms of either the GNU Lesser General Public License (LGPL) version 2.1 or the Mozilla Public License (MPL) version 1.1. Some files are available under more liberal terms, but we believe that in all cases, each file may be used under either the LGPL or the MPL.

See the following files in this directory for the precise terms and conditions of either license:

COPYING-LGPL-2.1
COPYING-MPL-1.1

Please see each file in the implementation for Copyright and licensing information.

------------------------------------------------------------------------

On Debian systems, the complete text of the GNU General Lesser Public
License 2.1 can be found in /usr/share/common-licenses/LGPL-2.1

The Mozilla Public License 1.1 (COPYING-MPL-1.1 above) follows:

MOZILLA PUBLIC LICENSE
Version 1.1

-----------

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the
Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to
the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original
Code, prior Modifications used by a Contributor, and the Modifications
made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the
combination of the Original Code and Modifications, in each case
including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally
accepted in the software development community for the electronic
transfer of data.

1.5. "Executable" means Covered Code in any form other than Source
Code.

1.6. "Initial Developer" means the individual or entity identified
as the Initial Developer in the Source Code notice required by Exhibit
A.

1.7. "Larger Work" means a work which combines Covered Code or
portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum
extent possible, whether at the time of the initial grant or
subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the
substance or structure of either the Original Code or any previous
Modifications. When Covered Code is released as a series of files, a
Modification is:
   A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

   B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

   (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and
(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by
3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
(a) Third Party Claims.
If Contributor has knowledge that a license under a third party’s intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will
know whom to contact. If Contributor obtains such knowledge after
the Modification is made available as described in Section 3.2,
Contributor shall promptly modify the LEGAL file in all copies
Contributor makes available thereafter and shall take other steps
(such as notifying appropriate mailing lists or newsgroups)
reasonably calculated to inform those who received the Covered
Code that new knowledge has been obtained.

(b) Contributor APIs.
If Contributor's Modifications include an application programming
interface and Contributor has knowledge of patent licenses which
are reasonably necessary to implement that API, Contributor must
also include this information in the LEGAL file.

(c) Representations.
Contributor represents that, except as disclosed pursuant to
Section 3.4(a) above, Contributor believes that Contributor's
Modifications are Contributor's original creation(s) and/or
Contributor has sufficient rights to grant the rights conveyed by
this License.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source
Code. If it is not possible to put such notice in a particular Source
Code file due to its structure, then You must include such notice in a
location (such as a relevant directory) where a user would be likely
to look for such a notice. If You created one or more Modification(s)
You may add your name as a Contributor to the notice described in
Exhibit A. You must also duplicate this License in any documentation
for the Source Code where You describe recipients' rights or ownership
rights relating to Covered Code. You may choose to offer, and to
charge a fee for, warranty, support, indemnity or liability
obligations to one or more recipients of Covered Code. However, You
may do so only on Your own behalf, and not on behalf of the Initial
Developer or any Contributor. You must make it absolutely clear than
any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial
Developer and every Contributor for any liability incurred by the
Initial Developer or such Contributor as a result of warranty,
support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the
requirements of Section 3.1-3.5 have been met for that Covered Code,
and if You include a notice stating that the Source Code version of
the Covered Code is available under the terms of this License,
including a description of how and where You have fulfilled the
obligations of Section 3.2. The notice must be conspicuously included
in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:
(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF
LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.


11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys’ fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.
Initial Developer may designate portions of the Covered Code as "Multiple-Licensed." "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A - Mozilla Public License.

"The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is _________________________________.

The Initial Developer of the Original Code is _________________________.

Portions created by ______________________ are Copyright (C) ______ _________________. All Rights Reserved.

Contributor(s): _________________________________.

Alternatively, the contents of this file may be used under the terms of the [____] license (the "[____] License"), in which case the provisions of [____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [____] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [____] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

1.1223 @babel/plugin-proposal-async-generator-functions 7.13.8
1.1223.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1224 json 1.8

1.1225 netty-handler 4.0.48.Final

1.1225.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
Open Source Used In DNAC Platform Antelope 4588

Found in path(s):
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

// Try the OpenJDK's proprietary implementation.

Found in path(s):
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/util/SelfSignedCertificate.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/SslCompletionEvent.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/ocsp/package-info.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/SslCloseCompletionEvent.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/OptionalSslHandler.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/ConscryptAlpnSslEngine.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/Ocsp/OcspClientHandler.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/util/BouncyCastleSelfSignedCertGenerator.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/CipherSuiteFilter.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/Java7SslParametersUtils.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/OpenSslServerSessionContext.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNegotiator.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011 The Netty Project
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
*/
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
~

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
~

No license file was found, but licenses were detected in source scan.
Found in path(s):
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/OpenSslCertificateException.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/DelegatingSslContext.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/ReferenceCountedOpenSslServerContext.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/PemX509Certificate.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/flush/package-info.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/ssl/Utils.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/Java8SslUtils.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/flush/FlushConsolidationHandler.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/PemEncoded.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/PemValue.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/OpenSslKeyMaterialManager.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/OpenSslExtendedKeyMaterialManager.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/ReferenceCountedOpenSslEngine.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/PemPrivateKey.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/ReferenceCountedOpenSslHandlerContext.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources.jar/io/netty/handler/ssl/ReferenceCountedOpenSslClientContext.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/traffic/package-info.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/traffic/GlobalTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/traffic/GlobalChannelTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
/*

Found in path(s):
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/package-info.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/ssl/NotSslRecordException.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/logging/package-info.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/timeout/package-info.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/stream/ChunkedInput.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/timeout/IdleState.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/timeout/ReadTimeOutException.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/stream/ChunkedFile.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/logging/LoggingHandler.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/timeout/IdleStateEvent.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/stream/package-info.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/stream/ChunkedWriteHandler.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/stream/ChunkedNioFile.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/stream/ChunkedNioStream.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/logging/LogLevel.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/timeout/TimeoutException.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/timeout/IdleStateEvent.java
* /opt/cola/permits/1202446677_1631568413.44/0/netty-handler-4-0-48-final-sources-jar/io/netty/handler/timeout/WriteTimeOutException.java
*/
1.1226 sway 2.0.6 2.0.6

1.1226.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Apigee Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1227 awaitility 3.0.0

1.1227.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licenser" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1228 image-size 0.7.4
1.1228.1 Available under license:
MIT

1.1229 event-emitter 0.3.5
1.1229.1 Available under license:
Copyright (C) 2012-2015 Mariusz Nowak (www.medikoo.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1230 punycode 2.1.1
1.1230.1 Available under license:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1231 nan 2.14.2
1.1231.1 Available under license :
The MIT License (MIT)

Copyright (c) 2018 NAN contributors

*NAN contributors listed at <https://github.com/nodejs/nan#contributors>*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1232 frameguard 2.0.0 2.0.0
1.1232.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1233 regexpu-core 2.0.0

1.1233.1 Available under license:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1234 escodegen 1.8.1

1.1234.1 Available under license:
Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

* Neither the names of the Mozilla Foundation nor the names of project
  contributors may be used to endorse or promote products derived from this
  software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1235 jsp 2.1
1.1235.1 Available under license:
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
Version 1.0

*  

1. Definitions.

o
1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

   + A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   
   + B. Any new file that contains any part of the Original Software or previous Modification; or
   
   + C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. *Patent Claims* means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. *Source Code* means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. *You* (or ?Your?) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, *You* includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, *control* means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered
Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients’ rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the
terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as ?Participant?) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.
6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly
granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any
distributor) shall survive termination.

* 

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING
NEGligence), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER
CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF
SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR
CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES
FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR
MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH
PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION
OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING
FROM SUCH PARTY?S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH
LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF
INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT
APPLY TO YOU.

* 

8. U.S. GOVERNMENT END USERS.

The Covered Software is a ?commercial item,? as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting
of ?commercial computer software? (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and ?commercial
computer software documentation? as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48
Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and
supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer
software under this License.

* 

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this
License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it
enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the
Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction?s
conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts
located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing
party responsible for costs, including, without limitation, court costs and reasonable attorneys? fees and expenses.
The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly
excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter
shall not apply to this License. You agree that You alone are responsible for compliance with the United States
export administration regulations (and the export control laws and regulation of any other countries) when You use,
distribute or otherwise make available any Covered Software.

*
10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

1.1236 babel-plugin-transform-es2015-arrow-functions 6.22.0 6.22.0

1.1236.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1237 checker-qual 2.10.0

1.1237.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1238 commons-daemon 1.0.13

1.1238.1 Available under license:

Apache Commons Daemon
Copyright 1999-2013 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1239 require-directory 2.1.1

1.1239.1 Available under license:

MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.1240 lazy 1.0.11

1.1240.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1241 path-root 0.1.1

1.1241.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1242 content-disposition 0.5.3
1.1242.1 Available under license:
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1243 forwarded 0.1.2 0.1.2
1.1243.1 Available under license:
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1244 symbol-observable 1.2.0 1.2.0

1.1244.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Copyright (c) Ben Lesh <ben@benlesh.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1245 path-exists 3.0.0
1.1245.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1246 promise 7.3.1 7.3.1
1.1246.1 Available under license:
Copyright (c) 2014 Forbes Lindesay

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1247 requires-port 1.0.0

1.1247.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1248 finalhandler 1.1.1 1.1.1

1.1248.1 Available under license:

(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1249 ical4j 1.0.2

1.1249.1 Available under license:
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2011, Ben Fortuna
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * o Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * o Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 *
 * o Neither the name of Ben Fortuna nor the names of any other contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 * A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

Found in path(s):
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/TzOffsetToTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/data/CalendarBuilderCustomRegistryTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/RepeatTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/LanguageTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/ContactTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/DatePropertyTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/VersionTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/NumberListTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/ComponentTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/CreatedTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/MemberTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/GraphTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/ImLocationTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/NameTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/TzNameTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/LocalityTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/ParentComponentTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/TimeZoneTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/CountryTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/LocationsTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/VersionTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/ImAddressTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/OwnerTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/CompressTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/ComponentsTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/DurTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/DurationTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/TzIdTest.java
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/net/fortuna/ical4j/model/property/FreeBusyTest.java

Open Source Used In DNAC Platform Antelope 4624
jar/net/fortuna/ical4j/model/property/DurationTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/parameter/RoleTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/AbstractPropertyTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/component/AvailableTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/component/VDVenueTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/PeriodListTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/TzUrlTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/CommentTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/CuTypeTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/PartStatTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/AltRepTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/DtListTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/DateListPropertyTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/AddressListTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/DateListTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/AlternateRepTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/RuntimeTest.java
* /opt/cola/permits/1000715208_1614362208.0/*ical4j-1-0-2-test-sources-
jar/net/fortuna/ical4j/model/property/RegexTestHarness.java

Open Source Used In DNAC Platform Antelope 4625
Open Source Used In DNAC Platform Antelope 4626
Open Source Used In DNAC Platform Antelope 4628
No license file was found, but licenses were detected in source scan.
# documentation and/or other materials provided with the distribution.
# o Neither the name of Ben Fortuna nor the names of any other contributors
# without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/log4j.properties
* /opt/cola/permits/1000715208_1614362208.0/0/ical4j-1-0-2-test-sources-jar/ical4j.properties

## 1.1250 jetty-io 9.4.41.v20210516

### 1.1250.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (“AGREEMENT”). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT’S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

“Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

“Contributor” means any person or entity that distributes the Program.

“Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.
"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or
conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose; ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If
that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does
not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.  
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl
These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay
The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1251 helmet-csp 1.2.2 1.2.2
1.1251.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1252 copy-descriptor 0.1.1
1.1252.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1253 send 0.13.2 0.13.2
1.1253.1 Available under license:
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1254 chokidar 1.7.0
1.1254.1 Available under license :

No license file was found, but licenses were detected in source scan.

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the “Software”), to deal
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1127119904_1611787887.02/0/chokidar-1-7-0-2-tgz/package/README.md

1.1255 shebang-command 2.0.0
1.1255.1 Available under license :

MIT License

Copyright (c) Kevin Mrtensson <kevinmartensson@gmail.com> (github.com/kevva)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1256 archiver-utils 1.3.0
1.1256.1 Available under license:

Copyright (c) 2015 Chris Talkington.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1257 semaphore-async-await 1.5.1

1.1257.1 Available under license:

Copyright (c) 2016 Jan Soendermann

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1258 junit-platform-commons 1.7.0

1.1258.1 Available under license:

Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

### 1. Definitions

**Contribution means:**
- **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- **b)** in the case of each subsequent Contributor:
  - **i)** changes to the Program, and
  - **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

**Contributor means** any person or entity that Distributes the Program.

**Licensed Patents** mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

**Program** means the Contributions Distributed in accordance with this Agreement.

**Recipient** means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

**Derivative Works** shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

**Modified Works** shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

**Distribute** means the acts of **a)** distributing or **b)** making available in any manner that enables the transfer of a copy.

**Source Code** means the form of a Program preferred for making modifications, including but not limited to software
source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

### 2. Grant of Rights

**a)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

**b)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

**c)** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

**d)** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

**e)** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

### 3. Requirements

**3.1** If a Contributor Distributes the Program in any form, then:

* **a)** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

* **b)** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
  * **i)** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  * **ii)** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect,
special, incidental and consequential damages, such as lost profits; 
* **iii)** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and 
* **iv)** requires any subsequent distribution of the Program by any party to be under a license that satisfies the 
requirements of this section 3.

**3.2** When the Program is Distributed as Source Code:

* **a)** it must be made available under this Agreement, or if the Program **(i)** is combined with other material 
in a separate file or files made available under a Secondary License, and **(ii)** the initial Contributor attached to 
the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available 
under the terms of such Secondary Licenses, and 
* **b)** a copy of this Agreement must be included with each copy of the Program.

**3.3** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of 
warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they 
Distribute, provided that Contributors may add their own appropriate notices.

### 4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners 
and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who 
includes the Program in a commercial product offering should do so in a manner which does not create potential 
liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, 
such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor 
(Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits 
and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts 
or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial 
product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or 
alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **a)** promptly 
notify the Commercial Contributor in writing of such claim, and **b)** allow the Commercial Contributor to 
control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The 
Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That 
Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or 
offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's 
responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other 
Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay 
any damages as a result, the Commercial Contributor must pay those damages.

### 5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY 
APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR 
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, 
ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR 
FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the 
appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights
under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

### 6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

#### Exhibit A - Form of Secondary Licenses Notice
This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.1259 libjpeg-turbo 6b

1.1259.1 Available under license:

libjpeg-turbo Licenses

libjpeg-turbo is covered by three compatible BSD-style open source licenses:

- The IJG (Independent JPEG Group) License, which is listed in [README.ijg](README.ijg)

  This license applies to the libjpeg API library and associated programs (any code inherited from libjpeg, and any modifications to that code.)

- The Modified (3-clause) BSD License, which is listed below

  This license covers the TurboJPEG API library and associated programs, as well as the build system.

- The zlib License[https://opensource.org/licenses/Zlib](https://opensource.org/licenses/Zlib)

  This license is a subset of the other two, and it covers the libjpeg-turbo SIMD extensions.

Complying with the libjpeg-turbo Licenses
This section provides a roll-up of the libjpeg-turbo licensing terms, to the best of our understanding.

1. If you are distributing a modified version of the libjpeg-turbo source, then:
   
   1. You cannot alter or remove any existing copyright or license notices from the source.

   **Origin**
   - Clause 1 of the IJG License
   - Clause 1 of the Modified BSD License
   - Clauses 1 and 3 of the zlib License

   2. You must add your own copyright notice to the header of each source file you modified, so others can tell that you modified that file (if there is not an existing copyright header in that file, then you can simply add a notice stating that you modified the file.)

   **Origin**
   - Clause 1 of the IJG License
   - Clause 2 of the zlib License

   3. You must include the IJG README file, and you must not alter any of the copyright or license text in that file.

   **Origin**
   - Clause 1 of the IJG License

2. If you are distributing only libjpeg-turbo binaries without the source, or if you are distributing an application that statically links with libjpeg-turbo, then:

   1. Your product documentation must include a message stating:

      This software is based in part on the work of the Independent JPEG Group.

      **Origin**
      - Clause 2 of the IJG license

   2. If your binary distribution includes or uses the TurboJPEG API, then your product documentation must include the text of the Modified BSD License (see below.)

      **Origin**
3. You cannot use the name of the IJG or The libjpeg-turbo Project or the contributors thereof in advertising, publicity, etc.

**Origin**
- IJG License
- Clause 3 of the Modified BSD License

4. The IJG and The libjpeg-turbo Project do not warrant libjpeg-turbo to be free of defects, nor do we accept any liability for undesirable consequences resulting from your use of the software.

**Origin**
- IJG License
- Modified BSD License
- zlib License

The Modified (3-clause) BSD License

Copyright (C)2009-2019 D. R. Commander. All Rights Reserved.
Copyright (C)2015 Viktor Szathmry. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the libjpeg-turbo Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS", AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Why Three Licenses?

The zlib License could have been used instead of the Modified (3-clause) BSD License, and since the IJG License effectively subsumes the distribution conditions of the zlib License, this would have effectively placed libjpeg-turbo binary distributions under the IJG License. However, the IJG License specifically refers to the Independent JPEG Group and does not extend attribution and endorsement protections to other entities. Thus, it was desirable to choose a license that granted us the same protections for new code that were granted to the IJG for code derived from their software.

1.1260 bootstrap 4.3.1 4.3.1

1.1260.1 Available under license:

The MIT License (MIT)

Copyright (c) 2011-2019 Twitter, Inc.
Copyright (c) 2011-2019 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1261 globals 11.7.0

1.1261.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1262 jetty-util 9.3.27.v20190418

1.1263 hadoop-annotations 3.2.2
1.1263.1 Available under license :

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of
org.iq80.levlod:leveldb-api (https://github.com/dain/leveldb), which has the
following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of
AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java),
which has the following notices:

This software includes third party software subject to the following copyrights:
  - XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
  - PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-
    2010 AOL Inc.

The binary distribution of this product bundles binaries of
Gson 2.2.4,
which has the following notices:

The Netty Project

=================
Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit
This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
* https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakar's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.proto.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.

Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:

* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices:

* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
  Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
  TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of "Java Concurrency in Practice" book annotations 1.0, which has the following notices:

* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
  Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of Jetty :: Http Utility 9.3.19,
Jetty :: IO Utility 9.3.19,
Jetty :: Security 9.3.19,
Jetty :: Server Core 9.3.19,
Jetty :: Servlet Handling 9.3.19,
Jetty :: Utilities 9.3.19,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19,
Jetty :: XML utilities 9.3.19,
which has the following notices:

Jetty Web Container
Copyright 1995-2016 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK
If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
  (Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by "GCC RUNtime Library Exception"

== Contributors ==
* Tatu Saloranta
  * Providing benchmark suite
* Alec Wysoker
  * Performance and memory usage improvement

The binary distribution of this product bundles binaries of Xerces2 Java Parser 2.9.1, which has the following notices:
* http://xerces.apache.org/xerces2-j/license-xerces2-j.html
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:
--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.
Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.
http://glyphicons.com/

Copyright Jan Kovak
Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically 'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors
Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library distributed under the terms of the Apache 2 license. Original source copyright: Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5 (https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
  which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1,
which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9
which has the following notices:
* =---------------------------------------------------------------------
  = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
  =---------------------------------------------------------------------

  This product includes software developed by
  Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 *
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
Copyright (c) 2008, 2009, 2010 Massachusetts Institute of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*++
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 * Abstract: The main routine
 */

For src/main/native/src/org/apache/hadoop/io/compress/lz4/[lz4.h,lz4.c,lz4hc.h,lz4hc.c],

/*
LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at:
- LZ4 source repository: http://code.google.com/p/lz4/
- LZ4 public forum: https://groups.google.com/forum/#!forum/lz4c

For hadoop-common-project/hadoop-common/src/main/native/gtest
---------------------------------------------------------------------
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1
---------------------------------------------------------------------
(GO license)
This is a work derived from Russ Cox’s RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

-------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni (https://github.com/fusesource/leveldbjni), which is available under the following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of FuseSource Corp. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

--------------------------------------------------------------------------------

Copyright (C) 2008-2016, SpryMedia Ltd.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

--------------------------------------------------------------------------------

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
em-helpers v0.5.13
em-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodeings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25
--------------------------------------------------------------------------------

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

--------------------------------------------------------------------------------

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this
file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the
License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
express or implied. See the License for the specific language governing permissions and
limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js

--------------------------------------------------------------------------------

The MIT License (MIT)

Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software
and associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT
LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE
OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js
The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery
Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
oj! Algorithms - version 43.0

--------------------------------------------------------------------------------


This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js
--------------------------------------------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js
--------------------------------------------------------------------------------

D3 is available under a 3-clause BSD license. For details, see:
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
HSQLDB Database 2.3.4
--------------------------------------------------------------------------------

(HSQL License)
"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group."

The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to,
   deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions
thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of
the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this
License. If You distribute the Covered Software in Executable form under a
different license, You must make it absolutely clear that any terms which differ
from this License are offered by You alone, not by the Initial Developer or
Contributor. You hereby agree to indemnify the Initial Developer and every
Contributor for any liability incurred by the Initial Developer or such
Contributor as a result of any such terms You offer.

3.6. Larger Works. You
may create a Larger Work by combining Covered Software with other code not
governed by the terms of this License and distribute the Larger Work as a single
product. In such a case, You must make sure the requirements of this License are
fulfilled for the Covered Software.

4. Versions of the License.

4.1.
New Versions. Sun Microsystems, Inc. is the initial license steward and may
publish revised and/or new versions of this License from time to time. Each
version will be given a distinguishing version number. Except as provided in
Section 4.3, no one other than the license steward has the right to modify this
License.

4.2. Effect of New Versions. You may always continue to use,
distribute or otherwise make the Covered Software available under the terms of
the version of the License under which You originally received the Covered
Software. If the Initial Developer includes a notice in the Original Software
prohibiting it from being distributed or otherwise made available under any
subsequent version of the License, You must distribute and make the Covered
Software available under the terms of the version of the License under which You
originally received the Covered Software. Otherwise, You may also choose to use,
distribute or otherwise make the Covered Software available under the terms of
any subsequent version of the License published by the license
steward.

4.3. Modified Versions. When You are an Initial Developer and You
want to create a new license for Your Original Software, You may create and use
a modified version of this License if You: (a) rename the license and remove
any references to the name of the license steward (except to note that the
license differs from this License); and (b) otherwise make it clear that the
license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS,
WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT
LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS,
MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY
COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO
THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:

jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
grizzly-http 2.2.21
grizzly-http-server 2.2.21
grizzly-http-servlet 2.2.21
grizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter
acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof),
either on an unmodified basis, with other Modifications, as Covered Software
and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the
making, using, or selling of Modifications made by that Contributor either alone
and/or in combination with its Contributor Version (or portions of such
combination), to make, use, sell, offer for sale, have made, and/or otherwise
dispose of: (1) Modifications made by that Contributor (or portions thereof);
and (2) the combination of Modifications made by that Contributor with its
Contributor Version (or portions of such combination).
(c) The licenses granted
in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications available to a third
party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2)
for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with
other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or
otherwise make available in Executable form must also be made available in
Source Code form and that Source Code form must be distributed only under the
terms of this License. You must include a copy of this License with every copy
of the Source Code form of the Covered Software You distribute or otherwise make
available. You must inform recipients of any such Covered Software in Executable
form as to how they can obtain such Covered Software in Source Code form in a
reasonable manner on or through a medium customarily used for software
exchange.

3.2. Modifications.
The Modifications that You create or to which
You contribute are governed by the terms of this License. You represent that You
believe Your Modifications are Your original creation(s) and/or You have
sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that
identifies You as the Contributor of the Modification. You may not remove or
alter any copyright, patent or trademark notices contained within the Covered
Software, or any notices of licensing or any descriptive text giving attribution
to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source
Code form that alters or restricts the applicable version of this License or the
recipients' rights hereunder. You may choose to offer, and to charge a fee for,
warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an
Initial Developer and You want to create a new license for Your Original Software. You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.
6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The
application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
By obtaining, using and/or copying this work, you (the licensee) agree that you
have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation,
with or without modification, for any purpose and without fee or royalty is
hereby granted, provided that you include the following on ALL copies of the
software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the
  redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and
  conditions. If none exist, the W3C Software Short Notice should be included
  (hypertext is preferred, text is permitted) within the body of any redistributed
  or derivative code.
- Notice of any changes or modifications to the files, including the date changes
  were made. (We recommend you provide URIs to the location from which the code is
derived.)

The binary distribution of this product bundles these dependencies under the
following license:
JUnit 4.11
Eclipse JDT Core 3.1.1
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of
intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable
law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

(2-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice” book annotations 1.0

(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of
this License.
"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:
You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH
EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the
following license:
  jamon-runtime 2.4.1

(MPL 2.0)
Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation
of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a
Contributor and that particular Contributors Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice
in Exhibit A, the Executable Form of such Source Code Form, and Modifications of
such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to
the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or
earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate
file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the
time of the initial grant or subsequently, any and all of the rights conveyed by
this License.
1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.
2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in
Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.
3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.
6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License
10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If you choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:

JDOM 1.1
Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally

Open Source Used In DNAC Platform Antelope 4717
The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4

--------------------------------------------------------------------------------
This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical
work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this
License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any
exercise by You of the rights granted under this License;
ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied,
and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URL if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with
respect to the Work licensed here. There are no understandings,
agreements or representations with respect to the Work not specified
here. Licensor shall not be bound by any additional provisions that
may appear in any communication from You. This License may not be
modified without the mutual written agreement of the Licensor and You.
f. The rights granted under, and the subject matter referenced, in this
License were drafted utilizing the terminology of the Berne Convention
for the Protection of Literary and Artistic Works (as amended on
September 28, 1979), the Rome Convention of 1961, the WIPO Copyright
Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996
and the Universal Copyright Convention (as revised on July 24, 1971).
These rights and subject matter take effect in the relevant
jurisdiction in which the License terms are sought to be enforced
according to the corresponding provisions of the implementation of
those treaty provisions in the applicable national law. If the
standard suite of rights granted under applicable copyright law
includes additional rights not granted under this License, such
additional rights are deemed to be included in the License; this
License is not intended to restrict the license of any rights under
applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty
whatsoever in connection with the Work. Creative Commons will not be
liable to You or any party on any legal theory for any damages
whatsoever, including without limitation any general, special,
incidental or consequential damages arising in connection to this
license. Notwithstanding the foregoing two (2) sentences, if Creative
Commons has expressly identified itself as the Licensor hereunder, it
shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the
Work is licensed under the CCPL, Creative Commons does not authorize
the use by either party of the trademark "Creative Commons" or any
related trademark or logo of Creative Commons without the prior
written consent of Creative Commons. Any permitted use will be in
compliance with Creative Commons' then-current trademark usage
guidelines, as may be published on its website or otherwise made
available upon request from time to time. For the avoidance of doubt,
this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.

For: hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs
/server/datanode/checker/AbstractFuture.java and
Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1264 commons-validator 1.6
1.1264.1 Available under license :

Apache Commons Validator
Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/). 

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1265 calcite-linq4j 1.26.0

1.1265.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache Calcite
Copyright 2012-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product is based on source code originally developed
by DynamoBI Corporation, LucidEra Inc., SQLstream Inc. and others
under the auspices of the Eigenbase Foundation
and released as the LucidDB project.

The web site includes files generated by Jekyll.

1.1266 ms 0.7.1
1.1266.1 Available under license:
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <rauchg@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1267 @babel/plugin-syntax-numeric-separator 7.10.4

1.1267.1 Available under license :

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1268 cli-boxes 2.2.1

1.1268.1 Available under license:
 MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1269 xsdlib 2013.6.1

1.1269.1 Available under license:
 No license file was found, but licenses were detected in source scan.

2013 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of Oracle nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):

No license file was found, but licenses were detected in source scan.

# Copyright (c) 2001-2013 Oracle and/or its affiliates. All rights reserved.
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
# - Redistributions of source code must retain the above copyright
#   notice, this list of conditions and the following disclaimer.
# - Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in the
#   documentation and/or other materials provided with the distribution.
# - Neither the name of Oracle nor the names of its
#   from this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Messages.properties
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Messages_ja.properties

No license file was found, but licenses were detected in source scan.

/*
 * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
 *
 * Copyright (c) 2001-2013 Oracle and/or its affiliates. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 * met:
 *
 * - Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 *
 * - Redistribution in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in the
 *   documentation and/or other materials provided with the distribution.
 *
 * - Neither the name of Sun Microsystems, Inc. or the names of
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
* 
* This software is provided "AS IS," without a warranty of any kind. ALL
* EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES,
* INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A
* PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND
* ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES OR LIABILITIES
* SUFFERED BY LICENSEE AS A RESULT OF OR RELATING TO USE, MODIFICATION
* OR DISTRIBUTION OF THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL
* SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA,
* OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR
* PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF
* LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE,
* EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*/

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/InternalImpl.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message_fr.properties
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message_ja.properties
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message.properties

No license file was found, but licenses were detected in source scan.

# Copyright (c) 2001-2013 Oracle and/or its affiliates. All rights reserved.
# modification, are permitted provided that the following conditions are
# - Redistributions of source code must retain the above copyright
# notice, this list of conditions and the following disclaimer.
# notice, this list of conditions and the following disclaimer in the
# documentation and/or other materials provided with the distribution.
# this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message_fr.properties
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message_ja.properties
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/message.properties

No license file was found, but licenses were detected in source scan.

/
* The Apache Software License, Version 1.1
* 
* Copyright (c) 1999-2003 The Apache Software Foundation. All rights
* reserved.
* 
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

* 1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

* 2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the
   distribution.

* 3. The end-user documentation included with the redistribution,
   if any, must include the following acknowledgment:
   "This product includes software developed by the
   Apache Software Foundation (http://www.apache.org/)."
   Alternately, this acknowledgment may appear in the software itself,
   if and wherever such third-party acknowledgments normally appear.

* 4. The names "Xerces" and "Apache Software Foundation" must
   not be used to endorse or promote products derived from this
   software without prior written permission. For written
   permission, please contact apache@apache.org.

* 5. Products derived from this software may not be called "Apache",
   nor may "Apache" appear in their name, without prior written
   permission of the Apache Software Foundation.

* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
  WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
  DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
  ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
  SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
  LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
  USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
  ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
  OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
  OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
  SUCH DAMAGE.

*=====================================================================

* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., http://www.apache.org. For more
* information on the Apache Software Foundation, please see
*/
Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/ParserForXMLSchema.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/regexp/RegexParser.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright (c) 2001-2013 Oracle and/or its affiliates. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* *
* - Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* *
* - Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* *
* - Neither the name of Oracle nor the names of its
* contributors may be used to endorse or promote products derived
* from this software without specific prior written permission.
* *
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ShortType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/DateTimeFactory.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/BigTimeDurationValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/NonPositiveIntegerType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2
jar/com/sun/msv/datatype/xsd/MinInclusiveFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/SimpleURType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/XSDatatype.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/WhiteSpaceFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/SerializationContext.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/XSDatatypeImpl.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DataTypeWithLexicalConstraintFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/UnsignedLongType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/Base64BinaryType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/IDType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/GYearMonthType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/IntegerType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/GDayType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ListValueType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DurationType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/ConcreteType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/TokenType.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/DataTypeWithFacet.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/AbstractCalendarParser.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/CalendarFormatter.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/TimeZone.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/CalendarParser.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/TimeZone.java
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-jar/com/sun/msv/datatype/xsd/datetime/TimeZone.java
No license file was found, but licenses were detected in source scan.

/*
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 1999-2013 The Apache Software Foundation. All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Xerces" and "Apache Software Foundation" must
 *    not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 *    nor may "Apache" appear in their name, without prior written
 *    permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.

* ====================================================================

* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., http://www.apache.org. For more
* information on the Apache Software Foundation, please see

*/

Found in path(s):
* /opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2-
* jar/com/sun/msv/datatype/regexp/RangeToken.java

No license file was found, but licenses were detected in source scan.

/*
* The Apache Software License, Version 1.1
*
*
* Copyright (c) 1999-2002 The Apache Software Foundation. All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* *
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* *
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
* *
* 3. The end-user documentation included with the redistribution,
*    if any, must include the following acknowledgment:
*    "This product includes software developed by the
*    Apache Software Foundation (http://www.apache.org/)."

Alternatively, this acknowledgment may appear in the software itself,
* if and wherever such third-party acknowledgments normally appear.
4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, International Business Machines, Inc., http://www.apache.org. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Found in path(s):
/opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2.jar/com/sun/msv/datatype/regexp/Token.java
/opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2.jar/com/sun/msv/datatype/regexp/ParseException.java
/opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2.jar/com/sun/msv/datatype/regexp/RegularExpression.java
/opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2.jar/com/sun/msv/datatype/regexp/Match.java
/opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2.jar/com/sun/msv/datatype/regexp/REUtil.java
/opt/cola/permits/1177942015_1625571932.43/0/xsdlib-2013-6-1-sources-2.jar/com/sun/msv/datatype/regexp/BMPattern.java

Open Source Used In DNAC Platform Antelope 4746
1.1270 react-ace 5.10.0 5.10.0

1.1270.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 James Hrisho

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1271 jackson-module-paranamer 2.10.5

1.1271.1 Available under license:

This copy of Jackson JSON processor `jackson-module-paranamer` module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing
Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

### 1.1272 byte-buddy-agent 1.9.3

#### 1.1272.1 Available under license:

No license file was found, but licenses were detected in source scan.

```java
/*
 * Copyright 2014 - 2018 Rafael Winterhalter
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):
- /opt/cola/permits/1114503625_1607518978.69/0/byte-buddy-agent-1-9-3-sources-1-jar/net/bytebuddy/agent/Installer.java
- /opt/cola/permits/1114503625_1607518978.69/0/byte-buddy-agent-1-9-3-sources-1-jar/net/bytebuddy/agent/package-info.java
- /opt/cola/permits/1114503625_1607518978.69/0/byte-buddy-agent-1-9-3-sources-1-jar/net/bytebuddy/agent/Attacher.java
- /opt/cola/permits/1114503625_1607518978.69/0/byte-buddy-agent-1-9-3-sources-1-jar/net/bytebuddy/agent/ByteBuddyAgent.java
- /opt/cola/permits/1114503625_1607518978.69/0/byte-buddy-agent-1-9-3-sources-1-jar/net/bytebuddy/agent/VirtualMachine.java
Apache HBase - Shaded Protocol
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1274 @babel/plugin-transform-modules-systemjs 7.13.8

1.1274.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.1275 velocity 2.0

1.1275.1 Available under license:

Apache Velocity - Engine
Copyright 2000-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1276 babel-core 6.26.3 6.26.3

1.1276.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1277 websocket-servlet 9.3.27.v20190418

1.1278 swagger-models 2.0.7

1.1278.1 Available under license:

```java
/**
   * Copyright 2017 SmartBear Software
   * <p>
   * Licensed under the Apache License, Version 2.0 (the "License");
   * you may not use this file except in compliance with the License.
   * You may obtain a copy of the License at
   * <p>
   * http://www.apache.org/licenses/LICENSE-2.0
   * <p>
   * Unless required by applicable law or agreed to in writing, software
   * distributed under the License is distributed on an "AS IS" BASIS,
   * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   * See the License for the specific language governing permissions and
   * limitations under the License.
   */

package io.swagger.v3.oas.models.info;

import java.util.Objects;

/**
   * License
   * <p>
   * @see "https://github.com/OAI/OpenAPI-Specification/blob/3.0.1/versions/3.0.1.md#licenseObject"
   */

public class License {
    private String name = null;
    private String url = null;
    private java.util.Map<String, Object> extensions = null;

    /**
     * returns the name property from a License instance.
     *
     * @return String name
     */
    ```
public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public License name(String name) {
    this.name = name;
    return this;
}

/**
 * returns the url property from a License instance.
 *
 * @return String url
 ***/

public String getUrl() {
    return url;
}

public void setUrl(String url) {
    this.url = url;
}

public License url(String url) {
    this.url = url;
    return this;
}

@Override
public boolean equals(java.lang.Object o) {
    if (this == o) {
        return true;
    }
    if (o == null || getClass() != o.getClass()) {
        return false;
    }
    License license = (License) o;
    return Objects.equals(this.name, license.name) &&
            Objects.equals(this.url, license.url) &&
            Objects.equals(this.extensions, license.extensions);
}

@Override
public int hashCode() {
    return Objects.hash(name, url, extensions);
}

public java.util.Map<String, Object> getExtensions() {
    return extensions;
}

public void addExtension(String name, Object value) {
    if (name == null || name.isEmpty() || !name.startsWith("x-")) {
        return;
    }
    if (this.extensions == null) {
        this.extensions = new java.util.LinkedHashMap<>();
    }
    this.extensions.put(name, value);
}

public void setExtensions(java.util.Map<String, Object> extensions) {
    this.extensions = extensions;
}

public License extensions(java.util.Map<String, Object> extensions) {
    this.extensions = extensions;
    return this;
}

@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("class License {\n    name: ").append(toIndentedString(name)).append("\n    url: ").append(toIndentedString(url)).append("\n    \}\n    return sb.toString();
}

/**
 * Convert the given object to string with each line indented by 4 spaces
 * (except the first line).
 */
private String toIndentedString(java.lang.Object o) {
    if (o == null) {
        return "null";
    }
    return o.toString().replace("\n", "\n    ");
}
1.1279 hbase-hadoop-compat 2.4.2

1.1279.1 Available under license:

Apache HBase - Hadoop Two Compatibility
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1280 babel-helpers 6.24.1
1.1280.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
  1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1281 jsreport-import-export 2.1.1

1.1281.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1282 fast-deep-equal 3.1.3 3.1.3
1.1282.1 Available under license :
MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1283 utile 0.3.0
1.1283.1 Available under license :
Copyright (c) 2011, Charlie Robbins & the Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1284 babel-plugin-transform-es2015-function-name 6.24.1

1.1284.1 Available under license:

Copyright (C) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1285 snappy-java 1.1.2.6

1.1286 contains-path 0.1.0

1.1286.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
1.1287 etag 1.7.0 1.7.0

1.1287.1 Available under license:
(The MIT License)

Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1288 traverse 0.6.6 0.6.6

1.1288.1 Available under license:

Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:
http://www.opensource.org/licenses/mit-license.php

Copyright 2010 James Halliday (mail@substack.net)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1289 htmlparser2 3.9.2 3.9.2
1.1289.1 Available under license:
Copyright 2010, 2011, Chris Winberry <chris@winberry.net>. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1290 sanitize-html-react 1.13.0 1.13.0
1.1290.1 Available under license:
Copyright (c) 2013, 2014, 2015 P'unk Avenue LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1291 object-component 0.0.3
1.1291.1 Available under license:
The MIT License (MIT)
Copyright (c) 2012 Mike Ihbe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1292 commons-compiler 3.0.11

1.1293 global-prefix 1.0.2
1.1293.1 Available under license:
The MIT License (MIT)
Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1294 papa-parse 5.1.1 5.1.1

1.1294.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Matthew Holt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1295 hikaricp-java7 2.4.13

1.1296 expat 2.1.0 7ubuntu0.16.04.5
1.1296.1 Available under license:

Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper
Copyright (c) 2001-2017 Expat maintainers

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1297 commons-digester 1.8

1.1298 form-data 2.3.3

1.1298.1 Available under license:

Copyright (c) 2012 Felix Geisendrfer (felix@debuggable.com) and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1299 bluebird 3.5.1 3.5.1
1.1299.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013-2017 Petka Antonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1300 sac 1.3
1.1300.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 1999 World Wide Web Consortium,
 * (Massachusetts Institute of Technology, Institut National de
 * Recherche en Informatique et en Automatique, Keio University). All
 * Rights Reserved. This program is distributed under the W3C's Software
 * Intellectual Property License. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even
 * the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See W3C License http://www.w3.org/Consortium/Legal/ for more details.
 *
 * $Id: ElementSelector.java,v 1.6 2000/02/14 15:54:12 plehegar Exp $

Open Source Used In DNAC Platform Antelope

/*

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/ElementSelector.java
No license file was found, but licenses were detected in source scan.

*/

* (c) COPYRIGHT 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
*
* $Id: SACMediaList.java,v 1.1 2000/02/16 21:27:32 plehegar Exp $
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/SACMediaList.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (c) 1999 World Wide Web Consortium,
* (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. This program is distributed under the W3C's Software Intellectual Property License. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
* See W3C License http://www.w3.org/Consortium/Legal/ for more details.
*
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/Selector.java
No license file was found, but licenses were detected in source scan.

/*

* Copyright (c) 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
* The original version of this interface comes from SAX :
* http://www.megginson.com/SAX/
*
* $Id: ErrorHandler.java,v 1.2 1999/09/26 10:10:34 plehegar Exp $
*/
Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/ErrorHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1999 World Wide Web Consortium,
* (Massachusetts Institute of Technology, Institut National de
* Recherche en Informatique et en Automatique, Keio University). All
* Rights Reserved. This program is distributed under the W3C's Software
* Intellectual Property License. This program is distributed in the
* hope that it will be useful, but WITHOUT ANY WARRANTY; without even
* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
* PURPOSE.
* See W3C License http://www.w3.org/Consortium/Legal/ for more details.
*
* $Id: ParserFactory.java,v 1.1 2000/02/14 15:54:49 plehegar Exp $
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/helpers/ParserFactory.java
No license file was found, but licenses were detected in source scan.

/*
* (c) COPYRIGHT 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
*
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/DescendantSelector.java
No license file was found, but licenses were detected in source scan.

/*
* (c) COPYRIGHT 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
*
* $Id: LangCondition.java,v 1.1 1999/09/26 09:54:37 plehegar Exp $
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/LangCondition.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 1999 World Wide Web Consortium
 * (Massachusetts Institute of Technology, Institut National de Recherche
 * en Informatique et en Automatique, Keio University).
 * All Rights Reserved. http://www.w3.org/Consortium/Legal/
 *
 * The original version of this interface comes from SAX :
 * http://www.megginson.com/SAX/
 *
 * $Id: CSSException.java,v 1.3 2002/06/17 14:09:36 plehegar Exp $
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/CSSException.java
No license file was found, but licenses were detected in source scan.

/*
 * (c) COPYRIGHT 1999 World Wide Web Consortium
 * (Massachusetts Institute of Technology, Institut National de Recherche
 * en Informatique et en Automatique, Keio University).
 * All Rights Reserved. http://www.w3.org/Consortium/Legal/
 *
 * $Id: CombinatorCondition.java,v 1.3 2000/03/08 20:55:41 plehegar Exp $
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/CombinatorCondition.java
No license file was found, but licenses were detected in source scan.

;/*
 * Copyright (c) 1999 World Wide Web Consortium,
 * (Massachusetts Institute of Technology, Institut National de
 * Recherche en Informatique et en Automatique, Keio University). All
 * Rights Reserved. This program is distributed under the W3C's Software
 * Intellectual Property License. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even
 * the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See W3C License http://www.w3.org/Consortium/Legal/ for more details.
 *
 * $Id: ConditionFactory.java,v 1.2 2000/02/15 02:07:34 plehegar Exp $
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/ConditionFactory.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 1999 World Wide Web Consortium
 * (Massachusetts Institute of Technology, Institut National de Recherche
 * en Informatique et en Automatique, Keio University).
 * All Rights Reserved. http://www.w3.org/Consortium/Legal/
 *
 * The original version of this interface comes from SAX :
 * http://www.megginson.com/SAX/
 *
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/Parser.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 1999 World Wide Web Consortium
 * (Massachusetts Institute of Technology, Institut National de Recherche
 * en Informatique et en Automatique, Keio University).
 * All Rights Reserved. http://www.w3.org/Consortium/Legal/
 *
 * $Id: SelectorList.java,v 1.1 1999/09/26 10:06:45 plehegar Exp $
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/SelectorList.java
No license file was found, but licenses were detected in source scan.

/*
 * (c) COPYRIGHT 1999 World Wide Web Consortium
 * (Massachusetts Institute of Technology, Institut National de Recherche
 * en Informatique et en Automatique, Keio University).
 * All Rights Reserved. http://www.w3.org/Consortium/Legal/
 *
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/NegativeSelector.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 1999 World Wide Web Consortium
 * (Massachusetts Institute of Technology, Institut National de Recherche
 * en Informatique et en Automatique, Keio University).
 */
All Rights Reserved. http://www.w3.org/Consortium/Legal/

The original version of this interface comes from SAX:
http://www.megginson.com/SAX/

$Id: Locator.java,v 1.1 1999/09/26 09:58:46 plehegar Exp$
/

Found in path(s):
$ /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/Locator.java$
No license file was found, but licenses were detected in source scan.

/*
(c) COPYRIGHT 1999 World Wide Web Consortium
(Massachusetts Institute of Technology, Institut National de Recherche
en Informatique et en Automatique, Keio University).
All Rights Reserved. http://www.w3.org/Consortium/Legal/
*/

$Id: SimpleSelector.java,v 1.2 1999/09/30 16:54:22 plehegar Exp$
/

Found in path(s):
$ /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/SimpleSelector.java$
No license file was found, but licenses were detected in source scan.

/*
(c) COPYRIGHT 1999 World Wide Web Consortium
(Massachusetts Institute of Technology, Institut National de Recherche
en Informatique et en Automatique, Keio University).
All Rights Reserved. http://www.w3.org/Consortium/Legal/
*/

$Id: PositionalCondition.java,v 1.4 2000/02/14 15:54:12 plehegar Exp$
/

Found in path(s):
$ /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/PositionalCondition.java$
No license file was found, but licenses were detected in source scan.

/*
(c) COPYRIGHT 1999 World Wide Web Consortium
(Massachusetts Institute of Technology, Institut National de Recherche
en Informatique et en Automatique, Keio University).
All Rights Reserved. http://www.w3.org/Consortium/Legal/
*/

/

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
* (c) COPYRIGHT 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
*
* $Id: ContentCondition.java,v 1.1 1999/09/26 09:54:37 plehegar Exp $
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/ContentCondition.java
No license file was found, but licenses were detected in source scan.

/*
* (c) COPYRIGHT 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
*
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/ProcessingInstructionSelector.java
No license file was found, but licenses were detected in source scan.

/*
* (c) COPYRIGHT 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
*
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/ConditionalSelector.java
No license file was found, but licenses were detected in source scan.

/*
Copyright (c) 1999 World Wide Web Consortium,
(Massachusetts Institute of Technology, Institut National de
Recherche en Informatique et en Automatique, Keio University). All
Rights Reserved. This program is distributed under the W3C's Software
* Intellectual Property License. This program is distributed in the
* hope that it will be useful, but WITHOUT ANY WARRANTY; without even
* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
* PURPOSE.
* See W3C License http://www.w3.org/Consortium/Legal/ for more details.
*
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/SelectorFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
*
* The original version of this interface comes from SAX:
* http://www.megginson.com/SAX/
*
* $Id: InputSource.java,v 1.2 1999/09/26 10:09:48 plehegar Exp $
*/

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/InputSource.java
No license file was found, but licenses were detected in source scan.

Keio University). All Rights Reserved.</h3>
University</a>). All Rights Reserved. http://www.w3.org/Consortium/Legal/</h3>
<p>Permission to use, copy, and modify this software and its documentation,
University</a>). All Rights Reserved.
<p>The name and trademarks of copyright holders may NOT be used in advertising

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/META-INF/COPYRIGHT.html
No license file was found, but licenses were detected in source scan.

/*/ 

* (c) COPYRIGHT 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
*
* $Id: CharacterDataSelector.java,v 1.3 1999/09/26 10:05:32 plehegar Exp $
*/
/*
* Copyright (c) 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* et en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
* The original version of this interface comes from SAX :
* http://www.megginson.com/SAX/
* $Id: CSSParseException.java,v 1.3 2000/02/15 02:07:34 plehegar Exp $
*/

/*
* (c) COPYRIGHT 1999 World Wide Web Consortium
* (Massachusetts Institute of Technology, Institut National de Recherche
* et en Informatique et en Automatique, Keio University).
* All Rights Reserved. http://www.w3.org/Consortium/Legal/
* $Id: NegativeCondition.java,v 1.2 1999/09/26 10:15:58 plehegar Exp $
*/

/*
* Copyright (c) 1999 World Wide Web Consortium,
* (Massachusetts Institute of Technology, Institut National de
* Recherche en Informatique et en Automatique, Keio University). All
* Rights Reserved. This program is distributed under the W3C's Software
* Intellectual Property License. This program is distributed in the
* hope that it will be useful, but WITHOUT ANY WARRANTY; without even
* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
* PURPOSE.
* See W3C License http://www.w3.org/Consortium/Legal/ for more details.
* $Id: Condition.java,v 1.8 2000/02/14 15:54:12 plehegar Exp $
*/
Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/Condition.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 1999 World Wide Web Consortium,
 * (Massachusetts Institute of Technology, Institut National de
 * Recherche en Informatique et en Automatique, Keio University). All
 * Rights Reserved. This program is distributed under the W3C's Software
 * Intellectual Property License. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even
 * the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See W3C License http://www.w3.org/Consortium/Legal/ for more details.
 *
 * $Id: AttributeCondition.java,v 1.5 2002/06/17 14:10:09 plehegar Exp $
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/AttributeCondition.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 1999 World Wide Web Consortium
 * (Massachusetts Institute of Technology, Institut National de Recherche
 * en Informatique et en Automatique, Keio University).
 * All Rights Reserved. http://www.w3.org/Consortium/Legal/
 *
 * $Id: LexicalUnit.java,v 1.9 2000/03/08 20:55:33 plehegar Exp $
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/LexicalUnit.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 1999 World Wide Web Consortium
 * (Massachusetts Institute of Technology, Institut National de Recherche
 * en Informatique et en Automatique, Keio University).
 * All Rights Reserved. http://www.w3.org/Consortium/Legal/
 *
 * $Id: DocumentHandler.java,v 1.4 2000/02/16 21:29:35 plehegar Exp $
 */

Found in path(s):
* /opt/cola/permits/1135936139_1613638474.83/0/sac-1-3-sources-3-jar/org/w3c/css/sac/DocumentHandler.java

Open Source Used In DNAC Platform Antelope 4789
1.1301 lodash-max 4.0.1

1.1301.1 Available under license:
Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1302 inquirer 0.12.0
1.1302.1 Available under license :
No license file was found, but licenses were detected in source scan.

Licensed under the MIT license.

Found in path(s):
* /opt/cola/permits/1162566323_1620862018.21/0/inquirer-0-12-0-4-tgz/package/README.md

1.1303 rimraf 2.6.3
1.1303.1 Available under license :
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1304 reap2 1.0.1
1.1304.1 Available under license :
MIT

1.1305 to-regex 3.0.2
1.1305.1 Available under license :
The MIT License (MIT)

Copyright (c) 2016-2018, Jon Schlinkert.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1306 test-exclude 6.0.0

1.1306.1 Available under license :
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1307 to-utf8 0.0.1

1.1307.1 Available under license :
MIT

1.1308 log4j-api 2.14.0
Apache Log4j API
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1309 upath 1.2.0

1.1309.1 Available under license:

Copyright(c) 2014-2019 Angelos Pikoulas (agelos.pikoulas@gmail.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1310 argparse 0.7.0

1.1311 debug 4.3.2

1.1311.1 Available under license:
(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software
and associated documentation files (the 'Software'), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT
LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1312 async 2.6.0

1.1312.1 Available under license:
Copyright (c) 2010-2017 Caolan McMahon

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

# License

Minami is free software, licensed under the Apache License, Version 2.0 (the
"License"). Commercial and non-commercial use are permitted in compliance with
the License.

Copyright (c) 2014-2015 Nijiko Yonskai <nijikokun@gmail.com> and the
[contributors to Minami](https://github.com/Nijikokun/minami/graphs/contributors).
All rights reserved.

You may obtain a copy of the License at:
http://www.apache.org/licenses/LICENSE-2.0

In addition, a copy of the License is included with this distribution.

As stated in Section 7, "Disclaimer of Warranty," of the License:

> Licensor provides the Work (and each Contributor provides its Contributions)
> on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
> express or implied, including, without limitation, any warranties or
> conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
> PARTICULAR PURPOSE. You are solely responsible for determining the
> appropriateness of using or redistributing the Work and assume any risks
> associated with Your exercise of permissions under this License.

The source code for JSDoc 3 is available at:
https://github.com/Nijikokun/minami

# Third-Party Software

Minami includes or depends upon the following third-party software, either in
whole or in part. Each third-party software package is provided under its own
license.

## JSDoc 3

JSDoc 3 is free software, licensed under the Apache License, Version 2.0 (the
"License"). Commercial and non-commercial use are permitted in compliance with
the License.

Copyright (c) 2011-2015 Michael Mathews <micmath@gmail.com> and the
All rights reserved.

You may obtain a copy of the License at:
http://www.apache.org/licenses/LICENSE-2.0

In addition, a copy of the License is included with this distribution.

As stated in Section 7, "Disclaimer of Warranty," of the License:

> Licensor provides the Work (and each Contributor provides its Contributions)
> on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
> express or implied, including, without limitation, any warranties or
> conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
> PARTICULAR PURPOSE. You are solely responsible for determining the
> appropriateness of using or redistributing the Work and assume any risks
> associated with Your exercise of permissions under this License.

The source code for JSDoc 3 is available at:
https://github.com/jsdoc3/jsdoc

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1313 source-map 0.6.1
1.1313.1 Available under license:

Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the names of the Mozilla Foundation nor the names of project contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1314 mute-stream 0.0.5
1.1314.1 Available under license :
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1315 to-buffer 1.1.1
1.1315.1 Available under license :
The MIT License (MIT)

Copyright (c) 2016 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1316 babel-plugin-transform-es2015-typeof-symbol 6.23.0 6.23.0
1.1316.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1317 atk 2.18.1-2ubuntu1
1.1317.1 Available under license:
GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.
Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
1.1318 serve-static 1.13.2 1.13.2
1.1318.1 Available under license :
(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1319 boxen 4.2.0
1.1319.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1320 jersey-container-jetty-http 2.25.1
1.1320.1 Available under license :

Found license ‘General Public License 2.0’ in * Copyright (c) 2011-2013 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only (“GPL”) or the Common Development * and Distribution License(“CDDL”) (collectively, the ”License”). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the ”Classpath” * exception as provided by Oracle in the GPL Version 2 section of the License’

Found license ‘General Public License 2.0’ in # Copyright (c) 2013-2014 Oracle and/or its affiliates. All rights reserved. # The contents of this file are subject to the terms of either the GNU # General Public License Version 2 only (“GPL”) or the Common Development # and Distribution License(“CDDL”) (collectively, the ”License”). You # may not use this file except in compliance with the License. You can # Oracle designates this particular file as subject to the ”Classpath” # exception as provided by Oracle in the GPL Version 2 section of the License’

Found license ‘General Public License 2.0’ in * Copyright (c) 2013-2014 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only (“GPL”) or the Common Development * and Distribution License(“CDDL”) (collectively, the ”License”). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the ”Classpath” * exception as provided by Oracle in the GPL Version 2 section of the License’

Found license ‘General Public License 2.0’ in * Copyright (c) 2013-2015 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only (“GPL”) or the Common Development * and Distribution License(“CDDL”) (collectively, the ”License”). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the ”Classpath” * exception as provided by Oracle in the GPL Version 2 section of the License’

1.1321 parseqs 0.0.5 0.0.5
1.1321.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1322 isarray 1.0.0
1.1322.1 Available under license :
No license file was found, but licenses were detected in source scan.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1110632783_1607450976.71/0/isarray-1-0-0-7-tgz/package/README.md

1.1323 find-up 4.1.0
1.1323.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1324 parquet-hadoop-bundle 1.10.0
1.1324.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
    distribution, then any Derivative Works that You distribute must
    include a readable copy of the attribution notices contained
    within such NOTICE file, excluding those notices that do not
    pertain to any part of the Derivative Works, in at least one
    of the following places: within a NOTICE text file distributed
    as part of the Derivative Works; within the Source form or
    documentation, if provided along with the Derivative Works; or,
    within a display generated by the Derivative Works, if and
    wherever such third-party notices normally appear. The contents
    of the NOTICE file are for informational purposes only and
    do not modify the License. You may add Your own attribution
    notices within Derivative Works that You distribute, alongside
    or as an addendum to the NOTICE text from the Work, provided
    that such additional attribution notices cannot be construed
    as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, \ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This product depends on Apache Thrift and includes it in this binary artifact.

Copyright: 2006-2010 The Apache Software Foundation.
Home page: https://thrift.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0
This product depends on SLF4J and includes SLF4J in this binary artifact. SLF4J is a simple logging facade for Java.

Copyright: 2004-2013 QOS.ch
Home page: http://www.slf4j.org/
License: http://slf4j.org/license.html (MIT license)

The following is the SLF4J license (MIT):

Copyright (c) 2004-2013 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

------------------------------------------------------------------------

This project includes code from Daniel Lemire's JavaFastPFOR project in this
binary artifact. The "Lemire" bit packing classes produced by parquet-generator
are derived from the JavaFastPFOR project.

Copyright: 2013 Daniel Lemire
Home page: http://lemire.me/en/
Project page: https://github.com/lemire/JavaFastPFOR
License: Apache License Version 2.0 http://www.apache.org/licenses/LICENSE-2.0

------------------------------------------------------------------------

This product depends on fastutil and includes it in this binary artifact.
Fastutil provides type-specific collection implementations.

Copyright: 2002-2014 Sebastiano Vigna
This product depends on Jackson and includes it in this binary artifact. Jackson is a high-performance JSON processor.

Copyright: 2007-2015 Tatu Saloranta and other contributors
Home page: http://jackson.codehaus.org/
License: http://www.apache.org/licenses/LICENSE-2.0.txt

1.1325 groovy-macro 3.0.8
1.1325.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * 
 *    http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1218309901_1634925743.24/0/groovy-macro-3-0-8-sources-jar/org/codehaus/groovy/macro/runtime/MacroContext.java
* /opt/cola/permits/1218309901_1634925743.24/0/groovy-macro-3-0-8-sources-jar/org/codehaus/groovy/macro/transform/MacroClass.java
* /opt/cola/permits/1218309901_1634925743.24/0/groovy-macro-3-0-8-sources-jar/org/codehaus/groovy/macro/runtime/MacroBuilder.java
* /opt/cola/permits/1218309901_1634925743.24/0/groovy-macro-3-0-8-sources-jar/org/codehaus/groovy/macro/matcher/ASTNodePredicate.java
* /opt/cola/permits/1218309901_1634925743.24/0/groovy-macro-3-0-8-sources-jar/org/codehaus/groovy/macro/matcher/ContextualClassCodeVisitor.java
No license file was found, but licenses were detected in source scan.

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
1.1327 regjsparsaer 0.6.9
1.1327.1 Available under license:
Copyright (c) Julian Viereck and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1328 profiler 1.0.2

1.1329 scheduler 0.19.1 0.19.1
1.1329.1 Available under license:
MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1330 resolve-from 1.0.1

1.1330.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1331 jaeger-core 0.23.0

1.1331.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was
submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing,
shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
1.1332 passport-strategy 1.0.0

1.1332.1 Available under license:
(The MIT License)

Copyright (c) 2011-2013 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1333 websocket-server 9.4.17.v20190418

1.1333.1 Available under license:

This program and the accompanying materials are made available under the
terms of the Eclipse Public License 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0, or the Apache Software License
2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution.

No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses
to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION
Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY
CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.
This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed.
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message
The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

- org.mortbay.jasper:apache-jsp
- org.apache.tomcat:tomcat-jasper
- org.apache.tomcat:tomcat-juli
- org.apache.tomcat:tomcat-jsp-api
- org.apache.tomcat:tomcat-el-api
- org.apache.tomcat:tomcat-jasper-el
- org.apache.tomcat:tomcat-api
- org.apache.tomcat:tomcat-util-scan
- org.apache.tomcat:tomcat-util
- org.mortbay.jasper:apache-el
- org.apache.tomcat:tomcat-jasper-el
- org.apache.tomcat:tomcat-el-api

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

- org.eclipse.jetty.toolchain:jetty-schemas

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1334 jsreport-base 2.0.2
1.1334.1 Available under license:
   MIT License
   Copyright (c) 2016 jsreport

Open Source Used In DNAC Platform Antelope 4847
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"); to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1335 has-cors 1.1.0 1.1.0

1.1335.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"); to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1336 copy-to-clipboard 3.3.1

1.1336.1 Available under license:

MIT License

Copyright (c) 2017 sudodoki <smd.deluzion@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1337 protostuff-api 1.7.1

1.1337.1 Available under license:

No license file was found, but licenses were detected in source scan.

//Licensed under the Apache License, Version 2.0 (the "License");
//you may not use this file except in compliance with the License.
//You may obtain a copy of the License at
//http://www.apache.org/licenses/LICENSE-2.0
//distributed under the License is distributed on an "AS IS" BASIS,
//Copyright 2008 Google Inc. All rights reserved.
//Redistribution and use in source and binary forms, with or without
//modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
Found in path(s):
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/EnumLite.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/UninitializedMessageException.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/ByteString.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/WireFormat.java
No license file was found, but licenses were detected in source scan.

//Licensed under the Apache License, Version 2.0 (the "License");
//you may not use this file except in compliance with the License.
//You may obtain a copy of the License at
//http://www.apache.org/licenses/LICENSE-2.0
//distributed under the License is distributed on an "AS IS" BASIS,
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/UnsignedNumberUtil.java
No license file was found, but licenses were detected in source scan.

//All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are met:
// 1. Redistributions of source code must retain the above copyright notice,
// this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright notice,
// this list of conditions and the following disclaimer in the documentation
// 3. Neither the name of protostuff nor the names of its contributors may be used

Found in path(s):
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/Tag.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/Exclude.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/Morph.java
No license file was found, but licenses were detected in source scan.

// Copyright 2008 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/ListAdapter.java

No license file was found, but licenses were detected in source scan.

//Licensed under the Apache License, Version 2.0 (the "License);
//you may not use this file except in compliance with the License.
//You may obtain a copy of the License at
//http://www.apache.org/licenses/LICENSE-2.0
//distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/B64Code.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/FilterOutput.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/NumberParser.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/WriteSession.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/FilterInput.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/IntSerializer.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/StatefulOutput.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/LinkedBuffer.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/Input.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/Message.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/CustomSchema.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/WriteSink.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/ProtostuffException.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/Output.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/GraphInput.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/Schema.java
* /opt/cola/permits/1202446559_1631568036.94/0/protostuff-api-1-7-1-sources-jar/io/protostuff/Pipe.java

1.1338 influx 5.0.7 5.0.7
1.1338.1 Available under license:
MIT License

Copyright (c) 2016 Connor Peet

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1339 readable-stream 1.1.14 1.1.14

1.1339.1 Available under license:

Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such
Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
   Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:
   a) promptly notify the Commercial Contributor in writing of such claim, and
   b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.
5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==============================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and
* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.  
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.  
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2
The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----
Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1341 rechoir 0.6.2

1.1341.1 Available under license:

Copyright (c) 2015 Tyler Kellen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1342 p-try 2.2.0 2.2.0

1.1342.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1343 concat-stream 1.6.2 1.6.2

1.1343.1 Available under license:

The MIT License

Copyright (c) 2013 Max Ogden

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1344 jsreport-pdf-utils 1.10.1

1.1344.1 Available under license:

MIT License

Copyright (c) 2017 jsreport
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1345 jersey-client 2.25.1
1.1345.1 Available under license:
    CDDL+GPL License
    http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.1346 depd 1.1.0
1.1346.1 Available under license:
    (The MIT License)

Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1347 colorette 1.2.2

1.1347.1 Available under license:
Copyright Jorge Bucaran <<https://jorgebucaran.com>>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1348 aircompressor 0.10

1.1349 ini 1.3.5

1.1349.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.1350 babel-helper-create-class-features-plugin 7.13.11

1.1350.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1351 js-string-escape 1.0.1

1.1351.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013 Jo Liss

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1352 trim-right 1.0.1
1.1352.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1353 joi 6.10.1
1.1353.1 Available under license :
Copyright (c) 2012-2014, Walmart and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTIONS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/joi/graphs/contributors

1.1354 babel-helper-get-function-arity 6.24.1

6.24.1

1.1354.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1355 is-absolute-url 2.1.0

1.1355.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1356 libthai 0.1.24 2

1.1356.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.
Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your
9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library,
if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: libthai
Upstream-Contact: Theppitak Karoonboonyanan <thep@linux.thai.net>
Source: ftp://linux.thai.net/pub/thailinux/software/libthai/

Files: *
Copyright: 2001-2013 Theppitak Karoonboonyanan <thep@linux.thai.net>,
          Pattara Kiatisivi <ott@linux.thai.net>,
          Vuthichai Ampornaramveth <vuthi@nii.ac.jp>,
          Poonlap Veerathanabutr <poonlap@linux.thai.net>,
          Chanop Silpa-Anan <chanop@debian.org>
License: LGPL-2.1+
This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.

On Debian systems, the complete text of the GNU Lesser General Public License version 2.1 can be found in `/usr/share/common-licenses/LGPL-2.1'.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

On Debian systems, the complete text of the GNU General Public License version 2 can be found in `/usr/share/common-licenses/GPL-2`.

1.1357 lodash-repeat 4.1.0

1.1357.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1358 bunyan-format 0.2.1 0.2.1
1.1358.1 Available under license:

Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1359 jws 3.1.5 3.1.5
1.1359.1 Available under license:
Copyright (c) 2013 Brian J. Brennan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1360 node-forge 0.10.0

1.1360.1 Available under license:

You may use the Forge project under the terms of either the BSD License or the GNU General Public License (GPL) Version 2.

The BSD License is recommended for most projects. It is simple and easy to understand and it places almost no restrictions on what you can do with the Forge project.

If the GPL suits your project better you are also free to use Forge under that license.

You don't have to do anything special to choose one license or the other and you don't have to notify anyone which license you are using. You are free to use this project in commercial projects as long as the copyright header is left intact.

If you are a commercial entity and use this set of libraries in your commercial software then reasonable payment to Digital Bazaar, if you can afford it, is not required but is expected and would be appreciated. If this library saves you time, then it's saving you money. The cost of developing the Forge software was on the order of several hundred hours and tens of thousands of dollars. We are attempting to strike a balance between helping
the development community while not being taken advantage of by lucrative commercial entities for our efforts.

New BSD License (3-clause)
Copyright (c) 2010, Digital Bazaar, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Digital Bazaar, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL DIGITAL BAZAAR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by
When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.
If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.
NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.1361 latest-version 5.1.0

1.1361.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1362 isstream 0.1.2 0.1.2

1.1362.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Rod Vagg

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1363 react-base16-styling 0.6.0 0.6.0

1.1363.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Alexander Kuznetsov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1364 batik-squiggle-ext 1.13

1.1364.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright
owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   You must give any other recipients of the Work or Derivative Works a copy of this License; and
   You must cause any modified files to carry prominent notices stating that You changed the files; and
   You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
   If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for
inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: HOW TO APPLY THE APACHE LICENSE TO YOUR WORK

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1365 hide-powered-by 1.0.0 1.0.0
1.1365.1 Available under license :
The MIT License (MIT)
Copyright (c) 2014, 2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1366 ignore-by-default 1.0.1
1.1366.1 Available under license :
ISC License (ISC)
Copyright (c) 2016, Mark Wubben

Permission to use, copy, modify, and/or distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright notice
and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1367 at-spi2-core 2.18.3-4ubuntu1

1.1367.1 Available under license:

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling.
it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards
changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

      (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above,

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.1368 ncp 1.0.1

1.1368.1 Available under license:
# MIT License

###Copyright (C) 2011 by Charlie McConnell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1369 define-property 1.0.0

1.1369.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015, 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1370 buffer-from 1.1.1 1.1.1

1.1370.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1371 lodash.get 4.4.2 4.4.2
### 1.1371.1 Available under license:

Copyright jQuery Foundation and other contributors [https://jquery.org/](https://jquery.org/)


This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at [https://github.com/lodash/lodash](https://github.com/lodash/lodash)

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: [http://creativecommons.org/publicdomain/zero/1.0/](http://creativecommons.org/publicdomain/zero/1.0/)

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
1.1372 loader-utils 1.2.3 1.2.3

1.1372.1 Available under license:

Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1373 jetty 9.3.27.v20190418

1.1373.1 Available under license:

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular
Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this
Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.1374 eventemitter2 0.4.14
1.1374.1 Available under license:

No license file was found, but licenses were detected in source scan.

(The MIT License)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal to use, copy, modify, merge, publish, distribute, sublicense, and/or sell to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1127088006_1611766545.73/0/eventemitter2-eventemitter2-0-4-14-0-g250dd1f-targ-gz/EventEmitter2-EventEmitter2-250dd1f/README.md

No license file was found, but licenses were detected in source scan.

/*! 
* EventEmitter2 
* https://github.com/hij1nx/EventEmitter2 
* 
* Copyright (c) 2013 hij1nx 
* Licensed under the MIT license. 
*/

Found in path(s):
* /opt/cola/permits/1127088006_1611766545.73/0/eventemitter2-eventemitter2-0-4-14-0-g250dd1f-targ-gz/EventEmitter2-EventEmitter2-250dd1f/lib/eventemitter2.js

No license file was found, but licenses were detected in source scan.

// Permission is hereby granted, free of charge, to any person obtaining a
// copy of this software and associated documentation files (the
// "Software"), to deal in the Software without restriction, including
// without limitation the rights to use, copy, modify, merge, publish,
// distribute, sublicense, and/or sell copies of the Software, and to permit
// The above copyright notice and this permission notice shall be included
// in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1127088006_1611766545.73/0/eventemitter2-eventemitter2-0-4-14-0-g250dd1f-tar-
1.1375 opentracing-util 0.31.0

1.1376 ansistyles 0.1.3 0.1.3

1.1376.1 Available under license:

Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1377 libxdmcp 1.1.2-1.1

1.1377.1 Available under license:

Copyright 1989, 1998 The Open Group

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

Author: Keith Packard, MIT X Consortium

1.1378 foreground-child 2.0.0

1.1378.1 Available under license :

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1379 parquet-common 1.10.0

1.1379.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the

* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/bytes/SingleBufferInputStream.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/ShouldNeverHappenException.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/SemanticVersion.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/Strings.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/Files.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/IOExceptionUtils.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/bytes/ConcatenatingByteArrayCollector.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/util/DynConstructors.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/glob/WildcardPath.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/glob/GlobNode.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/VersionParser.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/bytes/ByteBufferAllocator.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/glob/GlobParser.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/glob/GlobExpander.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/bytes/ByteBufferInputStream.java

No license file was found, but licenses were detected in source scan.
OPEN SOURCE USED IN DNAC PLATFORM ANTELOPE 4919

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
*/

/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/bytes/HeapByteBufferAllocator.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/Preconditions.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/ByteArrayOutputStream.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/Ints.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/BytesInput.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/hadoop/metadata/ColumnPath.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/hadoop/metadata/Canonicalizer.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/hadoop/BytesInput.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/hadoop/ByteArrayOutputStream.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/hadoop/BytesUtils.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/hadoop/light/InputStream.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/hadoop/light/DataOutputStream.java
/opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1.jar/org/apache/parquet/hadoop/light/Closeables.java

No license file was found, but licenses were detected in source scan.

/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * <p/>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p/>
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/OutputStreamCloseException.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/Exceptions.java

No license file was found, but licenses were detected in source scan.

/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * <p/>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p/>
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-
  jar/org/apache/parquet/Exceptions.java
jar/org/apache/parquet/io/PositionOutputStream.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-jar/org/apache/parquet/hadoop/metadata/CompressionCodecName.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-jar/org/apache/parquet/io/DelegatingSeekableInputStream.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-jar/org/apache/parquet/io/DelegatingPositionOutputStream.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-jar/org/apache/parquet/io/InputFile.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-jar/org/apache/parquet/io/OutputFile.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-jar/org/apache/parquet/io/SeekableInputStream.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-jar/org/apache/parquet/hadoop/codec/CompressionCodecNotSupportedException.java
* /opt/cola/permits/1136206342_1613708930.97/0/parquet-common-1-10-0-sources-1-jar/org/apache/parquet/compression/CompressionCodecFactory.java

1.1380 socket.io 1.7.4
1.1380.1 Available under license:
(The MIT License)

Copyright (c) 2014-2016 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Open Source Used In DNAC Platform Antelope 4921
1.1381 serve-favicon 2.5.0
1.1381.1 Available under license:
(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1382 unorm 1.6.0
1.1382.1 Available under license:
The software dual licensed under the MIT and GPL licenses. MIT license:

Copyright (c) 2008-2013 Matsuza <matsuza@gmail.com>, Bjarke Walling <bwp@bwp.dk>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GPL notice (please read the [full GPL license] online):

Copyright (C) 2008-2013 Matsuza <matsuza@gmail.com>, Bjarke Walling <bwp@bwp.dk>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

[full GPL license]: http://www.gnu.org/licenses/gpl-2.0-standalone.html

1.1383 prepend-http 1.0.4 1.0.4

1.1383.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.1384 readdirp 3.3.0
1.1384.1 Available under license:
MIT License

Copyright (c) 2012-2019 Thorsten Lorenz, Paul Miller (https://paulmillr.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1385 javax-annotation-api 1.2
1.1385.1 Available under license:
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You
represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.
4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant,) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH
PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.
The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying
it may be distributed under the terms of this General Public License. The "Program", below, refers to any such
program or work, and a "work based on the Program" means either the Program or any derivative work under
copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications
and/or translated into another language. (Hereinafter, translation is included without limitation in the term
"modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its
scope. The act of running the Program is not restricted, and the output from the Program is covered only if its
contents constitute a work based on the Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium,
provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and
disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and
give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty
protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the
Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you
also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of
      any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the
      Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this
      License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started
      running for such interactive use in the most ordinary way, to print or display an announcement including an
      appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and
      that users may redistribute the program under these conditions, and telling the user how to view a copy of this
      License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your
      work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from
the Program, and can be reasonably considered independent and separate works in themselves, then this License,
and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute
the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on
the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and
every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather,
the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on
the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as
to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.
You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.1386 ms 0.7.2

1.1386.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Zeit, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1387 objenesis 2.6

1.1387.1 Available under license:

// ------------------------------------------------------------------
// NOTICE file corresponding to the section 4d of The Apache License,
// Version 2.0, in this case for Objenesis
// ------------------------------------------------------------------

Objenesis
Copyright 2006-2017 Joe Walnes, Henri Tremblay, Leonardo Mesquita

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1388 xmldom 0.2.1

1.1388.1 Available under license:
You can choose any one of those:

The MIT License (MIT):

link:http://opensource.org/licenses/MIT

LGPL:
http://www.gnu.org/licenses/lgpl.html
1.1389 drill-jdbc-storage 1.19.0
1.1389.1 Available under license:

Drill : Contrib : Storage : JDBC
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1390 forever-agent 0.6.1 0.6.1

1.1390.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1391 zip 1.2.0
1.1391.1 Available under license:

Copyright 2009, 2010 Kristopher Michael Kowal. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1392 parquet-column 1.12.0
1.1392.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

Open Source Used In DNAC Platform Antelope 4950
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This product depends on fastutil and includes it in this binary artifact.
Fastutil provides type-specific collection implementations.

Copyright: 2002-2014 Sebastiano Vigna
Home page: http://fasutil.di.unimi.it/
License: http://www.apache.org/licenses/LICENSE-2.0.html

1.1393 jackson-datatype-jdk8 2.10.5

1.1393.1 Available under license :

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1595301005506
Build-Jdk: 1.8.0_162
Built-By: tatu
Bundle-Description: Add-on module for Jackson (http://jackson.codehaus.
.org) to support JDK 8 data types.
Bundle-DocURL: https://github.com/FasterXML/jackson-modules-java8/jack
son-datatype-jdk8
Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.txt
Bundle-ManifestVersion: 2
Bundle-Name: Jackson datatype: jdk8
Bundle-SymbolicName: com.fasterxml.jackson.datatype.jackson-datatype-j
dk8
Bundle-Vendor: FasterXML
1.1394 har-schema 2.0.0 2.0.0

1.1394.1 Available under license:

Copyright (c) 2015, Ahmad Nassri <ahmad@ahmadnassri.com>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES.
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1395 drange 1.1.1 1.1.1
1.1395.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 David Tudury

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1396 iconv-lite 0.4.24 0.4.24
1.1396.1 Available under license :
Copyright (c) 2011 Alexander Shtuchkin

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1397 rxjs 6.6.3

1.1397.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and eachContributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2015-2018 Google, Inc., Netflix, Inc., Microsoft Corp. and contributors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

@title
@description
The MIT License

Copyright (c) 2014-2018 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1398 jackson-datatype-joda 2.12.1
1.1398.1 Available under license:

This copy of Jackson JSON processor streaming parser/generator is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.1399 asm-tree 7.3.1
1.1399.1 Available under license:

No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
//    this software without specific prior written permission.
/**
 * Constructs a new {@link RecordComponentNode}. <i>Subclasses must not use this constructor</i>. Instead, they must use the `{@link #RecordComponentNode(int, int, String, String, String)}`
 * version.
 *
 * @param access the record component access flags (see `{@link org.objectweb.asm.Opcodes}`). The only valid value is `{@link Opcodes#ACC_DEPRECATED}`.
 * @param name the record component name.
 * @param descriptor the record component descriptor (see `{@link org.objectweb.asm.Type}`).
 * @param signature the record component signature.
 * @throws IllegalStateException If a subclass calls this constructor.
 * @deprecated this API is experimental.
 */

Found in path(s):
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/RecordComponentNode.java
No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/FieldInsnNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/LineNumberNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/ParameterNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/LocalVariableAnnotationNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/AnnotationNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/InvokeDynamicInsnNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/LookupSwitchInsnNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/InsnNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/TypeInsnNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/InsnList.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/IincInsnNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/ModuleOpenNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/TypeInsnNode.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/TypeList.java
* /opt/cola/permits/1135907466_1613628634.84/0/asm-tree-7-3-1-sources-1-jar/org/objectweb/asm/tree/InnerClassNode.java
No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom
* All rights reserved.

* Redistribution and use in source and binary forms, with or without
  modification, are permitted provided that the following conditions
  are met:
  1. Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.
  2. Redistributions in binary form must reproduce the above copyright
     notice, this list of conditions and the following disclaimer in the
     documentation and/or other materials provided with the distribution.
  3. Neither the name of the copyright holders nor the names of its
Open Source Used In DNAC Platform Antelope 4963

1.1400 joda-time 2.10.5
1.1400.1 Available under license :

=================================================================
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=================================================================

This product includes software developed by
Joda.org (https://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1401 babel-messages 6.23.0

1.1401.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1402 converter-jackson 2.8.1

1.1402.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446210_1631568067.11/0/converter-jackson-2-8-1-sources.jar/retrofit2/converter/jackson/JacksonRequestBodyConverter.java
* /opt/cola/permits/1202446210_1631568067.11/0/converter-jackson-2-8-1-sources.jar/retrofit2/converter/jackson/JacksonConverterFactory.java
* /opt/cola/permits/1202446210_1631568067.11/0/converter-jackson-2-8-1-sources.jar/retrofit2/converter/jackson/JacksonResponseBodyConverter.java

1.1403 xmlhttprequest-ssl 1.5.3

1.1403.1 Available under license:
Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1404 is-resolvable 1.1.0

1.1404.1 Available under license :

ISC License (ISC)
Copyright 2018 Shinnosuke Watanabe

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1405 object.omit 2.0.1

1.1405.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1406 type-of-is 3.5.1 3.5.1

1.1406.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1407 balanced-match 1.0.0

1.1407.1 Available under license:

(MIT)

Copyright (c) 2013 Julian Gruber &lt;julian@juliangruber.com&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1408 wide-align 1.1.3

1.1408.1 Available under license:
Copyright (c) 2015, Rebecca Turner <me@rebecca.org>

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1409 commons-io 1.3.2

1.1409.1 Available under license:
Apache Jakarta Commons IO
Copyright 2001-2007 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1410 asm 5.2
1.1411 hive-llap-client 3.1.2

1.1411.1 Available under license:

Hive Llap Client
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1412 commons-collections 3.2.2

1.1412.1 Available under license :
Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1413 commander 2.20.3
1.1413.1 Available under license:
(The MIT License)

Copyright (c) 2011 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1414 crc32-stream 2.0.0

1.1414.1 Available under license:
Copyright (c) 2014 Chris Talkington, contributors.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
1.1415 javax-inject 2.2.0-b21

1.1415.1 Available under license:

No license file was found, but licenses were detected in source scan.

2011 Oracle and/or its affiliates. All rights reserved.

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can obtain a copy of the License at https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html or packager/legal/LICENSE.txt. See the License for the specific language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each file and include the License file at packager/legal/LICENSE.txt.

GPL Classpath Exception:
Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or only the GPL Version 2, indicate your decision by adding "[Contributor] elects to include this software in this distribution under the [CDDL or GPL Version 2] license." If you don't indicate a single choice of license, a recipient has the option to distribute your version of this file under either the CDDL, the GPL Version 2 or to extend the choice of license to its licensees as provided above. However, if you add GPL Version 2 code and therefore, elected the GPL Version 2 license, then the option applies only if the new code is made subject to such option by the copyright holder.

Found in path(s):
* /opt/cola/permits/1136447767_1613767069.04/0/javax-inject-2-2-0-b21-sources-2-jar/pom.xml
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The JSR-330 Expert Group
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136447767_1613767069.04/0/javax-inject-2-2-0-b21-sources-2-jar/javax/inject/package-info.java
* /opt/cola/permits/1136447767_1613767069.04/0/javax-inject-2-2-0-b21-sources-2-jar/javax/inject/Qualifier.java
* /opt/cola/permits/1136447767_1613767069.04/0/javax-inject-2-2-0-b21-sources-2-jar/javax/inject/Singleton.java
* /opt/cola/permits/1136447767_1613767069.04/0/javax-inject-2-2-0-b21-sources-2-jar/javax/inject/Provider.java
* /opt/cola/permits/1136447767_1613767069.04/0/javax-inject-2-2-0-b21-sources-2-jar/javax/inject/Named.java
* /opt/cola/permits/1136447767_1613767069.04/0/javax-inject-2-2-0-b21-sources-2-jar/javax/inject/Scope.java
* /opt/cola/permits/1136447767_1613767069.04/0/javax-inject-2-2-0-b21-sources-2-jar/javax/inject/Inject.java

1.1416 batik-gui-util 1.13
1.1416.1 Available under license:

Apache Batik
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the
Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for
Standardization for the definition of character entities used in the software's
documentation.

This product includes images from the Tango Desktop Project
(http://tango.freedesktop.org/).
This product includes images from the Pasodoble Icon Theme

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1417 jsbn 0.1.1 0.1.1
1.1417.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1418 qdox 1.12.1
1.1419 object.values 1.1.1
1.1419.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1420 batik-xml 1.13
1.1420.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Open Source Used In DNAC Platform Antelope 4996
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

No license file was found, but licenses were detected in source scan.

*/
1.1421 regenerate 1.4.1
1.1421.1 Available under license :
MIT

1.1422 jython 2.7.1
1.1422.1 Available under license :

```
1)+(+V<cinit><cinit>[Copyright (c) 2000-2016 The Legion of the Bouncy Castle Inc.
(http://www.bouncycastle.org) DEALINGS IN THE SOFTWARE.XINCCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR|LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT ORVOTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHERPURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE^Permission is hereby granted, free of charge, to any person obtaining a copy of this software STHE
SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,dThe
above copyright notice and this permission notice shall be included in all copies or substantialband associated
documentation files (the "Software"), to deal in the Software without restriction, eand/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so.appendfincluding without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense,
java/io/PrintStreamjava/lang/Objectjava/lang/StringBuilderjava/lang/SystemlicenseText
lineSeparatormainorg/bouncycastle/LICENSEorg/bouncycastle/util/Stringsoutportions of the
Software.println$subject to the following conditions:toString
"&'Ljava/io/PrintStream;Ljava/lang/String;()Ljava/lang/String;(Ljava/lang/String;)V(Ijava/lang/String;)V-
(Ljava/lang/String:)Ljava/lang/StringBuilder;3#4$5,5*68091:
->
;;
/;
/=?
2<Code!1.#4H*C%?H
@ABH/YDFGF'GGFG
FGF''FFGF'GGFG
FGF''FFGF'GGFGF'GFFEA
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
```
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JUnit License

Common Public License Version 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide,
royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient’s responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows
4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL
If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2017 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in http://www.unicode.org/copyright.html

Permission is hereby granted, free of charge, to any person obtaining
a copy of the Unicode data files and any associated documentation
(the "Data Files") or Unicode software and any associated documentation
(the "Software") to deal in the Data Files or Software
without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, and/or sell copies of
the Data Files or Software, and to permit persons to whom the Data Files
or Software are furnished to do so, provided that either
(a) this copyright and permission notice appear with all copies
of the Data Files or Software, or
(b) this copyright and permission notice appear in associated
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale,
use or other dealings in these Data Files or Software without prior
written authorization of the copyright holder.

---------------------
Third-Party Software Licenses

This section contains third-party software notices and/or additional
terms for licensed third-party software components included within ICU
libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies of
the Software and that both the above copyright notice(s) and this
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale, use
or other dealings in this Software without prior written authorization
of the copyright holder.

All trademarks and registered trademarks mentioned herein are the
property of their respective owners.


# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
# # Redistributions of source code must retain the above copyright notice,
# # this list of conditions and the following disclaimer.
# # Redistributions in binary form must reproduce the above
# # copyright notice, this list of conditions and the following
# # disclaimer in the documentation and/or other materials provided with
# # the distribution.
# # Neither the name of Google Inc. nor the names of its
# # contributors may be used to endorse or promote products derived from
# # this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
#  - https://sourceforge.net/project/?group_id=1519
#  - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
#  - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#  - Its license terms and conditions are shown below.
#
# ---------COPYING.libtabe ---- BEGIN-------------------
#
# /*
# Copyright (c) 1999 TaBE Project.
# Copyright (c) 1999 Pai-Hsiang Hsiao.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
# *
# * Redistributions of source code must retain the above copyright
# notice, this list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions and the following disclaimer in
# the documentation and/or other materials provided with the
# distribution.
# * Neither the name of the TaBE Project nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-tsai4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsai4
#
# ---------------COPYING.libtabe-----END-----------------------------
#
#
# ---------------COPYING.ipadic-----BEGIN-------------------------------
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise. The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
# project or thereafter. Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# ----------------COPYING.ipadic-----END-------------------------------

3. Lao Word Break Dictionary Data (laodict.txt)

# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
# (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# ---------------------------------------------------------------
# Copyright (c) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.

# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.

# 4. Burmese Word Break Dictionary Data (burmesedict.txt)

# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
# github.com/kanyawtech/myanmar-karen-word-lists
#
# -----------------------------------------------

# Copyright (c) 2013, LeRoy Benjamin Sharon
# All rights reserved.

# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
# disclaimer. Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided
 Neither the name Myanmar Karen Word Lists, nor the names of its
 contributors may be used to endorse or promote products derived
 from this software without specific prior written permission.

 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
 CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
 INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
 BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
 TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
 TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
 THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 SUCH DAMAGE.

 5. Time Zone Database

 ICU uses the public domain data and code derived from Time Zone
 Database for its time zone support. The ownership of the TZ database
 is explained in BCP 175: Procedure for Maintaining the Time Zone
 Database section 7.

 7. Database Ownership

 The TZ database itself is not an IETF Contribution or an IETF
 document. Rather it is a pre-existing and regularly updated work
 that is in the public domain, and is intended to remain in the
 public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do
 not apply to the TZ Database or contributions that individuals make
 to it. Should any claims be made and substantiated against the TZ
 Database, the organization that is providing the IANA
 Considerations defined in this RFC, under the memorandum of
 understanding with the IETF, currently ICANN, may act in accordance
 with all competent court orders. No ownership claims will be made
 by ICANN or the IETF Trust on the database or the code. Any person
 making a contribution to the database or code waives all rights to
 future claims in that contribution or in the TZ Database.

 Apache License
 Version 2.0, January 2004
 http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Apache Commons Compress
Copyright 2002-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

The files in the package org.apache.commons.compress.archivers.sevenz
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)

JBoss (http://www.jboss.org) is distributed under the LGPL. Mockrunner
includes the jboss-j2ee.jar and jboss-jaas.jar.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.
6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.
For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.
10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries
If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
This product includes software developed by:
 o The Apache Software Foundation (http://www.apache.org/)

A. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING JYTHON

========================================================================

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2
1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Jython") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Jython alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2007 Python Software Foundation; All Rights Reserved" are retained in Jython alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Jython or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Jython.

4. PSF is making Jython available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF JYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF JYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING JYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Jython, Licensee agrees to be bound by the terms and conditions of this License Agreement.

Jython 2.0, 2.1 License
Open Source Used In DNAC Platform  Antelope  5035

================================
Copyright (c) 2000-2009 Jython Developers.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in
  the documentation and/or other materials provided with the distribution.

- Neither the name of the Jython Developers nor the names of
  its contributors may be used to endorse or promote products
  derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JPython 1.1.x Software License.
================================

IMPORTANT: PLEASE READ THE FOLLOWING AGREEMENT CAREFULLY.

BY CLICKING ON THE "ACCEPT" BUTTON WHERE INDICATED, OR BY INSTALLING,
COPYING OR OTHERWISE USING THE SOFTWARE, YOU ARE DEEMED TO HAVE AGREED TO
THE TERMS AND CONDITIONS OF THIS AGREEMENT.
JPython version 1.1.x

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and using JPython version 1.1.x in source or binary form and its associated documentation as provided herein ("Software").

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a non-exclusive, non-transferable, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright 1996-1999 Corporation for National Research Initiatives; All Rights Reserved" are both retained in the Software, alone or in any derivative version prepared by Licensee.

Alternatively, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes), provided, however, that such text is displayed prominently in the Software alone or in any derivative version prepared by Licensee: "JPython (Version 1.1.x) is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1006. The License may also be obtained from a proxy server on the Web using the following URL: http://hdl.handle.net/1895.22/1006."

3. In the event Licensee prepares a derivative work that is based on or incorporates the Software or any part thereof, and wants to make the derivative work available to the public as provided herein, then Licensee hereby agrees to indicate in any such work, in a prominently visible way, the nature of the modifications made to CNRI's Software.

4. Licensee may not use CNRI trademarks or trade name, including JPython or CNRI, in a trademark sense to endorse or promote products or services of Licensee, or any third party. Licensee may use the mark JPython in connection with Licensee's derivative versions that are based on or incorporate the Software, but only in the form "JPython-based ________________," or equivalent.

5. CNRI is making the Software available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
6. CNRI SHALL NOT BE LIABLE TO LICENSEE OR OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF; EVEN IF ADVISED OF THE POSSIBILITY THEREOF. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY SO THE ABOVE DISCLAIMER MAY NOT APPLY TO LICENSEE.

7. This License Agreement may be terminated by CNRI (i) immediately upon written notice from CNRI of any material breach by the Licensee, if the nature of the breach is such that it cannot be promptly remedied; or (ii) sixty (60) days following notice from CNRI to Licensee of a material remediable breach, if Licensee has not remedied such breach within that sixty-day period.

8. This License Agreement shall be governed by and interpreted in all respects by the law of the State of Virginia, excluding conflict of law provisions. Nothing in this Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee.

9. By clicking on the "ACCEPT" button where indicated, or by installing, copying or otherwise using the Software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

[ACCEPT BUTTON]

B. HISTORY OF THE SOFTWARE

=======================================================

JPython was created in late 1997 by Jim Hugunin. Jim was also the primary developer while he was at CNRI. In February 1999 Barry Warsaw took over as primary developer and released JPython version 1.1.

In October 2000 Barry helped move the software to SourceForge where it was renamed to Jython. Jython 2.0 and 2.1 were developed under the Jython specific license below.

From the 2.2 release on, Jython contributors have signed Python Software Foundation contributor agreements and releases are covered under the Python Software Foundation license version 2.

The standard library is covered by the Python Software Foundation license as well. See the Lib/LICENSE file for details.

The zxJDBC package was written by Brian Zimmer and originally licensed under the GNU Public License. The package is now covered by the Jython Software License.
The command line interpreter is covered by the Apache Software License. See the org/apache/LICENSE file for details.
/*--

Copyright (C) 2000-2003 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions, and the disclaimer that follows
these conditions in the documentation and/or other materials
provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <license AT jdom DOT org>.

4. Products derived from this software may not be called "JDOM", nor
may "JDOM" appear in their name, without prior written permission
from the JDOM Project Management <pm AT jdom DOT org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter AT jdom DOT org> and Brett McLaughlin <brett AT jdom DOT org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

===========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xerces Java distribution. ==
===========================================================================

Apache Xerces Java
Copyright 1999-2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

```java
3Y
Z[:]
(^
(\_a
b
cdef
gh
ij)
kl-m.n/o/p1q2r3s4t5u6v7wxSETUID
PERMISSIONInnerClassesHLorg/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;
SETGUISTICKYUSER_READ
USER_WRITEUSER_EXEC
GROUP_READGROUP_USER_WRITE
GROUP_EXEC
WORLD_READWORLD_WRITE
WORLD_EXECCodeISVALUESI[Lorg/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;
ION;valuesK()Lorg/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;CodeLineNumberTablevalueOf(Ljava/lang/String;);Lorg/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;LocalVariableTableNameLjava/lang/String;<init>(Ljava/lang/String;II)Vthis
Signature(I)Vfind(I)Ljava/util/Set;psetLjava/util/Set;LocalVariableTypeTableYLjava/util/Set<Lorg/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;>;class(I)ZLjava/lang/Enum<Lorg/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;>;StackMapTabley(I)Ljava/util/Set<Lorg/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;>;sclinit>()VZLjava/lang/Enum<Lorg/apache/commons/compress/archivers/dump/DumpArchiveEntry$PERMISSION;>;
```
1.1423 snappy 1.0.5

1.1423.1 Available under license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Some of the benchmark data in testdata/ is licensed differently:

- fireworks.jpeg is Copyright 2013 Steinar H. Gunderson, and is licensed under the Creative Commons Attribution 3.0 license (CC-BY-3.0). See https://creativecommons.org/licenses/by/3.0/ for more information.

- kppkn.gtb is taken from the Gaviota chess tablebase set, and is licensed under the MIT License. See https://sites.google.com/site/gaviotachessengine/Home/endgame-tablebases-1 for more information.

- paper-100k.pdf is an excerpt (bytes 92160 to 194560) from the paper Combinatorial Modeling of Chromatin Features Quantitatively Predicts DNA Replication Timing in _Drosophila_ by Federico Comoglio and Renato Paro, which is licensed under the CC-BY license. See http://www.ploscompbiol.org/static/license for more information.

- alice29.txt, asyoulik.txt, plraban12.txt and lcet10.txt are from Project Gutenberg. The first three have expired copyrights and are in the public domain; the latter does not have expired copyright, but is still in the public domain according to the license information (http://www.gutenberg.org/ebooks/53).

1.1424 jsch 0.1.55

1.1425 babel-plugin-transform-async-generator-functions 6.24.1

1.1425.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1426 camelcase 5.3.1 5.3.1

1.1426.1 Available under license:
MIT License

Copyright (c) Sindresorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1427 resourcelocator 1.0.1

1.1427.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under
common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You
represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.
COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. § 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This

Open Source Used In DNAC Platform Antelope 5051
alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.
To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent
modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.1428 metrics-health-checks 4.0.2

1.1429 fs-extra 1.0.0

1.1429.1 Available under license:
(The MIT License)

Copyright (c) 2011-2016 JP Richardson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1430 jquery 3.5.1 3.5.1
1.1430.1 Available under license:

Copyright JS Foundation and other contributors, https://js.foundation/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright JS Foundation and other contributors, https://js.foundation/

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/sizzle

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

All files located in the node_modules and external directories are
externally maintained libraries used by this software which have their
own licenses; we recommend you read them, as their terms may differ from
the terms above.

1.1431 jansi-osx 1.8
1.1431.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1432 html-escaper 2.0.2

1.1432.1 Available under license:

Copyright (C) 2017-present by Andrea Giammarchi - @WebReflection

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1433 parse-color 1.0.0

1.1433.1 Available under license:

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1434 es5-ext 0.10.53

1.1434.1 Available under license:

ISC License

Copyright (c) 2011-2019, Mariusz Nowak, @medikoo, medikoo.com

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1435 bcrypt-pbkdf 1.0.2 1.0.2
1.1435.1 Available under license:

The Blowfish portions are under the following license:

Blowfish block cipher for OpenBSD
Copyright 1997 Niels Provos <provos@physnet.uni-hamburg.de>
All rights reserved.

Implementation advice by David Mazieres <dm@lcs.mit.edu>.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The bcrypt_pbkdf portions are under the following license:

Copyright (c) 2013 Ted Unangst <tedu@openbsd.org>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Performance improvements (Javascript-specific):

Copyright 2016, Joyent Inc
Author: Alex Wilson <alex.wilson@joyent.com>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1436 ratelimitj-core 0.4.0

1.1436.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1437 json-buffer 3.0.0

1.1437.1 Available under license:

Copyright (c) 2013 Dominic Tarr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1438 jackson-jaxrs-xml-provider 2.10.4

1.1438.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.salaranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.
1.1439 component-bind 1.0.0 1.0.0
1.1439.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1440 helmet 2.3.0 2.3.0
1.1440.1 Available under license:

(The MIT License)

Copyright (c) 2012-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the 'Software'),
to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1441 babel-plugin-transform-es2015-destructuring 6.23.0 6.23.0

1.1441.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1442 color-convert 2.0.1 2.0.1

1.1442.1 Available under license:

Copyright (c) 2011-2016 Heather Arthur <fayearthur@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

1.1443 jersey-servlet 1.19

1.1444 nanoid 1.0.2
1.1444.1 Available under license:
The MIT License (MIT)

Copyright 2017 Andrey Sitnik <andrey@sitnik.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

1.1445 invariant 2.2.4 2.2.4
1.1445.1 Available under license:
The MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1446 release-zalgo 1.0.0

1.1446.1 Available under license:
ISC License (ISC)
Copyright (c) 2017, Mark Wubben <mark@novemberborn.net> (novemberborn.net)

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1447 http2-client 1.3.3 1.3.3

1.1447.1 Available under license:
MIT License

Copyright (c) 2019 Eyal.D <hisco@googlegroups.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1448 babylon 6.18.0
1.1448.1 Available under license :
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1449 object-component 0.0.3 0.0.3
1.1449.1 Available under license :
The MIT License (MIT)
Copyright (c) 2012 Mike Ihbe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1450 stax-ex 1.8

1.1451 lucene-sandbox 7.4.0

1.1451.1 Available under license:

    Apache License
    Version 2.0, January 2004
    http://www.apache.org/licenses/

    TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

/*
 * Copyright 2001-2004 Unicode, Inc.
 * *
 * Disclaimer
 * *
 * This source code is provided as is by Unicode, Inc. No claims are
 * made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at www.brics.dk/automaton/. Here is the copyright from those sources:

/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
* 
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
* 
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the manan/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
* *
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfoligik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---

The dictionary comes from Morfoligik project. Morfoligik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfoligik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

---

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczynski, Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0. See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0 See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin g Package (jaspell): http://jaspell.sourceforge.net/ License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
   analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
   analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
   http://snowball.tartarus.org/

The KStem stemmer in
   analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst) under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
   analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
   analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers (common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:
   analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
   analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
The Stempel analyzer (stempel) includes BSD-licensed software developed by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil, and Edmond Nolan.

The Polish analyzer (stempel) comes with a default stopword list that is BSD-licensed created by the Carrot2 project. The file resides in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt. See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/) is derived from Unicode data such as the Unicode Character Database. See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary (http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP) (http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===========================================================================
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from mecab-ipadic-2.7.0-20070801 which can be obtained from http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz or http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz
Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortious action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
IMPLIED, STATUTORY or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification, copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

===========================================================================
Nori Korean Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from

mecab-ko-dic-2.0.3-20170922

which can be obtained from

https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.0.3-20170922.tar.gz

1.1452 batik-gvt 1.13

1.1452.1 Available under license :

Apache Batik
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the
Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for
Standardization for the definition of character entities used in the software's
documentation.
This product includes images from the Tango Desktop Project (http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme (http://www.jesusda.com/projects/pasodoble).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1453 drill-format-hdf5 1.19.0

1.1453.1 Available under license:

Drill : Contrib : Format : HDF5
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1454 jsonpath-plus 3.0.0 3.0.0

1.1454.1 Available under license:
The MIT License (MIT)

Copyright (c) 2011-2019 Stefan Goessner, Subbu Allamaraju, Mike Brevoort, Robert Krahn, Brett Zamir, Richard Schneider

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1455 babel-plugin-transform-es2015-block-scoped-functions 6.22.0

1.1455.1 Available under license:
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is
being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International.  SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
   following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
   from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1456 getpass 0.1.7 0.1.7

1.1456.1 Available under license:

Copyright Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1457 websocket-server 9.3.27.v20190418

1.1458 is-data-descriptor 0.1.4

1.1458.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1459 which-module 2.0.0

1.1459.1 Available under license:

Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.1460 jackson 1.9.13
1.1460.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic
mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of
any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1461 options 0.0.6
1.1461.1 Available under license :
No license file was found, but licenses were detected in source scan.

/!
* Copyright(c) 2011 Einar Otto Stangvik <einaros@gmail.com>
* MIT Licensed
*/

Found in path(s):
* /opt/cola/permits/1110632234_1607050508.79/0/options-0-0-6-2-tgz/package/lib/options.js
No license file was found, but licenses were detected in source scan.

# options.js #

A very light-weight in-code option parsers for node.js.
## Usage ##

```js
var Options = require("options");

// Create an Options object
function foo(options) {
    var default_options = {
        foo : "bar"
    };

    // Create an option object with default value
    var opts = new Options(default_options);

    // Merge options
    opts = opts.merge(options);

    // Reset to default value
    opts.reset();

    // Copy selected attributes out
    var seled_att = opts.copy("foo");

    // Read json options from a file.
    opts.read("options.file"); // Sync
    opts.read("options.file", function(err){ // Async
        if(err) { // If error occurs
            console.log("File error.");
        } else {
            // No error
        }
    });

    // Attributes defined or not
    opts.isDefinedAndNonNull("foobar");
    opts.isDefined("foobar");
}
```

## License ##

(The MIT License)

Copyright (c) 2012 Einar Otto Stangvik &lt;einaros@gmail.com&gt;
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**1.1462 async-each 1.0.3**

**1.1462.1 Available under license :**

No license file was found, but licenses were detected in source scan.

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal to use, copy, modify, merge, publish, distribute, sublicense, and/or sell furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1110633431_1613673624.58/0/paulmillr-async-each-1-0-3-0-g9c0980f-1-tar-gz/paulmillr-async-each-9c0980f/README.md

No license file was found, but licenses were detected in source scan.

// async-each MIT license (by Paul Miller from https://paulmillr.com).

Found in path(s):
* /opt/cola/permits/1110633431_1613673624.58/0/paulmillr-async-each-1-0-3-0-g9c0980f-1-tar-gz/paulmillr-async-each-9c0980f/index.js
1.1463 okhttp 3.9.1

1.1463.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and where such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.1464 run-async 0.1.0

1.1464.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Simon Boudrias

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1465 has 1.0.3 1.0.3

1.1465.1 Available under license:

Copyright (c) 2013 Thiago de Arruda

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1466 hive 3.1.2

1.1466.1 Available under license:

Javolution - Java(tm) Solution for Real-Time and Embedded Systems
Copyright (c) 2012, Javolution (http://javolution.org/)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGligence OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2000-2005 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2002, Steve Souza (admin@jamonapi.com)
All rights reserved.
Modifications: No

Redistribution in binary form, with or without modifications, are permitted provided that the following conditions
are met:

* Redistributions in binary form must reproduce the above copyright notice, this
  list of conditions and the following disclaimer in the documentation and/or
  other materials provided with the distribution.
* If modifications are made to source code then this license should indicate 
that fact in the "Modifications" section above.

* Neither the author, nor the contributors may be used to endorse or promote 
products derived from this software without specific prior written 
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR 
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF 
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT 
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, 
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT 
LIMITED 
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; 
OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER 
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING 
IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF 
SUCH DAMAGE.

Mozilla Public License Version 2.0
==================================

1. Definitions
-------------

1.1. "Contributor"
means each individual or legal entity that creates, contributes to 
the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used 
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached 
the notice in Exhibit A, the Executable Form of such Source Code 
Form, and Modifications of such Source Code Form, in each case 
including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described 
in Exhibit B to the Covered Software; or
(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this
License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:
(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

-----------------------------------------------------------------------------------------------------------------------------------
If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

************************************************************************
**                                                                      **
**  6. Disclaimer of Warranty                                           **
**  -------------------------                                           **
**                                                                      **
**  Covered Software is provided under this License on an "as is"       **
**  basis, without warranty of any kind, either expressed, implied, or  **
**  statutory, including, without limitation, warranties that the       **
**  Covered Software is free of defects, merchantable, fit for a        **
************************************************************************
particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability
Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation
Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous
This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

---------------------------

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

-------------------------------------------

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.
Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.
Binary distributions of this software contain jars that are not licensed under the Apache License 2.0. Additional licenses attached to these jars are contained in the same directory as this NOTICE file.

Note also that some of the included jars impose additional restrictions on you if you modify the source code of those jars. In particular see:

* https://glassfish.dev.java.net/public/CDDLv1.0.html for information on the CDDL license that glassfish is distributed under and

* http://www.mozilla.org/MPL/MPL-1.1.txt for information on the Mozilla Public License which jamon-runtime and rhino are distributed under.

* https://opensource.org/licenses/cddl1.php for information on CDDL 1.0 which transaction-api is distributed under.

Copyright (c) 2001-2016, Arno Unkrig
Copyright (c) 2015-2016  TIBCO Software Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of JANINO nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2006 Paul Hammant & ThoughtWorks Inc
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2012 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following: A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications; B. Any new file that contains any part of the Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version,
or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as
4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY. COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE
INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS. The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

Binary distributions of this software contain jars that are not licensed under the Apache License 2.0. Additional licenses attached to these jars are contained in the same directory as this NOTICE file.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in
The Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2011, abego Software GmbH, Germany (http://www.abego.org)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the abego Software GmbH nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Hive
Copyright 2008-2019 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This project includes software licensed under the JSON license.
Apache License
Version 2.0, January 2017
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability contains
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE HIVE SUBCOMPONENTS:

The Apache Hive project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

For the SQLLine package:

Copyright (c) 2002, 2003, 2004, 2005 Marc Prud'hommeaux

From: http://sqlline.sourceforge.net/#license

"SQLLine is distributed under the BSD License, meaning that you are free to redistribute, modify, or sell the
software with almost no restrictions."

Statement from Marc Prud'hommeaux regarding inconsistent licenses in some SQLLine source files:

> SQLLine was once GPL, but it was changed to BSD a few years back.
> Any references to the GPL are vestigial. Hopefully the license
> declaration at http://sqlline.sourceforge.net/#license is sufficiently
> authoritative in this regard.

For the org.apache.hive.beeline.ClassNameCompleter class:

Copyright (c) 2002-2006, Marc Prud'hommeaux <mwp1@cornell.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the following
conditions are met:

Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For org.apache.hadoop.hive.llap.daemon.impl.PriorityBlockingDeque class:

The BSD 3-Clause License

Copyright (c) 2007, Aviad Ben Dov

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of Infomancers, Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For jquery.sparkline.js:

License: New BSD License (3-clause)

Copyright (c) 2012, Splunk Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Splunk Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For json.human.js/json.human.css:
Copyright (c) 2016 Mariano Guerra

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For argparse.py:

argparse is (c) 2006-2009 Steven J. Bethard <steven.bethard@gmail.com>.

The argparse module was contributed to Python as of Python 2.7 and thus was licensed under the Python license. Same license applies to all files in the argparse package project.

For details about the Python License, please see doc/Python-License.txt.

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc.  All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

Apache Hive Storage API

Copyright 2008-2017 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2016 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in http://www.unicode.org/copyright.html

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.
THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

---------------------

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.


# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
#   - https://sourceforge.net/project/?group_id=1519
#   - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#   - Its license terms and conditions are shown below.
#
# ----------COPYING.libtabe --- BEGIN-----------------
#
# * Copyright (c) 1999 TaBE Project.
# * Copyright (c) 1999 Pai-Hsiang Hsiao.
# * All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * . Neither the name of the TaBE Project nor the names of its
# * contributors may be used to endorse or promote products derived
# * from this software without specific prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.

*/
#
#
* Copyright (c) 1999 Computer Systems and Communication Lab,
* Institute of Information Science, Academia
* Sinica. All rights reserved.
*
* Redistributions and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* . Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* . Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.
* . Neither the name of the Computer Systems and Communication Lab
* nor the names of its contributors may be used to endorse or
* promote products derived from this software without specific
* prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.

*/
#
#
Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
University of Illinois
c-tsa4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsa4
#
# --------------COPYING.libtabe------END-----------------------------
#
#
# --------------COPYING.ipadic------BEGIN-------------------------------
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
Use, reproduction, and distribution of this software is permitted. Any copy of this software, whether in its original form or modified, must include both the above copyright notice and the following paragraphs.

Nara Institute of Science and Technology (NAIST), the copyright holders, disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall NAIST be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortuous action, arising out of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification, copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

--------------COPYING.ipadic-----END----------------------------------

3. Lao Word Break Dictionary Data (laodict.txt)

Copyright (c) 2013 International Business Machines Corporation
and others. All Rights Reserved.

Project: http://code.google.com/p/lao-dictionary/
License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
    (copied below)

This file is derived from the above dictionary, with slight
modifications.

Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification,
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer. Redistributions in
binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
4. Burmese Word Break Dictionary Data (burmesedict.txt)

# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
# github.com/kanyawtech/myanmar-karen-word-lists
#
# Copyright (c) 2013, LeRoy Benjamin Sharon
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
# disclaimer. Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided
# with the distribution.
#
# Neither the name Myanmar Karen Word Lists, nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
# BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

# 7.  Database Ownership

#    The TZ database itself is not an IETF Contribution or an IETF document. Rather it is a pre-existing and regularly updated work that is in the public domain, and is intended to remain in the public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do not apply to the TZ Database or contributions that individuals make to it. Should any claims be made and substantiated against the TZ Database, the organization that is providing the IANA Considerations defined in this RFC, under the memorandum of understanding with the IETF, currently ICANN, may act in accordance with all competent court orders. No ownership claims will be made by ICANN or the IETF Trust on the database or the code. Any person making a contribution to the database or code waives all rights to future claims in that contribution or in the TZ Database.

Copyright (c) 2002,2003,2004,2005,2006 Marc Prud'hommeaux
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote
products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2012 Terence Parr and Sam Harwell
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Hive Metastore

Copyright 2008-2017 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE Version 1.0 (CDDL-1.0)

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the
2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the
Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.
6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible
for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

Copyright (c) 2002-2006, Marc Prud'hommeaux <mwp1@cornell.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2004-2013 QOS.ch
All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2003-2008, Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability contains
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1467 lodash-padstart 4.6.1

1.1467.1 Available under license:
Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

**1.1468 mongodb 2.2.36**

**1.1468.1 Available under license:**

MongoDB-core uses third-party libraries or other resources that may be distributed under licenses different than the MongoDB-core software.

In the event that we accidentally failed to list a required notice, please bring it to our attention through any of the ways detailed here:

mongodb-dev@googlegroups.com

The attached notices are provided for information only.

For any licenses that require disclosure of source, sources are available at https://github.com/mongodb/node-mongodb-native.

1) License Notice for nan.h

-----------------------------

The MIT License (MIT)

-----------------------------

Copyright (c) 2016 NAN contributors

-----------------------------

NAN contributors listed at <https://github.com/nodejs/nan#contributors>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
   Version 2.0, January 2004
   http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed.
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1469 jsreport-handlebars 2.1.0

1.1469.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.
A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)


You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based
on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
1.1470 regjsparsen 0.1.5
1.1470.1 Available under license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1471 metrics-servlets 4.0.2

1.1472 lowercase-keys 1.0.1 1.0.1
1.1472.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1473 cycle 1.0.3
1.1473.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
cycle.js
2013-02-19

Public Domain.

NO WARRANTY EXPRESSED OR IMPLIED. USE AT YOUR OWN RISK.

This code should be minified before deployment.
See http://javascript.crockford.com/jsmin.html

USE YOUR OWN COPY. IT IS EXTREMELY UNWISE TO LOAD CODE FROM SERVERS YOU DO
NOT CONTROL.
*/

Found in path(s):
* /opt/cola/permits/1112087439_1607035176.04/0/dscape-cycle-1-0-3-0-g9123ac0-tar-gz/dscape-cycle-9123ac0/cycle.js

1.1474 opentype-js 1.1.0
1.1474.1 Available under license :
The MIT License (MIT)

Copyright (c) 2017 Frederik De Bleser

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

Open Source Used In DNAC Platform Antelope 5172
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1475.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1476 snmp-4j 2.6.3
1.1476.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - JavaLogAdapter.java
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
*/
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/WorkerPool.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - MessageProcessingModel.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/MessageProcessingModel.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - SecureTarget.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##
_##########################################################################*/
Open Source Used In DNAC Platform Antelope 5176

You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/SecureTarget.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
# SNMP4J 2 - AssignableFromArray.java
# Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
# Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

/*_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/AssignableFromIntArray.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
# SNMP4J 2 - StatusInformation.java
# Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
# Licensed under the Apache License, Version 2.0 (the "License");
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/mp/StatusInformation.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - DHOps.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/security/dh/DHOps.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - DummyTransport.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - TsmSecurityStateReference.java

Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - AuthHMAC192SHA256.java
Open Source Used In DNAC Platform Antelope  5179

### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
###   http://www.apache.org/licenses/LICENSE-2.0
###
### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
### limitations under the License.
###
#############################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/AuthHMAC192SHA256.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - SnmpUriCallback.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##   http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/uri/SnmpUriCallback.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##
_##########################################################################*/
SNMP4J 2 - AuthHMAC128SHA224.java

---

SNMP4J 2 - AuthHMAC128SHA224.java

---

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

---

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

---

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---

SNMP4J 2 - Snmp.java

---

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

---

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

---

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---

Open Source Used In DNAC Platform Antelope 5180
/*_############################################################################
_##  SNMP4J 2 - MessageDispatcher.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/MessageDispatcher.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - SNMP4JSettings.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/MessageDispatcher.java
No license file was found, but licenses were detected in source scan.
/* ##########################################################################
 ## SNMP4J 2 - AuthenticationProtocol.java
 ##
 ## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
 ##
 ## Licensed under the Apache License, Version 2.0 (the "License");
 ## you may not use this file except in compliance with the License.
 ## You may obtain a copy of the License at
 ##
 ##   http://www.apache.org/licenses/LICENSE-2.0
 ##
 ## Unless required by applicable law or agreed to in writing, software
 ## distributed under the License is distributed on an "AS IS" BASIS,
 ## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 ## See the License for the specific language governing permissions and
 ## limitations under the License.
 ##
 ########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/AuthenticationProtocol.java
No license file was found, but licenses were detected in source scan.

/* ##########################################################################
 ## SNMP4J 2 - RetrievalEvent.java
 ##
 ## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
 ##
 ## Licensed under the Apache License, Version 2.0 (the "License");
 ## you may not use this file except in compliance with the License.
 ## You may obtain a copy of the License at
 ##
 ##   http://www.apache.org/licenses/LICENSE-2.0
 ##
 ## Unless required by applicable law or agreed to in writing, software
 ## distributed under the License is distributed on an "AS IS" BASIS,
 ## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 ## See the License for the specific language governing permissions and
 ## limitations under the License.
 ##
 ########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/RetrievalEvent.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - PrivacyProtocol.java
_## Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_## http://www.apache.org/licenses/LICENSE-2.0
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_## _##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/security/PrivacyProtocol.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - DefaultUdpTransportMapping.java
_## Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_## http://www.apache.org/licenses/LICENSE-2.0
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_## _##########################################################################*/

Found in path(s):
No license file was found, but licenses were detected in source scan.

/* ##############################################################################
   ##  SNMP4J 2 - DefaultTlsTmSecurityCallback.java
   ##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
   ##  Licensed under the Apache License, Version 2.0 (the "License");
   ##  you may not use this file except in compliance with the License.
   ##  You may obtain a copy of the License at
   ##      http://www.apache.org/licenses/LICENSE-2.0
   ##
   ##  Unless required by applicable law or agreed to in writing, software
   ##  distributed under the License is distributed on an "AS IS" BASIS,
   ##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   ##  See the License for the specific language governing permissions and
   ##  limitations under the License.
   ##
   ########################################################################*/

Found in path(s):

/* ##############################################################################
   ##  SNMP4J 2 - ResponseEvent.java
   ##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
   ##  Licensed under the Apache License, Version 2.0 (the "License");
   ##  you may not use this file except in compliance with the License.
   ##  You may obtain a copy of the License at
   ##      http://www.apache.org/licenses/LICENSE-2.0
   ##
   ##  Unless required by applicable law or agreed to in writing, software
   ##  distributed under the License is distributed on an "AS IS" BASIS,
   ##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   ##  See the License for the specific language governing permissions and
   ##  limitations under the License.
   ##
   ########################################################################*/
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/event/ResponseEvent.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - UsmTimeEntry.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/UsmTimeEntry.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - Salt.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_########################################################################*/
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/Salt.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/

  _##  SNMP4J 2 - TransportMapping.java
  _##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
  _##  Licensed under the Apache License, Version 2.0 (the "License");
  _##  you may not use this file except in compliance with the License.
  _##  You may obtain a copy of the License at
  _##  http://www.apache.org/licenses/LICENSE-2.0
  _##
  _##  Unless required by applicable law or agreed to in writing, software
  _##  distributed under the License is distributed on an "AS IS" BASIS,
  _##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  _##  See the License for the specific language governing permissions and
  _##  limitations under the License.
  _##
  _##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/TransportMapping.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/

  _##  SNMP4J 2 - SimpleVariableTextFormat.java
  _##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
  _##  Licensed under the Apache License, Version 2.0 (the "License");
  _##  you may not use this file except in compliance with the License.
  _##  You may obtain a copy of the License at
  _##  http://www.apache.org/licenses/LICENSE-2.0
  _##
  _##  Unless required by applicable law or agreed to in writing, software
  _##  distributed under the License is distributed on an "AS IS" BASIS,
  _##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  _##  See the License for the specific language governing permissions and
  _##  limitations under the License.
  _##
  _##########################################################################*/
Open Source Used In DNAC Platform Antelope 5188

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/SimpleVariableTextFormat.java

No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/

### SNMP4J 2 - ScopedPDU.java
###
### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
### http://www.apache.org/licenses/LICENSE-2.0
###
### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
### limitations under the License.

/*_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/ScopedPDU.java

No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/

### SNMP4J 2 - NonStandardSecurityProtocol.java
###
### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
### http://www.apache.org/licenses/LICENSE-2.0
###
### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
### limitations under the License.
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/nonstandard/NonStandardSecurityProtocol.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
##_
##_ SNMP4J 2 - USM.java
##_
##_ Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##_
##_ Licensed under the Apache License, Version 2.0 (the "License");
##_ you may not use this file except in compliance with the License.
##_ You may obtain a copy of the License at
##_
##_ http://www.apache.org/licenses/LICENSE-2.0
##_
##_ Unless required by applicable law or agreed to in writing, software
##_ distributed under the License is distributed on an "AS IS" BASIS,
##_ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##_ See the License for the specific language governing permissions and
##_ limitations under the License.
##_
##_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/USM.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
##_
##_ SNMP4J 2 - AuthSHA2.java
##_
##_ Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##_
##_ Licensed under the Apache License, Version 2.0 (the "License");
##_ you may not use this file except in compliance with the License.
##_ You may obtain a copy of the License at
##_
##_ http://www.apache.org/licenses/LICENSE-2.0
##_
##_ Unless required by applicable law or agreed to in writing, software
##_ distributed under the License is distributed on an "AS IS" BASIS,
##_ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##_ See the License for the specific language governing permissions and
Open Source Used in DNAC Platform Antelope  5190

#########################################################################
## limitations under the License.
##
## SNMJP4 2 - SimpleOIDTextFormat.java
##
## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
##
#########################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/AuthSHA2.java
No license file was found, but licenses were detected in source scan.

#########################################################################
## SNMP4J 2 - TransportStateListener.java
##
## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##
#########################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/SimpleOIDTextFormat.java
No license file was found, but licenses were detected in source scan.
See the License for the specific language governing permissions and limitations under the License.

Licensed under the Apache License, Version 2.0 (the "License"; you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
### SNMP4J 2 - MaxAccess.java

#### Copyright (C) 2003-2016 Frank Fock (SNMP4J.org)

#### Licensed under the Apache License, Version 2.0 (the "License");
#### you may not use this file except in compliance with the License.
#### You may obtain a copy of the License at
####
#### http://www.apache.org/licenses/LICENSE-2.0

#### Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.

---

### SNMP4J 2 - MessageID.java

#### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

#### Licensed under the Apache License, Version 2.0 (the "License");
#### you may not use this file except in compliance with the License.
#### You may obtain a copy of the License at
####
#### http://www.apache.org/licenses/LICENSE-2.0

#### Unless required by applicable law or agreed to in writing, software
Open Source Used In DNAC Platform Antelope  5194

_##_ distributed under the License is distributed on an "AS IS" BASIS,  
_##_ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
_##_ See the License for the specific language governing permissions and  
_##_ limitations under the License.  
_##_  
_####  
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/MessageID.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
_##_ SNMP4J 2 - Address.java
_##_  
_##_ Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##_  
_##_ Licensed under the Apache License, Version 2.0 (the "License");
_##_ you may not use this file except in compliance with the License.
_##_ You may obtain a copy of the License at
_##_  
_##_ http://www.apache.org/licenses/LICENSE-2.0
_##_  
_##_ Unless required by applicable law or agreed to in writing, software  
_##_ distributed under the License is distributed on an "AS IS" BASIS,  
_##_ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
_##_ See the License for the specific language governing permissions and  
_##_ limitations under the License.  
_##_  
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/Address.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
_##_ SNMP4J 2 - SshAddress.java
_##_  
_##_ Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##_  
_##_ Licensed under the Apache License, Version 2.0 (the "License");
_##_ you may not use this file except in compliance with the License.
_##_ You may obtain a copy of the License at
_##_  
_##_ http://www.apache.org/licenses/LICENSE-2.0
_##_  
_##_ Unless required by applicable law or agreed to in writing, software
Open Source Used In DNAC Platform Antelope 5195

# Distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
###########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/SshAddress.java
No license file was found, but licenses were detected in source scan.

/*############################################################################
## SNMP4J 2 - SecurityLevel.java
##
## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
##
############################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/SecurityLevel.java
No license file was found, but licenses were detected in source scan.

/*############################################################################
## SNMP4J 2 - LogLevel.java
##
## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
##
############################################################################*/
%== SMNP4J 2 - RequestStatistics.java
%==
%== Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
%==
%== Licensed under the Apache License, Version 2.0 (the "License");
%== you may not use this file except in compliance with the License.
%== You may obtain a copy of the License at
%==
%== http://www.apache.org/licenses/LICENSE-2.0
%==
%== Unless required by applicable law or agreed to in writing, software
%== distributed under the License is distributed on an "AS IS" BASIS,
%== WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
%== See the License for the specific language governing permissions and
%== limitations under the License.
%==

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/RequestStatistics.java
No license file was found, but licenses were detected in source scan.

%== SMNP4J 2 - BEROutputStream.java
%==
%== Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
%==
%== Licensed under the Apache License, Version 2.0 (the "License");
%== you may not use this file except in compliance with the License.
%== You may obtain a copy of the License at
%==
%== http://www.apache.org/licenses/LICENSE-2.0
%==

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/log/LogLevel.java
No license file was found, but licenses were detected in source scan.

%== SNMP4J 2 - RequestStatistics.java
%==
%== Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
%==
%== Licensed under the Apache License, Version 2.0 (the "License");
%== you may not use this file except in compliance with the License.
%== You may obtain a copy of the License at
%==
%== http://www.apache.org/licenses/LICENSE-2.0
%==

Found in path(s):
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - VariantVariable.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - SecurityModel.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - TimeTicks.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - SMIConstants.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

No license file was found, but licenses were detected in source scan.
Open Source Used In DNAC Platform Antelope 5201

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - SnmpConstants.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - BERInputStream.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

No license file was found, but licenses were detected in source scan.
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - OID.java

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - Priv3DES.java

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/Priv3DES.java
No license file was found, but licenses were detected in source scan.

/* ############################################################################
##
##  SNMP4J 2 - TcpTransportMapping.java
##
##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
##
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##  You may obtain a copy of the License at
##
##    http://www.apache.org/licenses/LICENSE-2.0
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##  See the License for the specific language governing permissions and
##  limitations under the License.
##
########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/transport/TcpTransportMapping.java
No license file was found, but licenses were detected in source scan.

/* ############################################################################
##
##  SNMP4J 2 - TransportIpAddress.java
##
##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
##
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##
########################################################################*/
## You may obtain a copy of the License at
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
##
##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/smi/TransportIpAddress.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - ConsoleLogFactory.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##  See the License for the specific language governing permissions and
##  limitations under the License.
##
##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/log/ConsoleLogFactory.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - ConsoleLogAdapter.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##

Open Source Used In DNAC Platform Antelope 5204
Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snpmp4j/log/ConsoleLogAdapter.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - SnmpRequest.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snpmp4j/tools/console/SnmpRequest.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - TableEvent.java
_INLINE_FUNCTION
_##
/*
_INLINE_FUNCTION
_PARAMETERS
_##
/*
## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
## http://www.apache.org/licenses/LICENSE-2.0

## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.

##########################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/MessageDispatcherImpl.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - SchedulerTask.java
_##
##
##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##  You may obtain a copy of the License at
##  http://www.apache.org/licenses/LICENSE-2.0
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##  See the License for the specific language governing permissions and
##  limitations under the License.
##
##########################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/SchedulerTask.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - TlsTmSecurityCallback.java

______________________________________________________________
### SNMP4J 2 - PrivAESWith3DESKeyExtension.java

### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
### http://www.apache.org/licenses/LICENSE-2.0

### Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1.jar/org/snmp4j/security/nonstandard/PrivAESWith3DESKeyExtension.java
No license file was found, but licenses were detected in source scan.

/* _############################################################################
#
## SNMP4J 2 - UsmUserEvent.java
#
## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
##
########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1.jar/org/snmp4j/event/UsmUserEvent.java
No license file was found, but licenses were detected in source scan.
/* _############################################################################
  ## SNMP4J 2 - PrivAES.java
  ##
  ## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
  ##
  ## Licensed under the Apache License, Version 2.0 (the "License");
  ## you may not use this file except in compliance with the License.
  ## You may obtain a copy of the License at
  ##
  ##    http://www.apache.org/licenses/LICENSE-2.0
  ##
  ## Unless required by applicable law or agreed to in writing, software
  ## distributed under the License is distributed on an "AS IS" BASIS,
  ## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  ## See the License for the specific language governing permissions and
  ## limitations under the License.
  ##
  _##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/PrivAES.java
No license file was found, but licenses were detected in source scan.

/* _############################################################################
  ## SNMP4J 2 - AuthHMAC256SHA384.java
  ##
  ## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
  ##
  ## Licensed under the Apache License, Version 2.0 (the "License");
  ## you may not use this file except in compliance with the License.
  ## You may obtain a copy of the License at
  ##
  ##    http://www.apache.org/licenses/LICENSE-2.0
  ##
  ## Unless required by applicable law or agreed to in writing, software
  ## distributed under the License is distributed on an "AS IS" BASIS,
  ## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  ## See the License for the specific language governing permissions and
  ## limitations under the License.
  ##
  _##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/AuthHMAC256SHA384.java
No license file was found, but licenses were detected in source scan.

/* _############################################################################
### SNMP4J 2 - Gauge32.java
###
### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
### http://www.apache.org/licenses/LICENSE-2.0
###
### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
### limitations under the License.
###
########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snp4j/smi/Gauge32.java

No license file was found, but licenses were detected in source scan.

/* _############################################################################
### SNMP4J 2 - PropertiesTlsTmSecurityCallback.java
###
### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
### http://www.apache.org/licenses/LICENSE-2.0
###
### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
### limitations under the License.
###
########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - AbstractSnmpUtility.java
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
*/opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/util/AbstractSnmpUtility.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - TableUtils.java
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - TransportStateEvent.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - ThreadPool.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/ThreadPool.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - Integer32.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/Integer32.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - LogProxy.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_##########################################################################*/
/*_################################################################################
_## SNMP4J 2 - AuthHMAC384SHA512.java
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_################################################################################*/

/*_############################################################################
_## SNMP4J 2 - TsmSecurityParameters.java
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_############################################################################*/
Open Source Used In DNAC Platform Antelope  5216

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/security/TsmSecurityParameters.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
_##
_## SNMP4J 2 - DefaultThreadFactory.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_##      http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/util/DefaultThreadFactory.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
_##
_## SNMP4J 2 - Counter32.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_##      http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
No license file was found, but licenses were detected in source scan.

/* _############################################################################
_##
_## SNMP4J 2 - UsmSecurityStateReference.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/SMIAddress.java
No license file was found, but licenses were detected in source scan.

/* _##########################################################################*/
_##
_## SNMP4J 2 - SMIAddress.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/Counter32.java
No license file was found, but licenses were detected in source scan.
Open Source Used In DNAC Platform Antelope  5218

/*_############################################################################
_##
_## SNMP4J 2 - SnmpUriResponse.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/security/UsmSecurityStateReference.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_## SNMP4J 2 - BitString.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/Target.java

No license file was found, but licenses were detected in source scan.

/*  ####################################################################################################
  #
  # SNMP4J 2 - TLSTMExtendedTrustManager.java
  #
  # Copyright (C) 2003-2017  Frank Fock and Jochen Katz (SNMP4J.org)
  #
  # Licensed under the Apache License, Version 2.0 (the "License");
  # you may not use this file except in compliance with the License.
  # You may obtain a copy of the License at
  #
  # http://www.apache.org/licenses/LICENSE-2.0
  #
  # Unless required by applicable law or agreed to in writing, software
  # distributed under the License is distributed on an "AS IS" BASIS,
  # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  # See the License for the specific language governing permissions and
  # limitations under the License.
  #
  # ####################################################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/transport/tls/TLSTMExtendedTrustManager.java

No license file was found, but licenses were detected in source scan.

/*  ####################################################################################################
  #
  # SNMP4J 2 - VariantVariableCallback.java
  #
  # Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
  #
  # Licensed under the Apache License, Version 2.0 (the "License");
  # you may not use this file except in compliance with the License.
  # You may obtain a copy of the License at
  #
  # http://www.apache.org/licenses/LICENSE-2.0
  #
  # Unless required by applicable law or agreed to in writing, software
  # distributed under the License is distributed on an "AS IS" BASIS,
  # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  # See the License for the specific language governing permissions and
  # limitations under the License.
  #
  # ####################################################################################################*/
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

########################################################################/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/smi/VariantVariableCallback.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
##
## SNMP4J 2 - SecurityModels.java
##
## Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
##
##########################################################################/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/security/SnmpConfigurator.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
##
## SNMP4J 2 - SnmpConfigurator.java
##
## Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0

Open Source Used In DNAC Platform Antelope  5221
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/SnmpConfigurator.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - ReadonlyVariableCallback.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##   http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
############################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/ReadonlyVariableCallback.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - ThreadFactory.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
############################################################################*/
If you have any questions or need further assistance, please feel free to contact us. We look forward to hearing from you.

---

Open Source Used in DNAC Platform Antelope 5223

---

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - Opaque.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
##_ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/Opaque.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - MultiThreadedTrapReceiver.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/ThreadFactory.java

No license file was found, but licenses were detected in source scan.
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/test/MultiThreadedTrapReceiver.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - PrivAES256With3DESKeyExtension.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/security/nonstandard/PrivAES256With3DESKeyExtension.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - TimeoutModel.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
##########################################################################*/
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/TimeoutModel.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - TimedMessageID.java
_##
_## Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_############################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/TimedMessageID.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - DefaultTimeoutModel.java
_##
_## Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_############################################################################*/
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - BER.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - CommandResponderEvent.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1.jar/org/snmp4j/CommandResponderEvent.java
No license file was found, but licenses were detected in source scan.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1.jar/org/snmp4j/log/LogFactory.java
No license file was found, but licenses were detected in source scan.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1.jar/org/snmp4j/log/LogFactory.java
No license file was found, but licenses were detected in source scan.
Open Source Used In DNAC Platform Antelope 5228

_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)_
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/PDUFactory.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_## SNMP4J 2 - Null.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/Null.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_## SNMP4J 2 - EnumerationIterator.java
_##
_##
_##########################################################################*/
/*_############################################################################
_##  SNMP4J 2 - SecurityParameters.java
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##   http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/security/SecurityParameters.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - TreeListener.java
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##   http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/TreeListener.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - PrivAES192.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/PrivAES192.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - DefaultTcpTransportMapping.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/transport/DefaultTcpTransportMapping.java
No license file was found, but licenses were detected in source scan.

/* _############################################################################
_##  SNMP4J 2 - UdpAddress.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1.jar/org/snmp4j/smi/UdpAddress.java
No license file was found, but licenses were detected in source scan.

/* _############################################################################
_##  SNMP4J 2 - SecurityStateReference.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/SecurityStateReference.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - SnmpEngineListener.java
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License";
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
############################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/event/SnmpEngineListener.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - Variable.java
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License";
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
############################################################################*/
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - SubIndexInfoImpl.java
_##  Copyright (C) 2003-2016  Frank Fock (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
########################################################################*/

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - AssignableFromByteArray.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/AssignableFromByteArray.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/

_found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/VariableBinding.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/

_found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/UsmTimeTable.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
SNMP4J 2 - User.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - LogAdapter.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
No license file was found, but licenses were detected in source scan.

```
/*_##########################################################################*/
## SNMP4J 2 - CommunityTarget.java
##
## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
##
##############################################################################
```

No license file was found, but licenses were detected in source scan.

```
/*_##########################################################################*/
## SNMP4J 2 - TcpAddress.java
##
## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
## Licensed under the Apache License, Version 2.0 (the "License");
## you may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
```
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/TcpAddress.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
##
##  SNMP4J 2 - EngineIdCacheSize.java
##
##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##  You may obtain a copy of the License at
##
##  http://www.apache.org/licenses/LICENSE-2.0
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##
##  See the License for the specific language governing permissions and
##  limitations under the License.
##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/EngineIdCacheSize.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
##
##  SNMP4J 2 - AuthSHA.java
##
##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
##
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##  You may obtain a copy of the License at
##
##  http://www.apache.org/licenses/LICENSE-2.0
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##########################################################################*/
/*_############################################################################
_##  SNMP4J 2 - PDUv1.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
/*_############################################################################
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/PDUv1.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/AuthSHA.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
_##  SNMP4J 2 - MPv2c.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
/*_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/MPv2c.java
No license file was found, but licenses were detected in source scan.

Open Source Used In DNAC Platform Antelope 5241
See the License for the specific language governing permissions and
limitations under the License.

licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.

licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - VariableTextFormat.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - OIDTextFormat.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
SNMP4J 2 - PDU.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - DefaultPDUFactory.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
ducted under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/DefaultPDUFactory.java
  No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - TimerFactory.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
_##
_############################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/TimerFactory.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################
_##  SNMP4J 2 - AuthMD5.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##########################################################################*/
Open Source Used In DNAC Platform Antelope 5246

## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.

##########################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/AuthMD5.java
No license file was found, but licenses were detected in source scan.

/* _##########################################################################
_##  SNMP4J 2 - TableListener.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/TableListener.java
No license file was found, but licenses were detected in source scan.

/* _##########################################################################
_##  SNMP4J 2 - AssignableFromInteger.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.40/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/TableListener.java
No license file was found, but licenses were detected in source scan.

/* _##########################################################################
_##  SNMP4J 2 - AssignableFromInteger.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##########################################################################/
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/AssignableFromInteger.java
No license file was found, but licenses were detected in source scan.

SNMP4J 2 - Log4jLogFactory.java
---
Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/log/Log4jLogFactory.java
No license file was found, but licenses were detected in source scan.

SNMP4J 2 - SecurityProtocols.java
---
Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
### limitations under the License.
###
#############################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/security/SecurityProtocols.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - AuthenticationFailureEvent.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/event/AuthenticationFailureEvent.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - SshTransportAdapter.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
## Apache License, Version 2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---

### SNMP4J 2 - DefaultSshTransportMapping.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---

### SNMP4J 2 - Session.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/TlsAddress.java
     No license file was found, but licenses were detected in source scan.

/***************************************************************************
### SNMP4J 2 - TlsAddress.java
###
### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

###
### Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
###
/***************************************************************************

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/TreeEvent.java
     No license file was found, but licenses were detected in source scan.

/***************************************************************************
### SNMP4J 2 - TreeEvent.java
###
### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/TreeEvent.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - TSM.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/TSM.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - LogControl.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
## Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - PrivAES192With3DESKeyExtension.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - PrivAES256.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - MultiThreadedMessageDispatcher.java

Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
SNMP4J 2 - AssignableFromString.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - AuthGeneric.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - AssignableFromString.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
/*_############################################################################
_##  SNMP4J 2 - SNMPv3SecurityModel.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##      http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/security/SNMPv3SecurityModel.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - TLSTMExtendedTrustManagerFactory.java
_##
_##  Copyright (C) 2003-2017 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##      http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/transport/tls/TLSTMExtendedTrustManagerFactory.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - PduHandle.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/PduHandle.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - CertifiedTarget.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/CertifiedTarget.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - StateReference.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/StateReference.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - DHParameters.java
_##
_##  Copyright (C) 2003-2017 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/security/db/DHParameters.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - SecurityProtocol.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/security/SecurityProtocol.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - ByteArrayWindow.java
_##
_##  Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##    http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

Open Source Used In DNAC Platform Antelope 5260
No license file was found, but licenses were detected in source scan.

/*_############################################################################
##  SNMP4J 2 - SecurityNameMapping.java
##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##  You may obtain a copy of the License at
##  
##  http://www.apache.org/licenses/LICENSE-2.0
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##  See the License for the specific language governing permissions and
##  limitations under the License.
##
_########################################################################*/

No license file was found, but licenses were detected in source scan.

/*_############################################################################
##  SNMP4J 2 - Log4jLogAdapter.java
##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##  You may obtain a copy of the License at
##  
##  http://www.apache.org/licenses/LICENSE-2.0
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##  See the License for the specific language governing permissions and
##  limitations under the License.
##
_########################################################################*/
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/log/Log4jLogAdapter.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
### SNMP4J 2 - UsmUserTable.java
###
### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
### http://www.apache.org/licenses/LICENSE-2.0
###
### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
### limitations under the License.
###
###_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/UsmUserTable.java
No license file was found, but licenses were detected in source scan.

/*_##########################################################################*/
### SNMP4J 2 - MPv3.java
###
### Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
###
### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
### http://www.apache.org/licenses/LICENSE-2.0
###
### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/mp/MPv3.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
##
##  SNMP4J 2 - UnsignedInteger32.java
##
##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
##
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##  You may obtain a copy of the License at
##
##     http://www.apache.org/licenses/LICENSE-2.0
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##
##  See the License for the specific language governing permissions and
##  limitations under the License.
##
_##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/UnsignedInteger32.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
##
##  SNMP4J 2 - AssignableFromLong.java
##
##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
##
##  Licensed under the Apache License, Version 2.0 (the "License");
##  you may not use this file except in compliance with the License.
##  You may obtain a copy of the License at
##
##     http://www.apache.org/licenses/LICENSE-2.0
##
##  Unless required by applicable law or agreed to in writing, software
##  distributed under the License is distributed on an "AS IS" BASIS,
##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
No license file was found, but licenses were detected in source scan.

2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - MPv1.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  _##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
_##  limitations under the License.
_##
_##########################################################################*/

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - ArgumentParser.java
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##  _##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_##  See the License for the specific language governing permissions and
Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/util/ArgumentParser.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - AbstractTransportMapping.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
##
##
##########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/transport/AbstractTransportMapping.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##  SNMP4J 2 - OctetString.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
_##  you may not use this file except in compliance with the License.
_##  You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software
_##  distributed under the License is distributed on an "AS IS" BASIS,
_##  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
/// See the License for the specific language governing permissions and
/// limitations under the License.

########################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/OctetString.java
No license file was found, but licenses were detected in source scan.

/*_########################################################################
_
_## SNMP4J 2 - CertifiedIdentity.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/CertifiedIdentity.java
No license file was found, but licenses were detected in source scan.

/*_########################################################################
_
_## SNMP4J 2 - UsmUserListener.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
Open Source Used In DNAC Platform Antelope  

See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - IpAddress.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SNMP4J 2 - AbstractVariable.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.

########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/smi/AbstractVariable.java
No license file was found, but licenses were detected in source scan.

/* ########################################################################

# SNMP4J 2 - UsmUserEntry.java
#
# Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-jar/org/snmp4j/security/UsmUserEntry.java
No license file was found, but licenses were detected in source scan.

/* ########################################################################

# SNMP4J 2 - Counter64.java
#
# Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - UserTarget.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
##_############################################################################

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - MutableStateReference.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
##_############################################################################

No license file was found, but licenses were detected in source scan.
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - MutablePDU.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SNMP4J 2 - SnmpEngineEvent.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
__##
__## Unless required by applicable law or agreed to in writing, software
__## distributed under the License is distributed on an "AS IS" BASIS,
__## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
__## See the License for the specific language governing permissions and
__## limitations under the License.
__##
###
############################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/event/SnmpEngineEvent.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - UdpTransportMapping.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
############################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/transport/UdpTransportMapping.java
No license file was found, but licenses were detected in source scan.

/*_############################################################################
_## SNMP4J 2 - PduHandleCallback.java
_##
_## Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
_##
_## Licensed under the Apache License, Version 2.0 (the "License");
_## you may not use this file except in compliance with the License.
_## You may obtain a copy of the License at
_##
_## http://www.apache.org/licenses/LICENSE-2.0
_##
_## Unless required by applicable law or agreed to in writing, software
_## distributed under the License is distributed on an "AS IS" BASIS,
_## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
_## See the License for the specific language governing permissions and
_## limitations under the License.
_##
############################################################################

Open Source Used In DNAC Platform Antelope 5274

### http://www.apache.org/licenses/LICENSE-2.0

### Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

##########################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1.jar/org/snmp4j/mp/PduHandleCallback.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - MessageLength.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
_##
_##  http://www.apache.org/licenses/LICENSE-2.0
_##
_##  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
_##
_##########################################################################

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1.jar/org/snmp4j/transport/MessageLength.java

No license file was found, but licenses were detected in source scan.

/*_############################################################################
_##
_##  SNMP4J 2 - TransportListener.java
_##
_##  Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)
_##
_##  Licensed under the Apache License, Version 2.0 (the "License");
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/event/AuthenticationFailureListener.java
No license file was found, but licenses were detected in source scan.

SNMP4J 2 - UnsupportedAddressClassException.java
Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
  jar/org/snmp4j/transport/UnsupportedAddressClassException.java
No license file was found, but licenses were detected in source scan.

SNMP4J 2 - MessageLengthDecoder.java
SNMP4J 2 - PrivAES128.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

No license file was found, but licenses were detected in source scan.

SNMP4J 2 - MessageException.java

Copyright (C) 2003-2016 Frank Fock and Jochen Katz (SNMP4J.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

No license file was found, but licenses were detected in source scan.
## SNMP4J 2 - CounterListener.java

### Copyright (C) 2003-2016  Frank Fock and Jochen Katz (SNMP4J.org)

### Licensed under the Apache License, Version 2.0 (the "License");
### you may not use this file except in compliance with the License.
### You may obtain a copy of the License at
###
###     http://www.apache.org/licenses/LICENSE-2.0

### Unless required by applicable law or agreed to in writing, software
### distributed under the License is distributed on an "AS IS" BASIS,
### WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
### See the License for the specific language governing permissions and
### limitations under the License.

########################################################################*/

Found in path(s):
* /opt/cola/permits/1136700372_1613864639.4/0/snmp4j-2-6-3-sources-1-
jar/org/snmp4j/event/CounterListener.java

1.1477 model-mapper 1.1.0

1.1478 which 2.0.2

1.1478.1 Available under license :

MIT License

Copyright (c) 2017 Cédric Belin <cedric@belin.io>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
governed permissions and
limitations under the License.

###

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

Open Source Used In DNAC Platform Antelope 5279
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1479 os-locale 3.1.0
1.1479.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1480 mitt 1.1.3
1.1480.1 Available under license :
The MIT License (MIT)

Copyright (c) 2017 Jason Miller

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1481 @babelgenerator 7.13.9
1.1481.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1482 @babel/helper-function-name 7.12.13
1.1482.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1483 object-copy 0.1.0
1.1483.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1484 bitsyntax 0.1.0
1.1484.1 Available under license:
The MIT License (MIT)

Copyright (c) 2012-2014 Michael Bridgen <mikeb@squaremobius.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

bitsyntax-js copyright (c) 2012-2014 Michael Bridgen <mikeb@squaremobius.net>

This package, "bitsyntax-js", is licensed under the MIT License. A copy may be found in the file LICENSE-MIT in this directory, or downloaded from http://opensource.org/licenses/MIT.

1.1485 jackson-annotations 2.11.4

1.1485.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1486 chalk 1.1.3

1.1486.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1487 transaction-api 1.1

1.1487.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
Version 1.0

* 

1. Definitions.

0

1.1. ?Contributor? means each individual or entity that creates or contributes to the creation of Modifications.

0

1.2. ?Contributor Version? means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

0

1.3. ?Covered Software? means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

0

1.4. ?Executable? means the Covered Software in any form other than Source Code.

0

1.5. ?Initial Developer? means the individual or entity that first makes Original Software available under this License.

0

1.6. ?Larger Work? means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

0

1.7. ?License? means this document.

0

1.8. ?Licensable? means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. ?Modifications? means the Source Code and Executable form of any of the following:
   +
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   +
   B. Any new file that contains any part of the Original Software or previous Modification; or
   +
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. ?Original Software? means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. ?Patent Claims? means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. ?Source Code? means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. ?You? (or ?Your?) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, ?You? includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, ?control? means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

   Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
   +
   (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
   +
   (b) under Patent Claims Licensable by Initial Developer.
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.
6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software? (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation? as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and
9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

1.1488 jakarta-validation-api 2.0.2
1.1488.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Jakarta Bean Validation API
 *
 * License: Apache License, Version 2.0
 * See the license.txt file in the root directory or <http://www.apache.org/licenses/LICENSE-2.0>.
 */

Found in path(s):
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3-jar/javax/validation/Validator.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3-jar/javax/validation/metadata/CascadableDescriptor.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3-jar/javax/validation/ConstraintTarget.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/constraints/FutureOrPresent.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/constraints/Null.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/ConstraintValidatorContext.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/metadata/ExecutableDescriptor.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/groups/Default.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/MessageInterpolator.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/metadata/package-info.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/valueextraction/Unwrapping.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/valueextraction/ValueExtractor.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/constraintvalidation/ValidationTarget.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/metadata/MethodDescriptor.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/package-info.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/constraints/Digits.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/bootstrap/ProviderSpecificBootstrap.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/valueextraction/UnwrapByDefault.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/constraints/NegativeOrZero.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/ConstraintViolation.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/ConstraintDefinitionException.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/metadata/ValidateUnwrappedValue.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/constraintvalidation/SupportedValidationTarget.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/groups/package-info.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2-0-2-sources-3.jar/javax/validation/spi/ValidationProvider.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/ParameterNameProvider.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/constraints/Pattern.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/validation/NoProviderFoundException.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/Validation.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/ValidatorContext.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/constraints/NotBlank.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/spi/package-info.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/constraints/Size.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/metadata/ContainerDescriptor.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/metadata/ConstraintDescriptor.java
* /opt/cola/permits/1135880235_1613624044.2/0/jakarta-validation-api-2.0.2-sources-3.jar/javax/validation/metadata/ParameterDescriptor.java
No license file was found, but licenses were detected in source scan.

~ Jakarta Bean Validation API ~
1.1489 ansi-regex 2.1.1

1.1489.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1490 ncp 2.0.0 2.0.0

1.1490.1 Available under license :

# MIT License

###Copyright (C) 2011 by Charlie McConnell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1491 jsreport-scheduling 2.5.0
1.1491.1 Available under license :
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
1.1492.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1493 jaxb-runtime 2.3.1

1.1494 jetty-http-spi 9.3.27.v20190418

1.1495 chalk 2.4.2
1.1495.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1496 compression 1.6.2

1.1496.1 Available under license:
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1497 ignite-shmem 1.0.0

1.1497.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensorn shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed...
as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

================================================================================

Apache Ignite (incubating) Subcomponents:
The Apache Ignite project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

==============================================================================
For the pcollections library (https://github.com/blackdrag/pcollections)
==============================================================================

The MIT License (MIT)

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

==============================================================================
For SnapTree:
==============================================================================

BSD LICENSE

Copyright (c) 2009 Stanford University, unless otherwise specified.
All rights reserved.

This software was developed by the Pervasive Parallelism Laboratory of Stanford University, California, USA.

Permission to use, copy, modify, and distribute this software in source or binary form for any purpose with or without fee is hereby granted, provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of Stanford University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1498 registry-auth-token 4.2.1

1.1498.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Espen Hovlandsdal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1499 anymatch 1.3.2
1.1499.1 Available under license:
The ISC License

Copyright (c) 2014 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1500 js-tokens 3.0.2 3.0.2
1.1500.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014, 2015, 2016, 2017 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1501 lodash 4.17.15 4.17.15

1.1501.1 Available under license:

Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1502 is-negative-zero 2.0.1

1.1502.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1503 mime 1.4.1 1.4.1

1.1503.1 Available under license :

The MIT License (MIT)

Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.1504 axios 0.18.1 0.18.1

1.1504.1 Available under license :
Copyright (c) 2014-present Matt Zabriskie

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1505 unzipper 0.9.15

1.1505.1 Available under license :
Copyright (c) 2012 - 2013 Near Infinity Corporation

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

Commits in this fork are (c) Ziggy Jonsson (ziggy.jonsson.nyc@gmail.com) and fall under same licence structure as the original repo (MIT)

1.1506 ejs 2.6.2

1.1506.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1507 babel-helper-remap-async-to-generator 6.24.1 6.24.1
1.1507.1 Available under license :
   The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1508 negotiator 0.6.1 0.6.1
1.1508.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schlueter
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1509 coffee-script 2.5.1 2.5.1
1.1509.1 Available under license:
Copyright (c) 2009-2018 Jeremy Ashkenas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1510 pause-stream 0.0.11 0.0.11
1.1510.1 Available under license :

Dual Licensed MIT and Apache 2

The MIT License

Copyright (c) 2013 Dominic Tarr

Permission is hereby granted, free of charge,
to any person obtaining a copy of this software and
associated documentation files (the "Software"), to
deal in the Software without restriction, including
without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom
the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice
shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----------------------------------------------------------------------

Apache License
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2013 Dominic Tarr

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1511 commons-compress 1.14

1.1511.1 Available under license:

Apache Commons Compress
Copyright 2002-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

The files in the package org.apache.commons.compress.archivers.sevenz
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

http://www.apache.org/licenses/LICENSE-2.0
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1512 js-cookie 2.1.1

1.1512.1 Available under license:

Copyright 2014 Klaus Hartl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
Apache Tomcat
Copyright 1999-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1514 antlr-runtime 4.8-1
1.1514.1 Available under license :

No license file was found, but licenses were detected in source scan.

~ Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/META-INF/maven/org.antlr/antlr4-runtime/pom.xml
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.
 * Use of this file is governed by the BSD 3-clause license that
 * can be found in the LICENSE.txt file in the project root.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/misc/Array2DHashSet.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/dfa/DFASerializer.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/FailedPredicateException.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/tree/pattern/TextChunk.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/Lexer.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/VocabularyImpl.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/LexerActionType.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/ATNState.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/tree/TerminalNodeImpl.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/WritableToken.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/PlusBlockStartState.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/StarBlockStartState.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/CharStream.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/PredictionContext.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/tree/TerminalNode.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/misc/LogManager.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/CodePointTransitions.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/NoViableAltException.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/LexerMoreAction.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/tree/AbstractParseTreeVisitor.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/LexerAction.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/NotSetTransition.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/ArrayPredictionContext.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/Recognizer.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/tree/xpath/XPathTokenElement.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/LexerNoViableAltException.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/antlr/v4/runtime/atn/LexerModeAction.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/ANTLRErrorListener.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/CommonTokenStream.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/atn/SetTransition.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/atn/ATNDeserializationOptions.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/atn/DecisionInfo.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/xpath/XPathTokenAnywhereElement.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/xpath/XPathRuleElement.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/ErrorNode.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/pattern/ParseTreeMatch.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/atn/RuleTransition.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/dfa/DFAState.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/ANTLRFileStream.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/Expression.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/error/ParseException.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/error/RecognitionException.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/ErrorNode.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/TreeStream.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/TreeStream.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/pattern/ParseTreeMatch.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources-jar/org/antlr/v4/runtime/tree/pattern/Par...
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/tree/pattern/Chunk.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/ParserInterpreter.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/atn/ATNSimulator.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/atn/AtomTransition.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/atn/ProfilingATNSimulator.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/misc/MultiMap.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/misc/Interval.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/tree/pattern/TagChunk.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/RuleContext.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/misc/Triple.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/atn/ATNConfigSet.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/tree/xpath/XPathElement.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/misc/NotNull.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/atn/AtomTransition.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/atn/WildcardTransition.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/atn/PredictionMode.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/misc/OrderedHashSet.java
* /opt/ws_local/PERMITS_SQL/1092523039_1600741420.05/0/antlr4-runtime-4-8-1-sources.jar/org/antlr/v4/runtime/misc/FlexibleHashMap.java

---

Open Source Used In DNAC Platform Antelope 5345
Open Source Used In DNAC Platform Antelope 5346
Open Source Used In DNAC Platform Antelope 5347
Open Source Used In DNAC Platform Antelope 5348
1.1515 cookie-parser 1.4.4

1.1515.1 Available under license:
(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1516 through2 2.0.5

1.1516.1 Available under license:
# The MIT License (MIT)

**Copyright (c) Rod Vagg (the "Original Author") and additional contributors**

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1517 babel-plugin-syntax-async-functions
6.13.0
1.1517.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1518 babel-plugin-transform-es2015-classes 6.24.1

1.1518.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party is a member of Ecma International.  SEE THE
Open Source Used In DNAC Platform Antelope 5354

ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1519 babel-helper-define-map 6.26.0

1.1519.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The "Software" identified by reference to the Ecma Standard is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, 
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND 
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE 
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION 
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION 
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining 
a copy of this software and associated documentation files (the 
"Software"), to deal in the Software without restriction, including 
without limitation the rights to use, copy, modify, merge, publish, 
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND 
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE 
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION 
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION 
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without 
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright 
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright 
  notice, this list of conditions and the following disclaimer in the 
documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1520 wrap-ansi 6.2.0 6.2.0

1.1520.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1521 ansi-escapes 3.2.0

1.1521.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1522 activation 1.1

1.1522.1 Available under license:

Found license 'Common Development and Distribution License (CDDL) 1.1' in 'The contents of this file are subject to the terms of the Common Development and Distribution License (the "License"). You may not use this file except Copyright 2001-2005 Sun Microsystems, Inc. All Rights Reserved.'
Found license 'Common Development and Distribution License (CDDL) 1.1' in "* The contents of this file are subject to the terms * of the Common Development and Distribution License * (the "License"). You may not use this file except * Copyright 2002-2005 Sun Microsystems, Inc. All Rights Reserved.'
Found license 'Common Development and Distribution License (CDDL) 1.1' in "* The contents of this file are subject to the terms * of the Common Development and Distribution License * (the "License"). You may not use this file except * Copyright 1997-2005 Sun Microsystems, Inc. All Rights Reserved.'

1.1523 undefsafe 2.0.3
1.1523.1 Available under license :
The MIT License (MIT)

Copyright 2016 Remy Sharp, http://remysharp.com <remy@remysharp.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1524 regexpu-core 4.7.1
1.1524.1 Available under license :
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1525 jsonwebtoken 7.4.3
1.1525.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Auth0, Inc. <support@auth0.com> (http://auth0.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1526 jersey-hk2 2.32
1.1526.1 Available under license:

# Notice for Jersey
This content is produced and maintained by the Eclipse Jersey project.

* Project home: https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms
of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

### Angular JS, v1.6.6
* License MIT (http://www.opensource.org/licenses/mit-license.php)
* Project: http://angularjs.org
* Copyright: (c) 2010-2017 Google, Inc.

### aopalliance Version 1
* License: all the source code provided by AOP Alliance is Public Domain.
* Copyright: Material in the public domain is not protected by copyright

### Bean Validation API 2.0.2
* License: Apache License, 2.0
* Project: http://beanvalidation.org/1.1/
* Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
  * by the @authors tag.

### Hibernate Validator CDI, 6.1.2.Final
* License: Apache License, 2.0
* Project: https://beanvalidation.org/
* Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

### Bootstrap v3.3.7
* License: MIT license (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* Project: http://getbootstrap.com
* Copyright: 2011-2016 Twitter, Inc

### Google Guava Version 18.0
* License: Apache License, 2.0
* Copyright (C) 2009 The Guava Authors

### javax.inject Version: 1
* License: Apache License, 2.0
* Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA
* License: Apache License, 2.0
* Project: http://www.javassist.org/
* Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.10.1
* License: Apache License, 2.0
* Project: https://github.com/FasterXML/jackson-jaxrs-providers
* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4
* License: jquery.org/license
* Project: jquery.org
* Copyright: (c) jQuery Foundation

jQuery Barcode plugin 0.3
* Project: http://www.pasella.it/projects/jquery/barcode
* Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266
* License: CC0
* No copyright
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

KineticJS, v4.7.1
* License: MIT license (http://www.opensource.org/licenses/mit-license.php)
* Copyright: Eric Rowell

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0
* License: Apache License, 2.0
* Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified
Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the
Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

   iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial
Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.
5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses...
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program

Open Source Used In DNAC Platform Antelope 5371
proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed
only for noncommercial distribution and only if you received the
program in object code or executable form with such an offer, in
accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source code
means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to control
compilation and installation of the executable. However, as a special
exception, the source code distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies the
executable.

If distribution of executable or object code is made by offering access
to copy from a designated place, then offering equivalent access to copy
the source code from the same place counts as distribution of the source
code, even though third parties are not compelled to copy the source
along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt otherwise
to copy, modify, sublicense or distribute the Program is void, and will
automatically terminate your rights under this License. However, parties
who have received copies, or rights, from you under this License will
not have their licenses terminated so long as such parties remain in
full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and all
its terms and conditions for copying, distributing or modifying the
Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further restrictions
on the recipients' exercise of the rights granted herein. You are not
responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
    `show w'. This is free software, and you are welcome to redistribute
    it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

    Yoyodyne, Inc., hereby disclaims all copyright interest in the
    program `Gnomovision' (which makes passes at compilers) written by
    James Hacker.

    signature of Ty Coon, 1 April 1989
    Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

---

## CLASSPATH EXCEPTION
Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.1527 thrift 0.9.3

1.1527.1 Available under license:
This package was debianized by Thrift Developer's <dev@thrift.apache.org>.

This package and the Debian packaging is licensed under the Apache License, see `/usr/share/common-licenses/Apache-2.0'.

The following informations was copied from Apache Thrift LICENSE file.

SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

The following files contain some portions of code contributed under
the Thrift Software License (see doc/old-thrift-license.txt), and relicensed under the Apache 2.0 License:

- `compiler/cpp/Makefile.am`
- `compiler/cpp/src/generate/t_cocoa_generator.cc`
- `compiler/cpp/src/generate/t_cpp_generator.cc`
- `compiler/cpp/src/generate/t_csharp_generator.cc`
- `compiler/cpp/src/generate/t_erl_generator.cc`
- `compiler/cpp/src/generate/t_hs_generator.cc`
- `compiler/cpp/src/generate/t_java_generator.cc`
- `compiler/cpp/src/generate/t_ocaml_generator.cc`
- `compiler/cpp/src/generate/t_perl_generator.cc`
- `compiler/cpp/src/generate/t_php_generator.cc`
- `compiler/cpp/src/generate/t_py_generator.cc`
- `compiler/cpp/src/generate/t_rb_generator.cc`
- `compiler/cpp/src/generate/t_st_generator.cc`
- `compiler/cpp/src/generate/t_xsd_generator.cc`
- `compiler/cpp/src/main.cc`
- `compiler/cpp/src/parse/t_field.h`
- `compiler/cpp/src/parse/t_program.h`
- `compiler/cpp/src/platform.h`
- `compiler/cpp/src/thriftl.ll`
- `compiler/cpp/src/thrifty.yy`
- `lib/csharp/src/Protocol/TBinaryProtocol.cs`
- `lib/csharp/src/Protocol/TField.cs`
- `lib/csharp/src/Protocol/TList.cs`
- `lib/csharp/src/Protocol/TMap.cs`
- `lib/csharp/src/Protocol/TMessage.cs`
- `lib/csharp/src/Protocol/TMessageType.cs`
- `lib/csharp/src/Protocol/TProtocol.cs`
- `lib/csharp/src/Protocol/TProtocolException.cs`
- `lib/csharp/src/Protocol/TProtocolFactory.cs`
- `lib/csharp/src/Protocol/TProtocolUtil.cs`
- `lib/csharp/src/Protocol/TSet.cs`
- `lib/csharp/src/Protocol/TStruct.cs`
- `lib/csharp/src/Protocol/TTType.cs`
- `lib/csharp/src/Server/TServer.cs`
- `lib/csharp/src/Server/TSimpleServer.cs`
- `lib/csharp/src/Server/TTThreadPoolServer.cs`
- `lib/csharp/src/TApplicationException.cs`
- `lib/csharp/src/Thrift.csproj`
- `lib/csharp/src/Thrift.sln`
- `lib/csharp/src/TProcessor.cs`
- `lib/csharp/src/Transport/TServerSocket.cs`
- `lib/csharp/src/Transport/TServerTransport.cs`
- `lib/csharp/src/Transport/TSocket.cs`
- `lib/csharp/src/Transport/TStreamTransport.cs`
- `lib/csharp/src/Transport/TTransport.cs`
lib/csharp/src/Transport/TTransportException.cs
lib/csharp/src/Transport/TTransportFactory.cs
lib/csharp/ThriftMSBuildTask/Properties/AssemblyInfo.cs
lib/csharp/ThriftMSBuildTask/ThriftBuild.cs
lib/csharp/ThriftMSBuildTask/ThriftMSBuildTask.csproj
lib/rb/lib/thrift.rb
lib/st/README
lib/st/thrift.st
test/OptionalRequiredTest.cpp
test/OptionalRequiredTest.thrift
test/ThriftTest.thrift

--------------------------------------------------
For the aclocal/ax_boost_base.m4 and contrib/fb303/aclocal/ax_boost_base.m4 components:

#   Copyright (c) 2007 Thomas Porschberg <thomas@randspringer.de>
#
#   Copying and distribution of this file, with or without
#   modification, are permitted in any medium without royalty provided
#   the copyright notice and this notice are preserved.

--------------------------------------------------
For the compiler/cpp/src/md5.[ch] components:

/*
Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch
ghost@aladdin.com
*/
For the lib/rb/setup.rb: Copyright (c) 2000-2005 Minero Aoki,
lib/ocaml/OCamlMakefile and lib/ocaml/README-OCamlMakefile components:
    Copyright (C) 1999 - 2007 Markus Mottl

Licensed under the terms of the GNU Lesser General Public License 2.1
(see doc/lgpl-2.1.txt for the full terms of this license)

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with
separate copyright notices and license terms. Your use of the source
code for the these subcomponents is subject to the terms and
conditions of the following licenses.

Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

For the aclocal/ax_boost_base.m4 and contrib/fb303/aclocal/ax_boost_base.m4 components:

# Copyright (c) 2007 Thomas Porschberg <thomas@randspringer.de>
#
# Copying and distribution of this file, with or without
# modification, are permitted in any medium without royalty provided
# the copyright notice and this notice are preserved.

For the compiler/cpp/src/thrift/md5.[ch] components:

/*
Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved.
This software is provided ‘as-is’, without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch
ghost@aladdin.com

* /

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Tue Oct 24 12:28:44 CDT 2006

Copyright (c) <2006> <Martin J. Logan, Erlware>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software (OTP Base, fslib, G.A.S) and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Thrift
Copyright 2006-2010 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1528 mongodb-core 2.1.20 2.1.20

1.1528.1 Available under license:
Mongodb-core uses third-party libraries or other resources that may be distributed under licenses different than the Mongodb-core software.

In the event that we accidentally failed to list a required notice, please bring it to our attention through any of the ways detailed here:

mongodb-dev@googlegroups.com

The attached notices are provided for information only.

For any licenses that require disclosure of source, sources are available at https://github.com/mongodb/node-mongodb-native.

1) License Notice for nan.h
-------------------------------

The MIT License (MIT)
Copyright (c) 2016 NAN contributors
-----------------------------------

NAN contributors listed at <https://github.com/nodejs/nan#contributors>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (yyyy) [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1529 jackson 2.12.1
1.1529.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0. To find the details that apply to this artifact see the accompanying LICENSE file.
## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1530 fs-readfile-promise 2.0.1 2.0.1
1.1530.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 - 2015 Shinnosuke Watanabe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1531 glob-parent 3.1.0
1.1531.1 Available under license:
The ISC License

Copyright (c) 2015 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1532 @babel/plugin-proposal-class-properties 7.13.0
1.1532.1 Available under license:
The MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1533 default-require-extensions 3.0.0
1.1533.1 Available under license:
MIT License

Copyright (c) Node.js contributors, James Talmage <james@talmage.io> (github.com/jamestalmage)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1534 @babel/plugin-proposal-optional-catch-binding 7.13.8
1.1534.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
1.1535 junit-jupiter-api 5.7.0
1.1535.1 Available under license:
Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

### 1. Definitions

Contribution means:

* **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
* **b)** in the case of each subsequent Contributor:
  * **i)** changes to the Program, and
  * **ii)** additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)** distributing or **b)** making available in any manner that enables the transfer
Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

### 2. Grant of Rights

**a)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

**b)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

**c)** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

**d)** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

**e)** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

### 3. Requirements

**3.1** If a Contributor Distributes the Program in any form, then:

* **a)** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

* **b)** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
  * **i)** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied,
including warranties or conditions of title and non-infringement, and implied warranties or conditions of
merchantability and fitness for a particular purpose;
* **(ii)** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect,
special, incidental and consequential damages, such as lost profits;
* **(iii)** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
* **(iv)** requires any subsequent distribution of the Program by any party to be under a license that satisfies the
requirements of this section 3.

**3.2** When the Program is Distributed as Source Code:

* **(a)** it must be made available under this Agreement, or if the Program **(i)** is combined with other material
in a separate file or files made available under a Secondary License, and **(ii)** the initial Contributor attached to
the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available
under the terms of such Secondary Licenses, and
* **(b)** a copy of this Agreement must be included with each copy of the Program.

**3.3** Contributors may not remove or alter any copyright, trademark, attribution notices, disclaimers of
warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they
Distribute, provided that Contributors may add their own appropriate notices.

### 4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners
and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner which does not create potential
liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering,
such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor
(Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits
and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts
or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial
product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or
alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **(a)** promptly
notify the Commercial Contributor in writing of such claim, and **(b)** allow the Commercial Contributor to
tool, and cooperate with the Commercial Contributor to, the defense and any related settlement negotiations. The
Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That
Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or
offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's
responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other
Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay
any damages as a result, the Commercial Contributor must pay those damages.

### 5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY
APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION,
ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the
appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights
under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable
laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

### 6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY
APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or
enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such
provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit)
alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes
such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such
litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or
conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and
distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this
Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the
Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the
right to publish new versions (including revisions) of this Agreement from time to time. No one other than the
Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement
Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable
separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program
(including Contributions) may always be Distributed subject to the version of the Agreement under which it was
received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the
Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual
property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All
rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is
intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are
created under this Agreement.

#### Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: [name license(s), version(s), and exceptions or additional permissions here].

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.1536 babel-helper-call-delegate 6.24.1

1.1536.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The "Software identified by reference to the Ecma Standard* ("Software")" is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1537 should.js 13.2.3 13.2.3

1.1537.1 Available under license :

Copyright(c) 2010-2013 TJ Holowaychuk <tj@vision-media.ca>
Copyright(c) 2013-2017 Denis Bardadym <bardadymchik@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1538 readdirp 3.2.0
1.1538.1 Available under license:
MIT License

Copyright (c) 2012-2019 Thorsten Lorenz, Paul Miller (https://paulmillr.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1539 lodash-some 4.6.0
1.1539.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1540 password-hash 1.2.2

1.1540.1 Available under license:

Copyright (c) 2011 David Wood <bitprobe@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1541 puppeteer 1.11.0

1.1541.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2017 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1542 hpkp 1.2.0 1.2.0
1.1542.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1543 crypto-random-string 2.0.0

1.1543.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1544 jetty-xml 9.4.17.v20190418

1.1544.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:
a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or
otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in
any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==============================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
   http://www.apache.org/licenses/LICENSE-2.0.html

   and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.  
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.  
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from
following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html
/org.eclipse.jetty.toolchain:jetty-schemas

------

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1545 execa 1.0.0

1.1545.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1546 once 1.4.0

1.1546.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1547 babel-plugin-transform-es2015-parameters 6.24.1 6.24.1

1.1547.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
1.1548 websocket-servlet 9.4.17.v20190418

1.1548.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.
"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warrants or
conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose; ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If
that Commercial Contributor then makes performance claims, or offers
warranties related to Product X, those performance claims and warranties are
such Commercial Contributor's responsibility alone. Under this section, the
Commercial Contributor would have to defend claims against the other
Contributors related to those performance claims and warranties, and if a
court requires any other Contributor to pay any damages as a result, the
Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR
IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE,
NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each
Recipient is solely responsible for determining the appropriateness of using
and distributing the Program and assumes all risks associated with its
exercise of rights under this Agreement, including but not limited to the
risks and costs of program errors, compliance with applicable laws, damage to
or loss of data, programs or equipment, and unavailability or interruption of
operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY
CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION
LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of the
remainder of the terms of this Agreement, and without further action by the
parties hereto, such provision shall be reformed to the minimum extent
necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Program itself
(excluding combinations of the Program with other software or hardware)
infringes such Recipient’s patent(s), then such Recipient’s rights granted
under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient’s rights under this Agreement shall terminate if it fails to
comply with any of the material terms or conditions of this Agreement and does
not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed subsequently incorporated within the Work.
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl
These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------

Mortbay
The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1549 servlet-api 3.1.0
1.1549.1 Available under license:
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the
3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION.
6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item., as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.
9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction’s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys’ fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.
When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to
achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program’s name and a brief idea of what it does.

Copyright (C) 

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2
Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file’s header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.1550 flagged-respawn 1.0.1

1.1550.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2018 Tyler Kellen <tyler@sleekcode.net>, Blaine Bublitz <blaine.bublitz@gmail.com>, and Eric Schoffstall <yo@contra.io>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1551 summernote 0.8.15 0.8.15

1.1551.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1552 is-extglob 1.0.0

1.1552.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1553 setimmediate 1.0.5 1.0.5
1.1553.1 Available under license:
Copyright (c) 2012 Barnesandnoble.com, llc, Donavon West, and Domenic Denicola

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1554 decamelize 1.2.0 1.2.0
1.1554.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1555 core-js 2.5.7 2.5.7

1.1555.1 Available under license:
Copyright (c) 2014-2018 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1556 commons-io 2.5

1.1556.1 Available under license:
Apache Commons IO
Copyright 2002-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1557 batik-bridge 1.13

1.1557.1 Available under license:

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/META-INF/maven/org.apache.xmlgraphics/batik-bridge/pom.xml

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#   http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/resources/Messages.properties

No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

*/
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/Messages.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/UserAgentViewport.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGEfeOffsetElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGBrokenLinkProvider.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/ConcreteTextPainter.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/UserAgent.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGCircleElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/RelaxedScriptSecurity.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/GVTBuilder.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/GlyphLayout.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGDокументBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/UpdateManager.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/svg12/SVG12BridgeExtension.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/CSSFontFace.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/GraphicsNodeBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/DefaultScriptSecurity.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGEllipseElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/DocumentBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
Open Source Used In DNAC Platform Antelope 5464
jar/org/apache/batik/bridge/BridgeUpdateHandler.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/UpdateManagerListener.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/svg12/SVG12FocusManager.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/ScriptSecurity.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/AnimationSupport.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/AnimatableGenericSVGBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/FontFace.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/FlowTextPainter.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/SVGImageElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/SVGPolygonElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/BasicTextPainter.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/BridgeEventSupport.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/WindowWrapper.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/svg12/XBLContentElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/StrokingTextPainter.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/SVGFeDisplacementMapElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/BridgeContext.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/StyleReference.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/TextLayoutFactory.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/NoLoadScriptSecurity.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-jar/org/apache/batik/bridge/DocumentJarClassLoader.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGFontFamily.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/PaintServer.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGFeGaussianBlurElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGDescElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGFeGVTFont.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/RelaxedExternalResourceSecurity.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/MarkerBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/DynamicGVTBuilder.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/svg12/SVGMultiImageElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/RhinoInterpreter.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGFontFace.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/svg12/DefaultContentSelector.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/FilterPrimitiveBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/InterruptedBridgeException.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/SVGAltGlyphElementBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/ConcreteTextLayoutFactory.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/FilterPrimitiveBridge.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/ConcreteTextLayoutFactory.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/ConcreteTextLayoutFactory.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/ConcreteTextLayoutFactory.java
* /opt/cola/permits/1146878544_1616835051.89/0/batik-bridge-1-13-sources-1-
jar/org/apache/batik/bridge/ConcreteTextLayoutFactory.java
1.1558 immer 8.0.1 8.0.1
1.1558.1 Available under license:
MIT License

Copyright (c) 2017 Michel Weststrate

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1559 parquet-hadoop 1.12.0
1.1559.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This product includes code from Twitter's ElephantBird project.

* parquet-hadoop's UnmaterializableRecordCounter.java includes code from ElephantBird's LzoRecordReader.java

Copyright: 2012-2014 Twitter
Home page: https://github.com/twitter/elephant-bird
License: http://www.apache.org/licenses/LICENSE-2.0

1.1560 fromentries 1.3.2
1.1560.1 Available under license:
The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1561 guava 30.0-jre

1.1561.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/*
 * Holder for web specializations of methods of {code Floats}. Intended to be empty for regular
 * version.
 */

Found in path(s):
*/opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/FloatsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2014 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
*/opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/parallel/TrustedListenableFutureTask.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/math/Quantiles.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/MoreObjects.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/ListenerCallQueue.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/math/MathPreconditions.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/Hasher.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/math/DoubleUtils.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/AbstractStreamingHasher.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
Open Source Used In DNAC Platform Antelope 5479

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/ForwardingLoadingCache.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/base/Optional.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/base/Enums.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/net/MediaType.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/ForwardingCache.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/primitives/UnsignedLong.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/hash/Murmur3_128HashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/RemovalCause.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/Cache.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/hash/Funnels.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/hash/HashCode.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/hash/HashingOutputStream.java
jar/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/Absent.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/Queues.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/hash/Hashing.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

Open Source Used In DNAC Platform Antelope 5480
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/UncaughtExceptionHandler.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/Atomics.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/annotations/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the
 * License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * either
 * express or implied. See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):

/*
 * Copyright (C) 2017 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
Open Source Used In DNAC Platform Antelope 5483

/*

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/primitives/ImmutableLongArray.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingCondition.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingLock.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2020 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
* */
/**
* Holder for web specializations of methods of { @code Doubles }. Intended to be empty for regular
* version.
* */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/primitives/DoublesMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2010 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/MinMaxPriorityQueue.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ForwardingImmutableCollection.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/AbstractSequentialIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ForwardingListMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/UnmodifiableListIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ForwardingSortedSetMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/RowSortedTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ForwardingSetMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/SortedMapDifference.java
No license file was found, but licenses were detected in source scan.

/
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/AbstractBiMap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/AbstractListMultimap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/SingletonImmutableSet.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ForwardingMultimap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/LinkedHashMultimap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/NaturalOrdering.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ForwardingSortedSet.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/AbstractMapEntry.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/HashMultimap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ForwardingListIterator.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/LinkedListMultimap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/MutableClassToInstanceMap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/SetMultimap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Multisets.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ForwardingMultiset.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Multimap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ReverseNaturalOrdering.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Multimaps.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ForwardingObject.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/TreeMultiset.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ArrayListMultimap.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Ordering.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Iterators.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ForwardingQueue.java

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/MapDifference.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/TreeMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/AbstractMapBasedMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/NullsFirstOrdering.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/LexicographicalOrdering.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ConcurrentHashMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/BitMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/NullsLastOrdering.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Maps.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/AbstractSetMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ByFunctionOrdering.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ReverseOrdering.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/AbstractMapBasedMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ListMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Lists.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/RegularImmutableSet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Iterables.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/ForwardingMapEntry.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Sets.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/Interners.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at

Open Source Used In DNAC Platform Antelope 5487
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/math/ToDoubleRounder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/hash/Java8Compatibility.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/math/BigDecimalMath.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
 * may not use this file except in compliance with the License. You may
 * obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 * implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collect/Streams.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/AggregateFutureState.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/FarmHashFingerprint64.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/Platform.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/LittleEndianByteArray.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This following method is a modified version of one found in
 * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
 * which contained the following notice:
 *
 * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
 * the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
 *
 * Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing permissions and limitations under
 */
* the License.
*/
/**
* Holder for web specializations of methods of [code Shorts]. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

*/
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of [code Objects} only in web. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

*/
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/base/Verify.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/io/CharSequenceReader.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/util/concurrent/Runnables.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/base/Utf8.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
Open Source Used In DNAC Platform Antelope 5493

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/AbstractValueGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/DirectedGraphConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/UndirectedGraphConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/DirectedNetworkConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/StandardMutableValueGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/Collectors.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ValueGraphBuilder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/MoreCollectors.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/LinkedHashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/GraphBuilder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ImmutableValueGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/CollectCollectors.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/EndpointPair.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/GraphConstants.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/StandardMutableGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/StandardMutableNetwork.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/CollectCollectors.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ForwardingValueGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ElementOrder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/MapIteratorCache.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 */
* http://www.apache.org/licenses/LICENSE-2.0
 *
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/ImmutableSupplier.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/ExecutionSequencer.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableSet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/Hashing.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

/ *
* This method was rewritten in Java from an intermediate step of the Murmur hash function in
* http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
* following header:
*
* MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
* hereby disclaims copyright to this source code.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/Hashing.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/RegularImmutableList.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ComparisonChain.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/EmptyImmutableSetMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ComputationException.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ImmutableClassToInstanceMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ArrayTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ImmutableTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/SingletonImmutableList.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/RegularImmutableSortedSet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/AbstractIndexedListIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/DiscreteDomain.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ImmutableSortedSetFauxxverideShim.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ImmutableSortedMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ImmutableSetMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/ForwardingTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/ImmutableAsList.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/SingletonImmutableTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/ImmutableEnumSet.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy of
 * the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/SortedMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java
* /opt/ cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/CollectSpliterators.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
*/
* This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
* and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
* *
* As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
* java.util.HashMap class.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/Striped.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/SuccessorsFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/RecursiveDeleteOption.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/InsecureRecursiveDeleteException.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/Network.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ImmutableNetwork.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/Graph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/graph/Graph.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

---

Open Source Used In DNAC Platform Antelope 5499
Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/ServiceManagerBridge.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/Internal.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/primitives/Platform.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/JdkBackedImmutableBiMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/IndexedImmutableSet.java

Open Source Used In DNAC Platform Antelope 5500
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMap.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/AtomicLongMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/GwtTransient.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/JdkPattern.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/GwtTransient.java
jar/com/google/common/base/PatternCompiler.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/CommonPattern.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/CommonMatcher.java

No license file was found, but licenses were detected in source scan.

/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/AtomicDoubleArray.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/Striped64.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/hash/LongAdder.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/MultimapBuilder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/ImmutableMapEntry.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/ImmutableMapEntry.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/Platform.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/UnsignedBytes.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/reflect/TypeResolver.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/html/HtmlEscapers.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/Cut.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/SettableFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/DenseImmutableTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/SignedBytes.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/io/ByteArrayDataInput.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/graph/IncidentEdgeSet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/CompactHashing.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
Open Source Used In DNAC Platform Antelope 5506

* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/io/BaseEncoding.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/io/Closer.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/sha/SipHashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/sha/SipHashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/sha/ChecksumHashFunction.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/html/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/html/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/html/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/html/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1.jar/com/google/common/html/package-info.java
/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/Predicate.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/ByteStreams.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/Function.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/MultiInputStream.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/eventbus/Subscribe.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/LineBuffer.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/Objects.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/FinalizableReference.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/LineReader.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/Closeables.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/eventbus/AllowConcurrentEvents.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/FinalizableReferenceQueue.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/eventbus/DeadEvent.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/FinalizablePhantomReference.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/package-info.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/Files.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collect/Interners.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/eventbus/package-info.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 The Guava Authors
*/
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/CollectPreconditions.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/Table.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/Range.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/UnmodifiableIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/Platform.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ImmutableListMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ImmutableMapEntrySet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/Collections2.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/Serialization.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/PeekingIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/HashBasedTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/StandardTable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ImmutableMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/SingletonImmutableBiMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ImmutableBiMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
  jar/com/google/common/collection/ImmutableBiMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
/**
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 * /
 **/

* Holder for web specializations of methods of { @code Ints }. Intended to be empty for regular
* version.
*
No license file was found, but licenses were detected in source scan.
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/CompactLinkedHashMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/FilteredKeySetMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/DescendingMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/FilteredMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/RangeMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ForwardingDeque.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/TreeRangeMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ForwardingNavigableSet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ForwardingNavigableMap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/TransformedIterator.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/FilteredEntrySetMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/SortedMultisetBridge.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/AbstractSortedKeySortedSetMultimap.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
No license file was found, but licenses were detected in source scan.

/* Copyright (C) 2006 The Guava Authors */
/* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except */
/* in compliance with the License. You may obtain a copy of the License at */
/* */
/* http://www.apache.org/licenses/LICENSE-2.0 */
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/annotations/VisibleForTesting.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * Outer class that exists solely to let us write { @code Partially.GwtIncompatible } instead of plain
 * { @code GwtIncompatible }. This is more accurate for { @link Futures#catching }, which is available
 * under GWT but with a slightly different signature.
 *
 * &lt; We can't use { @code PartiallyGwtIncompatible } because then the GWT compiler wouldn't recognize
 * it as a { @code GwtIncompatible } annotation. And for { @code Futures.catching }, we need the GWT
 * compiler to autostrip the normal server method in order to expose the special, inherited GWT
 * version.
 */

Found in path(s):

/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 */
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collection/FluentIterable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/Floats.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/thirdparty/publicsuffix/TrieParser.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/io/MultiReader.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/util/concurrent/ListenableFutureTask.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/Doubles.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/Chars.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/internal/Finalizer.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/Matcher.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/Bytes.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
/*
* Copyright (C) 2007 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

/**
* Returns an array containing all of the elements in the specified collection. This method
* returns the elements in the order they are returned by the collection's iterator. The returned
* array is "safe" in that no references to it are maintained by the collection. The caller is
* thus free to modify the returned array.
*<p>This method assumes that the collection size doesn't change while the method is running.
*<p>TODOR(kevinb): support concurrently modified collections?
*<p>@param c the collection for which to return an array of elements
*/

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}</b>
 * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
 * version.
 * @throws UnsupportedOperationException always
 * @deprecated <b>Pass a key of type {@code Comparable} to use {@link
 * ImmutableSortedMap#of(Comparable, Object)}. </b>
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-
jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.
/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the
 * License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
 * express or implied. See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ImmutableSortedMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/SortedIterables.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/ForwardingSortedMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/RangeSet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/GeneralRange.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/AbstractRangeSet.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/SortedIterable.java
* /opt/cola/permits/1104665043_1604644117.38/0/guava-30-0-jre-sources-1-jar/com/google/common/collection/Count.java

1.1562 sprintf-js 1.0.3
1.1562.1 Available under license :
MIT License

Copyright (c) 2019 Johannes Tegnr / Jitesoft

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1563 flat 4.1.1
1.1563.1 Available under license:
Copyright (c) 2014, Hugh Kennedy
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the nor the names of its contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.1564 ret 0.2.2 0.2.2

1.1564.1 Available under license:
MIT License

Copyright (C) 2011 by fent

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1565 performance-now 2.1.0 2.1.0

1.1565.1 Available under license:
Copyright (c) 2013 Braveg1rl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1566 memoize-one 4.1.0 4.1.0

1.1566.1 Available under license:

MIT License

Copyright (c) 2017 Alexander Reardon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1567 librsvg 2.40.13-3

1.1567.1 Available under license:

GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.
This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor’s protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don’t assume that anything in it is
the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.
The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a
table of data to be supplied by an application program that uses
the facility, other than as an argument passed when the facility
is invoked, then you must make a good faith effort to ensure that,
in the event an application does not supply such function or
table, the facility still operates, and performs whatever part of
its purpose remains meaningful.

(For example, a function in a library to compute square roots has
a purpose that is entirely well-defined independent of the
application. Therefore, Subsection 2d requires that any
application-supplied function or table used by this function must
be optional: if the application does not supply it, the square
root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.
This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

d) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the source code distributed need not include anything that is normally
distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors’ reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include....
anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.
If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.
NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

  Gnomovision version 69, Copyright (C) year name of author
  Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
  This is free software, and you are welcome to redistribute it
  under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

1.1568 lowercase-keys 2.0.0
1.1568.1 Available under license:
    MIT License

    Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

    Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1569 babel-polyfill 6.26.0

1.1570 accepts 1.3.7

1.1570.1 Available under license :
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1571 options 0.0.6 0.0.6

1.1571.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1572 component-emitter 1.1.2

1.1572.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1573 cliui 6.0.0 6.0.0

1.1573.1 Available under license:

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1574 babel-helper-call-delegate 6.24.1 6.24.1

1.1574.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1575 unbox-primitive 1.0.1
1.1575.1 Available under license:
MIT License

Copyright (c) 2019 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1576 latencyutils 2.0.3

1.1576.1 Available under license:

The person who associated a work with this deed has dedicated the work to the public domain by waiving all of his or her rights to the work worldwide under copyright law, including all related and neighboring rights, to the extent allowed by law.

You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission. See Other Information below.

This license is acceptable for Free Cultural Works.

Other Information
In no way are the patent or trademark rights of any person affected by CC0, nor are the rights that other persons may have in the work or in how the work is used, such as publicity or privacy rights.

Unless expressly stated otherwise, the person who associated a work with this deed makes no warranties about the work, and disclaims liability for all uses of the work, to the fullest extent permitted by applicable law.

When using or citing the work, you should not imply endorsement by the author or the affirmer.

Public Domain, per Creative Commons CC0
http://creativecommons.org/publicdomain/zero/1.0/
1.1577 hasha 5.2.2

1.1577.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1578 de.huxhorn.lilith.data.logging.protobuf

0.9.44

1.1578.1 Available under license:
GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the
Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this
license document.


You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this
license document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of
this License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer
system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
Version.
e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.

1.1579 object-inspect 1.9.0
1.1579.1 Available under license:
   MIT License

Copyright (c) 2013 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1580 ws 6.1.4
1.1580.1 Available under license:
The MIT License (MIT)

Copyright (c) 2011 Einar Otto Stangvik <einaros@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1581 v8flags 3.2.0
1.1581.1 Available under license :
   The MIT License (MIT)

   Copyright (c) 2014-2018 Tyler Kellen <tyler@sleekcode.net>, Blaine Bublitz <blaine.bublitz@gmail.com>, and
   Eric Schoffstall <yo@contra.io>

   Permission is hereby granted, free of charge, to any person obtaining a copy
   of this software and associated documentation files (the "Software"), to deal
   in the Software without restriction, including without limitation the rights
   to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
   copies of the Software, and to permit persons to whom the Software is
   furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all
   copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
   IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
   FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
   AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
   LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
   OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
   SOFTWARE.

1.1582 ansi-regex 5.0.0 5.0.0
1.1582.1 Available under license :
   MIT License

   Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
   documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
   rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
   persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1583 babel-plugin-syntax-trailing-function-commas 6.22.0 6.22.0

1.1583.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1584 groovy-swing 3.0.8

1.1584.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 *Licensed to the Apache Software Foundation (ASF) under one
 *or more contributor license agreements. See the NOTICE file
 *distributed with this work for additional information
 *regarding copyright ownership. The ASF licenses this file
 *to you under the Apache License, Version 2.0 (the
 *"License"); you may not use this file except in compliance
 *with the License. You may obtain a copy of the License at
 */
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/PropertyBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/model/NestedValueModel.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/TableModelFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/model/PropertyModel.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/ClosureSourceBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/FullBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/BindPath.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/PropertyBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/JTableMetaMethods.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/model/FormModel.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/AbstractFullBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/JComponentProperties.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/AbstractButtonProperties.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/TriggerBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/DialogFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/BindFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/runtime/SwingGroovyMethods.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/WidgetFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-
Open Source Used In DNAC Platform Antelope 5553

- jar/groovy/swing/model/ClosureModel.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/GridBagFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/RichActionWidgetFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/TextArgWidgetFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/WindowFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/JComboBoxProperties.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/ComponentFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/AbstractFullBinding.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/SwingBorderFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/impl/TableLayout.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/ComboBoxFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/ListFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/ClosureSourceBinding.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/inspect/swingui/TableSorter.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/ButtonFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/TitledBorderFactory.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/inspect/swingui/TableMap.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/SwingTimerTriggerBinding.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/TargetBinding.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/FormattedTextFactory.groovy
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/DefaultTableColumn.java
- /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/TabbedPaneFactory.groovy
jar/groovy/swing/factory/EtchedBorderFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/EmptyBorderFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/iml/TableLayoutCell.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/MapFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/BoxLayoutFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/SwingBuilder.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/org/codehaus/groovy/binding/ClosureTriggerBinding.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/binding/JTextComponentProperties.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/apache/groovy/swing/binding/PropertyPathFullBinding.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/table/TableMap.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/iml/ListWrapperListModel.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/BindProxyFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/apache/groovy/swing/binding/BindingProxy.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/apache/groovy/swing/binding/FullBinding.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/binding/EventTriggerBinding.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/apache/groovy/swing/binding/BindingUpdatable.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/ActionFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/SeparatorFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/apache/groovy/swing/binding/PropertyPathFullBinding.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/CompoundBorderFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/iml/ClosureRenderer.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/RootPaneContainerFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/RootPaneContainerFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/apache/groovy/swing/binding/AggregateBinding.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/RootPaneContainerFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/apache/groovy/swing/binding/AggregateBinding.java
* /opt/cola/permits/1218299960_1634924418.62/groovy-swing-3.0-8-sources.jar/groovy/swing/factory/RootPaneContainerFactory.groovy
Open Source Used In DNAC Platform Antelope 5556

* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/AggregateBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/BindingProxy.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/model/DefaultTableModel.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/JTableProperties.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/PropertyChangeProxyTargetBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/model/ValueHolder.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/JComboBoxMetaMethods.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/TitledBorderFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/JSpinnerProperties.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/ClosureTriggerBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/BindPath.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/BevelBorderFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/BindGroupFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/impl/TableLayoutRow.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/ImageIconFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/impl/DefaultAction.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/MutualPropertyBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/TriggerBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/impl/TableLayoutRow.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/impl/TableLayoutRow.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/impl/TableLayoutRow.java
jar/org/apache/groovy/swing/binding/EventTriggerBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/CellEditorFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/impl/ClosureCellEditor.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/MutualPropertyBinding.java
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/CollectionFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/factory/ColumnFactory.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/LookAndFeelHelper.groovy
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/AbstractSyntheticMetaMethods.groovy

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.

Found in path(s):
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/apache/groovy/swing/binding/package.html
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/impl/package.html
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/model/package.html
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/org/codehaus/groovy/binding/package.html
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/package.html
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/package.html
* /opt/cola/permits/1218299960_1634924418.62/0/groovy-swing-3-0-8-sources-jar/groovy/swing/binding/package.html

Open Source Used In DNAC Platform Antelope 5557
1.1585 is-my-ip-valid 1.0.0 1.0.0

1.1585.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1586 safe-buffer 5.2.1 5.2.1

1.1586.1 Available under license:

The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1587 hsts 1.0.0 1.0.0

1.1587.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1588 stealthy-require 1.1.1

1.1588.1 Available under license:
ISC License

Copyright (c) 2017, Nicolai Kamenzky and contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
1.1589 uglify-js 3.11.3

1.1589.1 Available under license:

UglifyJS is released under the BSD license:

Copyright 2012-2019 (c) Mihai Bazon <mihai.bazon@gmail.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1590 parquet-common 1.8.1

1.1590.1 Available under license:

Apache Parquet Common
Copyright 2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
TERM AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1591 1to2 1.0.0

1.1591.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*********************************************************************
* NAN - Native Abstractions for Node.js
*
* Copyright (c) 2016 NAN contributors
*
* MIT License <https://github.com/nodejs/nan/blob/master/LICENSE.md>
*********************************************************************/

Found in path(s):
* /opt/cola/permits/1127086437_1611781577.89/0/1to2-1-0-0-2-tgz/package/1to2.js

1.1592 crc32-stream 3.0.1

1.1592.1 Available under license:
Copyright (c) 2014 Chris Talkington, contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT

Open Source Used In DNAC Platform Antelope 5565
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1593 har-validator 2.1.3 2.1.3
1.1593.1 Available under license:
Copyright (c) 2015, Ahmad Nassri <ahmad@ahmadnassri.com>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1594 @babel/parser 7.11.5
1.1594.1 Available under license:
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1595.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1596 swagger-jaxrs 1.5.0

1.1597 lodash-find 4.6.0

1.1597.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1598 through 2.3.8 2.3.8
1.1598.1 Available under license:
The MIT License

Copyright (c) 2011 Dominic Tarr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom
the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice
shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache License, Version 2.0

Copyright (c) 2011 Dominic Tarr

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1599 async 1.5.2
1.1599.1 Available under license :

The MIT License (MIT)

Copyright (c) Spatie bvba <info@spatie.be>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1600 classmate 0.8.0

1.1600.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1601 cookie 0.1.5 0.1.5

1.1601.1 Available under license :
(The MIT License)

Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
1.1602 callsites 0.2.0

1.1602.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1603 atk 2.18.0 1

1.1603.1 Available under license:

GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA  02111-1307  USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain
designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs

---

Open Source Used In DNAC Platform Antelope 5577
(which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a
table of data to be supplied by an application program that uses
the facility, other than as an argument passed when the facility
is invoked, then you must make a good faith effort to ensure that,
in the event an application does not supply such function or
table, the facility still operates, and performs whatever part of
its purpose remains meaningful.

(For example, a function in a library to compute square roots has
a purpose that is entirely well-defined independent of the
application. Therefore, Subsection 2d requires that any
application-supplied function or table used by this function must
be optional: if the application does not supply it, the square
root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.
6. As an exception to the Sections above, you may also compile or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

d) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the source code distributed need not include anything that is normally
distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.
It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safer to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
This package was debianized by Akira TAGOH <tagoh@debian.org> on Sun, 10 Mar 2002 20:41:39 +0900.

It was downloaded from <http://ftp.acc.umu.se/pub/GNOME/sources/atk/>.

Upstream Authors:
Bill.Haneman <bill.haneman@sun.com>
Marc.Mulcahy <marc.mulcahy@sun.com>
Padraig.Obriain <padraig.obriain@sun.com>

Copyright:

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

See /usr/share/common-licenses/LGPL-2 on your debian system.

1.1604 grpc-api 1.27.1
1.1604.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute
must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1605 asm 2.2.0-b21

1.1605.1 Available under license:
No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2007 INRIA, France Telecom
 * All rights reserved.
 */
* Redistribution and use in source and binary forms, with or without
  * modification, are permitted provided that the following conditions
  * are met:
  * 1. Redistributions of source code must retain the above copyright
  *    notice, this list of conditions and the following disclaimer.
  * 2. Redistributions in binary form must reproduce the above copyright
  *    notice, this list of conditions and the following disclaimer in the
  *    documentation and/or other materials provided with the distribution.
  * 3. Neither the name of the copyright holders nor the names of its
  *    contributors may be used to endorse or promote products derived from
  *    this software without specific prior written permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
  * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
  * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
  * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
  * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
  * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
  * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
  * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
  * THE POSSIBILITY OF SUCH DAMAGE.
  */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/util/Traceable.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/util/ASMifiable.java

No license file was found, but licenses were detected in source scan.

2011 Oracle and/or its affiliates. All rights reserved.

The contents of this file are subject to the terms of either the GNU
General Public License Version 2 only ("GPL") or the Common Development
and Distribution License("CDDL") (collectively, the "License"). You
may not use this file except in compliance with the License. You can
obtain a copy of the License at
https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
or packager/legal/LICENSE.txt. See the License for the specific
language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each
file and include the License file at packager/legal/LICENSE.txt.

GPL Classpath Exception:
Oracle designates this particular file as subject to the "Classpath"
 exception as provided by Oracle in the GPL Version 2 section of the License file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or only the GPL Version 2, indicate your decision by adding "[Contributor] elects to include this software in this distribution under the [CDDL or GPL Version 2] license." If you don't indicate a single choice of license, a recipient has the option to distribute your version of this file under either the CDDL, the GPL Version 2 or to extend the choice of license to its licensees as provided above. However, if you add GPL Version 2 code and therefore, elected the GPL Version 2 license, then the option applies only if the new code is made subject to such option by the copyright holder.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-jar/pom.xml
No license file was found, but licenses were detected in source scan.

* ASM XML Adapter
* Copyright (c) 2004, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/package.html
No license file was found, but licenses were detected in source scan.

ASM XML Adapter
Copyright (c) 2004, Eugene Kuleshov
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/asm-xml.dtd
No license file was found, but licenses were detected in source scan.

/***
* ASM XML Adapter
* Copyright (c) 2004, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/SAXFieldAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/ASMContentHandler.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/SAXClassAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/SAXAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/SAXCodeAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/SAXAnnotationAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/xml/Processor.java

No license file was found, but licenses were detected in source scan.

/** *
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2005 INRIA, France Telecom
* All rights reserved.
* *
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/commons/InstructionAdapter.java

No license file was found, but licenses were detected in source scan.

2005 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF  
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-  
  jar/org/objectweb/asm/tree/analysis/package.html  
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-  
  jar/org/objectweb/asm/attrs/package.html  
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-  
  jar/org/objectweb/asm/package.html  
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-  
  jar/org/objectweb/asm/util/package.html  
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-  
  jar/org/objectweb/asm/signature/package.html  
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-  
  jar/org/objectweb/asm/tree/package.html  
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-  
  jar/org/objectweb/asm/commons/package.html

No license file was found, but licenses were detected in source scan.

/** *
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2007 INRIA, France Telecom
* All rights reserved.
* *
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
* *
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

Open Source Used In DNAC Platform Antelope 5595
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/util/TraceFieldVisitor.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/tree/TableSwitchInsnNode.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/util/ASMifierFieldVisitor.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/util/CheckSignatureAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/FieldVisitor.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/AnnotationVisitor.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/ClassReader.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/commons/RemappingSignatureAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/util/TraceAbstractVisitor.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/Edge.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/tree/analysis/SourceValue.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/tree/analysis/SimpleVerifier.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/Handler.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/tree/analysis/BasicInterpreter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/tree/VarInsnNode.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/tree/IincInsnNode.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/commons/RemappingMethodAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/util/CheckAnnotationAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1-
  jar/org/objectweb/asm/tree/MultiANewArrayInsnNode.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/tree/analysis/AnalyzerException.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/AnnotationWriter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/commons/RemappingAnnotationAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/tree/InsnNode.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/Item.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/tree/MethodInsnNode.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/tree/analysis/BasicVerifier.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/commons/JSRInlinerAdapter.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/MethodVisitor.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/tree/TypeInsnNode.java
* /opt/ws_local/PERMITS_SQL/1048949653_1591433284.08/0/asm-all-repackaged-2-2-0-b21-sources-1.jar/org/objectweb/asm/util/CheckFieldAdapter.java
Open Source Used In DNAC Platform Antelope
1.1606 minlog 1.3.0

1.1606.1 Available under license:
Copyright (c) 2008, Nathan Sweet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Esoteric Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1607 uuid 3.3.2

1.1607.1 Available under license:
The MIT License (MIT)

Copyright (c) 2010-2016 Robert Kieffer and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1608 readable-stream 2.3.7 2.3.7

1.1608.1 Available under license:

Node.js is licensed for use as follows:

```---
Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
---
```

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

```---
Copyright Joyent, Inc. and other Node contributors. All rights reserved.
---
```
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1609 avro 1.9.1
1.1609.1 Available under license :
Apache Avro
Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1610 set-value 2.0.1

1.1610.1 Available under license:

   The MIT License (MIT)

   Copyright (c) 2014-2017, Jon Schlinkert

   Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1611 clone-response 1.0.2

1.1611.1 Available under license:

MIT License

Copyright (c) 2017 Luke Childs

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1612 libvpx 1.7.0-1367-gac3eccdc2
1.1612.1 Available under license:

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2005-2012 x264 project

Authors: Loren Merritt <lorenm@u.washington.edu>
Anton Mitrofanov <BugMaster@narod.ru>
Jason Garrett-Glaser <darkshikari@gmail.com>
Henrik Gramner <hengar-6@student.ltu.se>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2011 The LibYuv Project Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# This file contains a list of people who've made non-trivial
# contribution to the Google C++ Testing Framework project. People
# who commit code to the project are encouraged to add their names
# here. Please keep the list sorted by first names.

Ajay Joshi <jaj@google.com>
Balzs Dn <balazs.dan@gmail.com>
Bharat Mediratta <bharat@menalto.com>
1.1613 path-to-regexp 1.8.0 1.8.0

1.1613.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1614 concat-stream 1.6.2

1.1614.1 Available under license:

The MIT License

Copyright (c) 2013 Max Ogden

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1615 regjsgen 0.2.0 0.2.0

1.1615.1 Available under license:

Copyright 2014 Benjamin Tan <demoneaux@gmail.com> (https://d10.github.io/)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1616 is-my-json-valid 2.17.2 2.17.2
1.1616.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1617 css-select 1.2.0
1.1617.1 Available under license :
Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2007-2013 Diego Perini (http://www.iport.it)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1618 hbase 2.4.2

1.1618.1 Available under license:

--

This product includes portions of the Bootstrap project v3.0.0
Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

This project bundles a copy of the Jquery minified javascript library version 1.8.3 under the terms of the MIT license.

Copyright 2012 Jquery Foundation and other contributors

http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
This project bundles portions of the 'JQuery' project under the terms of the MIT license.

Copyright 2012 jQuery Foundation and other contributors
http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

----

This project bundles a derivative of portions of the 'Asciidoctor' project
under the terms of the MIT license.

The MIT License
Copyright (C) 2012-2015 Dan Allen, Ryan Waldron and the Asciidoctor Project

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
This project incorporates portions of the 'Protocol Buffers' project available under a '3-clause BSD' license.

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE
License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that
offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic
2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."
;c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.
4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only
use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full
compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this
License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.

Apache HBase

Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--

This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0

--

Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14 and v21, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/util/Bytes.java'
'hbase-common/src/main/java/org/apache/hadoop/hbase/util/ByteBuf.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

--
This product includes portions of Jetty project, specially
'hbase-shaded-hbase-shaded-testing-util/src/main/resources/org/apache/hadoop/hbase/shaded/org/mortbay/jetty/webapp/webdefault.xml'

In Ruby’s source distribution, this would describe a number of C source files
that have different licenses than Ruby itself. None of those apply to JRuby,
so we have this file here as a placeholder.

For details of licensing of this collective work, see LICENSE

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

${project.name}
Copyright ${project.inceptionYear}-${build.year} The Apache Software Foundation

This product includes software developed at
1.1619 @babel/helper-validator-identifier
7.10.4
1.1619.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1620 jetty-util-ajax 9.4.34.v20201102
1.1620.1 Available under license:
This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION
Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement , including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.
This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object...
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

====================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-------
Eclipse

The following artifacts are EPL.
  * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
  * org.eclipse.jetty.orbit:javax.security.auth.message
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception. 
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following
artifacts may be included in the distribution or downloaded when ALPN
module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications
are hosted at github and both modified and original are under GPL v2 with
classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the
terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl
The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.
1.1622 multer 1.4.1

1.1622.1 Available under license:

Copyright (c) 2014 Hage Yaapa <http://www.hacksparrow.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1623 ansi-colors 3.2.3

1.1623.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-present, Brian Woodward.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1624 after 0.8.2 0.8.2
1.1624.1 Available under license:

Copyright (c) 2011 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1625 jackson-mapper-asl 1.9.13
1.1625.1 Available under license:

No license file was found, but licenses were detected in source scan.

/* Jackson JSON-processor.
 *
 * Copyright (c) 2007- Tatu Saloranta, tatu.saloranta@iki.fi
 *
 * Licensed under the License specified in file LICENSE, included with
 * the source code and binary code bundles.
 * You may not use this file except in compliance with the License.
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
1.626 aws-sign2 0.7.0

1.626.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner.
owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1627 arraybuffer.slice 0.0.6

1.1627.1 Available under license:

No license file was found, but licenses were detected in source scan.

```javascript
```
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers.

## Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0. To find the details that apply to this artifact see the accompanying LICENSE file.

## Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1630 oauth-sign 0.8.2 0.8.2

1.1630.1 Available under license:
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to
You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and
assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1631 node-ensure 0.0.0

1.1631.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1632 source-map-resolve 0.5.3

1.1632.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014, 2015, 2016, 2017, 2018, 2019 Simon Lydell
Copyright (c) 2019 ZHAO Jinxiang

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1633 isarray 2.0.1

1.1633.1 Available under license:

No license file was found, but licenses were detected in source scan.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1169345659_1622208380.2/0/juliangruber-isarray-v2-0-1-0-gd7f0058-1-tar-gz/juliangruber-isarray-1d71795/README.md
1.1634 @babel/plugin-transform-exponentiation-operator 7.12.13
1.1634.1 Available under license :
   MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1635 brace-expansion 1.1.11
1.1635.1 Available under license :
   MIT License

Copyright (c) 2013 Julian Gruber <julian@juliangruber.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.1636 anymatch 3.1.1
1.1636.1 Available under license:
   The ISC License

Copyright (c) 2019 Elan Shanker, Paul Miller (https://paulmillr.com)

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE.

1.1637 1to2 1.0.0 1.0.0
1.1637.1 Available under license:
   The MIT License (MIT)
   Copyright (c) 2018 NAN contributors

NAN contributors listed at https://github.com/nodejs/nan#contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1638 wrap-ansi 5.1.0
1.1638.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1639 graceful-fs 4.2.4
1.1639.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter, Ben Noordhuis, and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.1640 is-npm 4.0.0

1.1640.1 Available under license:
   MIT License

Copyright (c) Sindresorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1641 cron-parser 2.7.3

1.1641.1 Available under license:
   The MIT License (MIT)

Copyright (c) 2014-2016 Harri Siirak

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1642 array-slice 1.1.0

1.1642.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1643 deep-is 0.1.3

1.1643.1 Available under license:
Copyright (c) 2012, 2013 Thorsten Lorenz <thlorenz@gmx.de>
Copyright (c) 2012 James Halliday <mail@substack.net>
Copyright (c) 2009 Thomas Robinson <280north.com>

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.1644 string.prototype.trimstart 1.0.1

1.1644.1 Available under license:
MIT License

Copyright (c) 2017 Khaled Al-Ansari

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1645 randomatic 3.1.1

1.1645.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1646 locate-path 5.0.0 5.0.0
1.1646.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1647 babel-helper-get-function-arity 6.24.1
1.1647.1 Available under license:
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1648 path-is-absolute 1.0.1 1.0.1

1.1648.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1649 protobuf-java 2.5.31
1.1649.1 Available under license:
No license file was found, but licenses were detected in source scan.

// Copyright 2008 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
//     * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka/protobuf/ServiceException.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka/protobuf/Message.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka/protobuf/ExtensionRegistry.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka/protobuf/BlockingRpcChannel.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka/protobuf/UnmodifiableLazyStringList.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka/protobuf/AbstractMessage.java
Open Source Used In DNAC Platform Antelope 5672

* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/RepeatedFieldBuilder.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/RpcController.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/Parser.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/LazyField.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/UnknownFieldSet.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/SingleFieldBuilder.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/Descriptors.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/GeneratedMessage.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/CodedInputStream.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/WireFormat.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/Service.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/LazyStringArrayList.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/MessageLite.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/RopeByteString.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/ProtocolMessageEnum.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/BoundedByteString.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/GeneratedMessageLite.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/MessageLiteOrBuilder.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/InvalidProtocolBufferException.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/DynamicMessage.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/RpcChannel.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/ExtensionRegistryLite.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/LazyStringList.java
* /opt/ws_local/PERMITS_SQL/1062462128_1595367389.64/0/akka-protobuf-2-12-2-5-31-sources-jar/akka-protobuf/UninitializedMessageException.java
1.1650 jsreport-cli 2.2.5

1.1650.1 Available under license:

MIT License

Copyright (c) 2017 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1651 is-extglob 2.1.1

1.1651.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1652 joni 2.1.31

1.1652.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Permission is hereby granted, free of charge, to any person obtaining a copy of
 * this software and associated documentation files (the "Software"), to deal in
 * the Software without restriction, including without limitation the rights to
 * use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
 * of the Software, and to permit persons to whom the Software is furnished to do
 * so, subject to the following conditions:
 * *
 * The above copyright notice and this permission notice shall be included in all
* copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/constants/internal/StackType.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/CaptureTreeNode.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/BitStatus.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/OPSize.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/exception/SyntaxException.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/UnsetAddrList.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/exception/InternalException.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/constants/internal/AnchorType.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/Search.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/OptAnchorInfo.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/ByteCodeMachine.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/NameEntry.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/StackMachine.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/Regex.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/constants/internal/Arguments.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/Analyser.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/ast/AnchorNode.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/OptMapInfo.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/OptExactInfo.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/constants/internal/EncloseType.java
* /opt/ws_local/PERMITS_SQL/1045445922_1591004900.63/0/joni-2-1-31-sources-jar/org/joni/ast/CClassNode.java
1.1653 events 2.1.0

1.1653.1 Available under license:

MIT License

Copyright (c) 2018 Gnosis Ltd

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1654 fontconfig 2.11.94 0ubuntu1.1

1.1654.1 Available under license:

This package was debianized by Colin Walters <walters@debian.org> on Sun, 13 Oct 2002 15:01:50 -0400

It was downloaded from http://www.fontconfig.org/

Upstream Author: Keith Packard

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Keith Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Keith Packard makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

KEITH PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL KEITH PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2005 Patrick Lam
Copyright 2009 Roozbeh Pournader
Copyright 2008,2009 Red Hat, Inc.
Copyright 2008 Danilo egan
Copyright 2012 Google, Inc.
Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the name of the author(s) not be used in
advertising or publicity pertaining to distribution of the software without
specific, written prior permission. The authors make no
representations about the suitability of this software for any purpose. It
is provided "as is" without express or implied warranty.

THE AUTHOR(S) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY SPECIAL, INDIRECT OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

1.1655 babel-plugin-transform-reserved-words 7.12.13

1.1655.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1656 utils-merge 1.0.1 1.0.1
1.1656.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013-2017 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1657 is-finite 1.1.0
1.1657.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1658 scylla-driver-core 3.10.2-scylla-1

1.1658.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright DataStax, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
/**
 * Special exception that gets emitted to {@link LatencyTracker}s with the latencies of cancelled
 * speculative executions. This allows those trackers to choose whether to ignore those latencies or
 * not.
 */

Found in path(s):
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/CancelledSpeculativeExecutionException.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright DataStax, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
// Copyright 2008 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/VIntCoding.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright DataStax, Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/exceptions/UnsupportedProtocolVersionException.java
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/AbstractReconnectionHandler.java
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/querybuilder/BindMarker.java
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/ClusteringOrder.java
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/AbstractData.java
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/schemabuilder/AlterKeyspace.java
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
jar/com/datastax/driver/core/InboundTrafficMeter.java
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-
Open Source Used In DNAC Platform Antelope 5689
Open Source Used In DNAC Platform Antelope  5690
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Polynomial chosen from https://users.ece.cmu.edu/~koopman/crc/index.html, by Philip Koopman
 * <p>This webpage claims a copyright to Philip Koopman, which he licenses under the Creative
 * Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0)
 * <p>It is unclear if this copyright can extend to a 'fact' such as this specific number,
 * particularly as we do not use Koopman's notation to represent the polynomial, but we anyway
 * attribute his work and link the terms of his license since they are not incompatible with our
 * usage and we greatly appreciate his work.
 * <p>This polynomial provides hamming distance of 8 for messages up to length 105 bits; we only
 * support 8-64 bits at present, with an expected range of 40-48.
 */

Found in path(s):
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-jar/com/datastax/driver/core/Crc.java
No license file was found, but licenses were detected in source scan.

Copyright DataStax, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-jar/META-INF/maven/com.scylladb/scylla-driver-core/pom.xml
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 ScyllaDB
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-jar/com/datastax/driver/core/ShardingInfo.java
No license file was found, but licenses were detected in source scan.

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1202446501_1631567962.06/0/scylla-driver-core-3-10-2-scylla-1-sources-jar/com/datastax/driver/core/Driver.properties

1.1659 abbrev 1.1.1

1.1659.1 Available under license:

This software is dual-licensed under the ISC and MIT licenses.
You may use this software under EITHER of the following licenses.

--------

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

---------

Copyright Isaac Z. Schlueter and Contributors
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1660 lowercase-keys 1.0.1

1.1660.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1661 istanbul-lib-coverage 3.0.0

1.1661.1 Available under license:

Copyright 2012-2015 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1662 jackson 1.0.0

1.1662.1 Available under license:

luminis OSGi bundle

This product includes software developed at luminis (http://opensource.luminis.net/).
Copyright (c) 2006, 2007 luminis
Licensed under the new BSD.

This product includes software developed at
The OSGi Alliance (http://www.osgi.org/).
Copyright (c) 2006 The OSGi Alliance
Licensed under the Apache Software License version 2.0.

This product includes software developed at
aQute SARL (http://www.aqute.biz/).
Copyright (c) 2006 aQute SARL
Licensed under the Apache Software License version 2.0.

/*
$Id: LICENSE.txt,v 1.1.1.1 2004/07/01 13:59:13 jvanzy1 Exp$
*/

Copyright 2002 (C) The Codehaus. All Rights Reserved.

Redistribution and use of this software and associated documentation
("Software"), with or without modification, are permitted provided
that the following conditions are met:

1. Redistributions of source code must retain copyright
   statements and notices. Redistributions must also contain a
   copy of this document.

2. Redistributions in binary form must reproduce the
   above copyright notice, this list of conditions and the
   following disclaimer in the documentation and/or other
   materials provided with the distribution.

3. The name "classworlds" must not be used to endorse or promote
   products derived from this Software without prior written
   permission of The Codehaus. For written permission, please
   contact bob@codehaus.org.

4. Products derived from this Software may not be called "classworlds"
   nor may "classworlds" appear in their names without prior written
   permission of The Codehaus. "classworlds" is a registered
   trademark of The Codehaus.

5. Due credit should be given to The Codehaus.
   (http://classworlds.codehaus.org/).

THIS SOFTWARE IS PROVIDED BY THE CODEHAUS AND CONTRIBUTORS
`AS IS' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE CODEHAUS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/
BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000,2002,2003 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[[]]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
This copy of Jackson JSON processor is licensed under the Lesser General Public License (LGPL), version 2.1 ("the License").
See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.gnu.org/licenses/licenses.html

A copy is also included with both the the downloadable source code package and jar that contains class bytecodes, as file "LGPL 2.1". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/lgpl"; and in jar "META-INF/"
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

== NOTICE file corresponding to the section 4 d of the Apache License, Version 2.0, ==
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

This product includes also software developed by:
- the W3C consortium (http://www.w3c.org),
- the SAX project (http://www.saxproject.org)

Please read the different LICENSE files present in the root directory of this distribution.

Javolution - Java(TM) Solution for Real-Time and Embedded Systems
Copyright (c) 2006, Javolution (http://javolution.org)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Apache License
* Version 2.0, January 2004
* http://www.apache.org/licenses/
* TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
* 1. Definitions.
* "License" shall mean the terms and conditions for use, reproduction,
* and distribution as defined by Sections 1 through 9 of this document.
* "Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."
* "Contributor" shall mean Licensor and any individual or Legal Entity
  on behalf of whom a Contribution has been received by Licensor and
  subsequently incorporated within the Work.

* 2. Grant of Copyright License. Subject to the terms and conditions of
  this License, each Contributor hereby grants to You a perpetual,
  worldwide, non-exclusive, no-charge, royalty-free, irrevocable
  copyright license to reproduce, prepare Derivative Works of,
  publicly display, publicly perform, sublicense, and distribute the
  Work and such Derivative Works in Source or Object form.

* 3. Grant of Patent License. Subject to the terms and conditions of
  this License, each Contributor hereby grants to You a perpetual,
  worldwide, non-exclusive, no-charge, royalty-free, irrevocable
  (except as stated in this section) patent license to make, have made,
  use, offer to sell, sell, import, and otherwise transfer the Work,
  where such license applies only to those patent claims licensable
  by such Contributor that are necessarily infringed by their
  Contribution(s) alone or by combination of their Contribution(s)
  with the Work to which such Contribution(s) was submitted. If You
  institute patent litigation against any entity (including a
  cross-claim or counterclaim in a lawsuit) alleging that the Work
  or a Contribution incorporated within the Work constitutes direct
  or contributory patent infringement, then any patent licenses
  granted to You under this License for that Work shall terminate
  as of the date such litigation is filed.

* 4. Redistribution. You may reproduce and distribute copies of the
  Work or Derivative Works thereof in any medium, with or without
  modifications, and in Source or Object form, provided that You
  meet the following conditions:

  (a) You must give any other recipients of the Work or
  Derivative Works a copy of this License; and

  (b) You must cause any modified files to carry prominent notices
  stating that You changed the files; and

  (c) You must retain, in the Source form of any Derivative Works
  that You distribute, all copyright, patent, trademark, and
  attribution notices from the Source form of the Work,
  excluding those notices that do not pertain to any part of
  the Derivative Works; and

  (d) If the Work includes a "NOTICE" text file as part of its
  distribution, then any Derivative Works that You distribute must
  include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement You may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
It is provided "AS IS" without warranty of any kind, either express or implied.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "["]
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This copy of Jackson JSON processor is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/

A copy is also included with both the the downloadable source code package and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/asl"; and in jar "META-INF/"

/************************************************************************
* CruiseControl, a Continuous Integration Toolkit
* Copyright (c) 2001-2003, ThoughtWorks, Inc.
* 651 W Washington Ave. Suite 500
* Chicago, IL 60661 USA
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*     + Redistributions of source code must retain the above copyright
*       notice, this list of conditions and the following disclaimer.
*
*     + Redistributions in binary form must reproduce the above
*       copyright notice, this list of conditions and the following
*       disclaimer in the documentation and/or other materials provided
*       with the distribution.
*
*     + Neither the name of ThoughtWorks, Inc., CruiseControl, nor the
*       names of its contributors may be used to endorse or promote
*       products derived from this software without specific prior
*       written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
************************************************************************/
Maven Wagon API
Copyright 2003-2006 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Maven Ant Task
Copyright 2002-2008 The Apache Software Foundation

This product includes/uses software(s) developed by ‘an unknown organization’
- Unnamed - ant:ant:jar:1.6.2

This product includes/uses software(s) developed by ‘Apache Software Foundation’ (http://www.apache.org/)
- Maven Wagon File Provider (http://maven.apache.org/wagon/wagon-providers/wagon-file)
- Maven Wagon API (http://maven.apache.org/wagon/wagon-provider-api)
- XML Commons External Components XML APIs (http://xml.apache.org/commons/#external)

This product includes/uses software(s) developed by ‘Codehaus’ (http://www.codehaus.org/)
- Default Plexus Container
- Plexus Common Utilities (http://plexus.codehaus.org/plexus-utils)

This product includes/uses software(s) developed by ‘JTidy’ (http://jtidy.sf.net)
- JTidy (http://jtidy.sourceforge.net)

This product includes/uses software(s) developed by ‘The Apache Software Foundation’ (http://www.apache.org/)
- Maven Artifact (http://maven.apache.org/maven-artifact)
- Maven Artifact Manager (http://maven.apache.org/maven-artifact-manager)
- Maven Error Diagnostics (http://maven.apache.org/maven-error-diagnostics)
- Maven Model (http://maven.apache.org/maven-model)
- Maven Plugin Registry Model (http://maven.apache.org/maven-plugin-registry)
- Maven Profile Model (http://maven.apache.org/maven-profile)
- Maven Project Builder (http://maven.apache.org/maven-project)
- Maven Repository Metadata Model (http://maven.apache.org/maven-repository-metadata)
- Maven Local Settings Model (http://maven.apache.org/maven-settings)
This product includes/uses software(s) developed by 'The Codehaus' (http://codehaus.org/)
- classworlds (http://classworlds.codehaus.org/)
This product currently only contains code developed by authors
of specific components, as identified by the source code files;
if such notes are missing files have been created by
Tatu Saloranta.

For additional credits (generally to people who reported problems)
see CREDITS file.

/* ====================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000-2002 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *    "This product includes software developed by the
 *    Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation", "Jakarta-Oro"
 *    must not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache"
 *    or "Jakarta-Oro", nor may "Apache" or "Jakarta-Oro" appear in their
 *    name, without prior written permission of the Apache Software Foundation.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
*/

Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any,
   must include the following acknowledgment:

   "This product includes software developed by the Indiana University
   Extreme! Lab (http://www.extreme.indiana.edu/)."

   Alternately, this acknowledgment may appear in the software itself,
   if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana University" and "Indiana University Extreme! Lab"
   must not be used to endorse or promote products derived from this
   software without prior written permission. For written permission,
   please contact http://www.extreme.indiana.edu/.

5. Products derived from this software may not use "Indiana University"
   name nor may "Indiana University" appear in their name, without prior
   written permission of the Indiana University.
THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2006, 2007 luminis

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Luminis nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as “Not a Contribution.”

"Contributor” shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
1.1663 oas-resolver 2.5.4 2.5.4
1.1663.1 Available under license:
BSD 3-Clause License

Copyright (c) 2016, Mermade Software
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1664 mv 2.1.1 2.1.1
1.1664.1 Available under license:
Copyright (c) 2014 Andrew Kelley

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation files
(the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge,
publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1665 babel-plugin-syntax-async-generators
6.13.0

1.1665.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1666 shelljs 0.3.0 0.3.0

1.1666.1 Available under license:

Copyright (c) 2012, Artur Adib <aadib@mozilla.com>
All rights reserved.

You may use this project under the terms of the New BSD license as follows:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Artur Adib nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ARTUR ADIB BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1667 domhandler 2.4.2
1.1667.1 Available under license :

Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1668 lodash.isundefined 3.0.1
1.1668.1 Available under license :

Copyright 2012-2015 The Dojo Foundation <http://dojofoundation.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1669 is-data-descriptor 1.0.0
1.1669.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1670 safe-regex 1.1.0
1.1670.1 Available under license :
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1671 syslog-java-client 1.1.7

1.1672 netty 4.1.46.Final
1.1672.1 Available under license:
Copyright (c) 2003-2008 Yuta Mori All Rights Reserved.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

The person or persons who have associated work with this document (the
"Dedicator" or "Certifier") hereby either (a) certifies that, to the best of
his knowledge, the work of authorship identified is in the public domain of
the country from which the work is published, or (b) hereby dedicates whatever
copyright the dedicators holds in the work of authorship identified below (the
"Work") to the public domain. A certifier, moreover, dedicates any copyright
interest he may have in the associated work, and for these purposes, is
described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this
work. Certifier recognizes that his good faith efforts may not shield him from
liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to
the detriment of the Dedicator's heirs and successors. Dedicator intends this
dedication to be an overt act of relinquishment in perpetuity of all present
and future rights under copyright law, whether vested or contingent, in the
Work. Dedicator understands that such relinquishment of all rights includes
the relinquishment of all rights to enforce (by lawsuit or otherwise) those
copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be
freely reproduced, distributed, transmitted, used, modified, built upon, or
otherwise exploited by anyone for any purpose, commercial or non-commercial,
and in any way, including by methods that have not yet been invented or
conceived.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License

Copyright (c) 2009 William Kinney

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
This copy of Aalto XML processor is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/

A copy is also included with both the the downloadable source code package
and jar that contains class bytecodes, as file "ASL 2.0". In both cases,
that file should be located next to this file: in source distribution
the location should be "release-notes/asl"; and in jar "META-INF/"
(BSD License: http://www.opensource.org/licenses/bsd-license)

Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the
following conditions are met:

* Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other
materials provided with the distribution.

* Neither the name of the Webbit nor the names of
its contributors may be used to endorse or promote products
derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Netty Project
================

Please visit the Netty web site for more information:

* https://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

-------------------------------------------------------------------------------

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)

* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
  * HOMEPAGE:
    * http://www.slf4j.org/

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* NOTICE:
  * license/NOTICE.harmony.txt
* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
  * HOMEPAGE:
    * http://archive.apache.org/dist/harmony/

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
  * HOMEPAGE:
    * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
  * HOMEPAGE:
    * https://github.com/y-256/libdivsufsort

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* license/LICENSE.jctools.txt (MIT License)
* HOMEPAGE:
  * https://github.com/javaconcurrencytools/jctools
This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
  * HOMEPAGE:
    * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
  * HOMEPAGE:
    * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
  * HOMEPAGE:
    * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
  * HOMEPAGE:
    * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
  * HOMEPAGE:
    * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:
This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/snappy

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jboss-remoting/jboss-marshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/caliper

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which
can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Cory Benfield. It can be obtained at:

* LICENSE:
  * license/LICENSE.hyper-hpack.txt (MIT License)
* HOMEPAGE:
  * https://github.com/python-hyper/hpack/

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Tatsuhiro Tsujikawa. It can be obtained at:

* LICENSE:
  * license/LICENSE.nghttp2-hpack.txt (MIT License)
* HOMEPAGE:
  * https://github.com/nghttp2/nghttp2/

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/
This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user has everything necessary to run the Maven build.

* LICENSE:
  * license/LICENSE.maven-wrapper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/takari/maven-wrapper

This product contains the dnsinfo.h header file, that provides a way to retrieve the system DNS configuration on MacOS.

This private header is also used by Apple's open source mDNSResponder (https://opensource.apple.com/tarballs/mDNSResponder/).

* LICENSE:
  * license/LICENSE.dnsinfo.txt (Apache License 2.0)
* HOMEPAGE:

The MIT License (MIT)

Copyright (c) 2014 Cory Benfield

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above

---

Open Source Used In DNAC Platform Antelope  5738
copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Harmony

Copyright 2006, 2010 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Copyright (c) 2010-2011 Matthew J. Francis and Contributors of the jbzip2 Project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2009-2010 Ning, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuate of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa
Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc. (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Ymnk, JCraft, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/
*
* Copyright (c) 2004-2007 QOS.ch
* All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
* 
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
* 
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
* /
*/
* Copyright (c) 2004-2006, 2008, 2009, 2011 Apple Inc. All rights reserved.
* *
* @APPLE_LICENSE_HEADER_START@
* *
* This file contains Original Code and/or Modifications of Original Code
* as defined in and that are subject to the Apple Public Source License
* Version 2.0 (the 'License'). You may not use this file except in
* compliance with the License. Please obtain a copy of the License at
* http://www.opensource.apple.com/apsl/ and read it before using this
* file.
* *
* The Original Code and all software distributed under the License are
* distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER
* EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES,
* INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT.
* Please see the License for the specific language governing rights and
* limitations under the License.
* *
* @APPLE_LICENSE_HEADER_END@
* /
Protocol Buffers - Google's data interchange format
Copyright 2013 Google Inc. All rights reserved.
https://developers.google.com/protocol-buffers/

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, 
defend, and hold each Contributor harmless for any liability 
incurred by, or claims asserted against, such Contributor by reason 
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, 
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by 
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all 
other entities that control, are controlled by, or are under common 
control with that entity. For the purposes of this definition, 
"control" means (i) the power, direct or indirect, to cause the 
direction or management of such entity, whether by contract or 
otherwise, or (ii) ownership of fifty percent (50%) or more of the 
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity 
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, 
including but not limited to software source code, documentation 
source, and configuration files.

"Object" form shall mean any form resulting from mechanical 
transformation or translation of a Source form, including but 
not limited to compiled object code, generated documentation, 
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or 
Object form, made available under the License, as indicated by a 
copyright notice that is included in or attached to the work 
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object 
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1673 readable-stream 3.6.0

1.1673.1 Available under license:

Node.js is licensed for use as follows:

```text
===

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===
```

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:
Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1674 jsreport-office 1.2.0
1.1674.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1675 color-name 1.1.4 1.1.4

1.1675.1 Available under license:

The MIT License (MIT)
Copyright (c) 2015 Dmitry Ivanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1676 netty-transport-native-epoll

4.1.42.Final

1.1676.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
*    http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* Set the {@code TCP_MD5SIG} option on the socket. See @code linux/tcp.h} for more details.
* Keys can only be set on, not read to prevent a potential leak, as they are confidential.
* Allowing them being read would mean anyone with access to the channel could get them.
* /
/**
* Set the {@code TCP_QUICKACK} option on the socket. See <a href="http://linux.die.net/man/7/tcp">TCP_QUICKACK</a>
* for more details.
*/

Found in path(s):
*/opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-
jar/io/netty/channel/epoll/EpollSocketChannelConfig.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
*/opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-
jar/io/netty/channel/epoll/EpollDatagramChannelConfig.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
*/opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-
jar/io/netty/channel/epoll/EpollTcpInfo.java
No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-
  jar/netty_epoll_native.c
* /opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-
  jar/io/netty/channel/epoll/Native.java
No license file was found, but licenses were detected in source scan.

/
* Copyright 2014 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
 * Set the {TCP_MD5SIG} option on the socket. See {linux/tcp.h} for more details.
 * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
 * Allowing them being read would mean anyone with access to the channel could get them.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-
  jar/io/netty/channel/epoll/EpollServerSocketChannelConfig.java
No license file was found, but licenses were detected in source scan.

/
* Copyright 2016 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-jar/netty_epoll_linuxsocket.h
* /opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-jar/netty_epoll_linuxsocket.c
* /opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-jar/io/netty/channel/epoll/LinuxSocket.java
* /opt/ws_local/PERMITS_SQL/1072871039_1595214875.08/0/netty-transport-native-epoll-4-1-42-final-sources-1-jar/io/netty/channel/epoll/NativeStaticallyReferencedJniMethods.java

1.1677 qs 6.5.2 6.5.2

1.1677.1 Available under license:

Copyright (c) 2014 Nathan LaFreniere and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*   *   *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors

1.1678 parser-core 5.7

1.1678.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Apache HTTPD & NGINX Access log parsing made easy
 * Copyright (C) 2011-2019 Niels Basjes
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446034_1631568081.13/0/parser-core-5-7-sources-jar/nl/basjes/parse/core/exceptions/DissectionFailure.java
* /opt/cola/permits/1202446034_1631568081.13/0/parser-core-5-7-sources-jar/nl/basjes/parse/core/Parsable.java
* /opt/cola/permits/1202446034_1631568081.13/0/parser-core-5-7-sources-jar/nl/basjes/parse/core/Casts.java
* /opt/cola/permits/1202446034_1631568081.13/0/parser-core-5-7-sources-jar/nl/basjes/parse/core/ParsedField.java
* /opt/cola/permits/1202446034_1631568081.13/0/parser-core-5-7-sources-jar/nl/basjes/parse/core/Parser.java
* /opt/cola/permits/1202446034_1631568081.13/0/parser-core-5-7-sources-jar/nl/basjes/parse/core/Dissector.java
* /opt/cola/permits/1202446034_1631568081.13/0/parser-core-5-7-sources-jar/nl/basjes/parse/core/exceptions/FatalErrorDuringCallOfSetterMethod.java
* /opt/cola/permits/1202446034_1631568081.13/0/parser-core-5-7-sources-jar/nl/basjes/parse/core/Field.java
1.1679 swagger-schema-official 2.0.0-bab6bed 2.0.0-bab6bed

1.1679.1 Available under license:

The MIT License (MIT)
Copyright (c) 2012 Mike Ihbe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1680 parquet-jackson 1.12.0

1.1680.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licenser" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
1.1682.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Andres Suarez <zertosh@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1683 httpclient5 5.0.3

1.1683.1 Available under license:

Apache HttpClient

Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1684 babel-plugin-polyfill-corejs3 0.1.7
1.1684.1 Available under license:
MIT License

Copyright (c) 2014-present Nicol Ribaudo and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1685 socket.io-adapter 1.1.2
1.1685.1 Available under license:
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1686 find-up 3.0.0
1.1686.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1687 micromatch 3.1.10
1.1687.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1688 mimic-fn 1.2.0
1.1688.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1689 mime-types 2.1.27
1.1689.1 Available under license :
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1690 socket.io-parser 2.3.1 2.3.1

1.1690.1 Available under license :
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1691 tomcat-annotations-api 8.5.46

1.1691.1 Available under license :
Apache Tomcat
Copyright 1999-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

Open Source Used In DNAC Platform Antelope 5779
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
1.1692 jsreport-resources 2.1.0

1.1692.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.
You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.1693 httpasyncclient 4.1.3

1.1693.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
Apache HttpComponents AsyncClient
Copyright 2010-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.1694 batik-css 1.13
1.1694.1 Available under license:

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-jar/org/apache/batik/css/engine/value/resources/Messages.properties
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-jar/org/apache/batik/css/parser/resources/Messages.properties
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-jar/org/apache/batik/css/engine/resources/Messages.properties

No license file was found, but licenses were detected in source scan.
jar/org/apache/batik/css/engine/sac/CSSDirectAdjacentSelector.java
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-
jar/org/apache/batik/css/parser/Parser.java
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-
jar/org/apache/batik/css/parser/ScannerUtilities.java
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-
jar/org/apache/batik/css/engine/CSSStyleSheetNode.java
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-
jar/org/apache/batik/css/engine/FontFaceRule.java
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-
jar/org/apache/batik/css/engine/sac/CSSAndCondition.java
* /opt/cola/permits/1135872673_1613620733.01/0/batik-css-1-13-sources-1-
jar/org/apache/batik/css/engine/value/svg12/ICCNamedColor.java

1.1695 lodash-isfunction 3.0.9

1.1695.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

The MIT License (MIT)

Copyright 2012-2016 The Dojo Foundation <http://dojofoundation.org/>
Based on Underscore.js, copyright 2009-2016 Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2012-2015 The Dojo Foundation <http://dojofoundation.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====
Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

Copyright 2012-2016 The Dojo Foundation <http://dojofoundation.org/>
Based on Underscore.js, copyright 2009-2016 Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2012-2015 The Dojo Foundation <http://dojofoundation.org/>
Based on Underscore.js 1.7.0, copyright 2009-2015 Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.


1.1696.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1697 jersey-entity-filtering 2.25.1

1.1697.1 Available under license:

CDDL+GPL License
http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.1698 whatwg-fetch 3.4.1

1.1698.1 Available under license:

Copyright (c) 2014-2016 GitHub, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1699 tephra-api 0.6.0
1.1699.1 Available under license:
   Apache License
   Version 2.0, January 2004
   http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices
that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and
description of purpose be included on the same "printed page" as the copyright notice for easier identification within
third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1700 jcommander 1.30

1.1700.1 Available under license:
No license file was found, but licenses were detected in source scan.

/**
* Copyright (C) 2011 the original author or authors.
* See the notice.md file distributed with this work for additional
* information regarding copyright ownership.
*
* Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/IParameterValidator.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/validators/PositiveInteger.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/validators/NoValidator.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/validators/NoValueValidator.java

No license file was found, but licenses were detected in source scan.

/**
 * Copyright (C) 2010 the original author or authors.
 * See the notice.md file distributed with this work for additional
 * information regarding copyright ownership.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/converters/IntegerConverter.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/ParametersDelegate.java
Open Source Used In DNAC Platform Antelope 5808

* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/MissingCommandException.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/ResourceBundle.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/converters/BooleanConverter.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/converters/FloatConverter.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/converters/DoubleConverter.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/converters/LongConverter.java
* /opt/cola/permits/1136701934_1613865330.83/0/jcommander-1-30-sources-1-jar/com/beust/jcommander/converters/BigDecimalConverter.java
1.1701 lodash.flattendeep 4.4.0

1.1701.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/
Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1702 c3p0 0.9.5.4
1.1702.1 Available under license:

Found license 'Eclipse Public License 1.0' in 'This library is free software; you can redistribute it and/or modify 1) The GNU Lesser General Public License (LGPL), version 2.1, as published by the Free Software Foundation 2) The Eclipse Public License (EPL), version 1.0 This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. LGPL v2.1: http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html' Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:

i) changes to the Program, and
ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.
2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:
a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS),
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or
enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such
provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit)
alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes
such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such
litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or
conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and
distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this
Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the
Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the
right to publish new versions (including revisions) of this Agreement from time to time. No one other than the
Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement
Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable
separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program
(including Contributions) may always be distributed subject to the version of the Agreement under which it was
received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the
Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b)
above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this
Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United
States of America. No party to this Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.
Found license 'GNU Lesser General Public License' in 'This library is free software; you can redistribute it and/or
modify 1) The GNU Lesser General Public License (LGPL), version 2.1, as published by the Free Software
Foundation 2) The Eclipse Public License (EPL), version 1.0 This software is distributed in the hope that it will be
useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. LGPL v2.1: http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html'
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Codec
Copyright 2002-2019 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (https://www.apache.org/).

Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===============================================================================
The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

1.1704 promise 2.1.0
1.1704.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Martin Poelstra <martin@beryllium.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1705 ansi-regex 3.0.0
1.1705.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1706 @babel/runtime 7.0.0-beta.44 7.0.0-beta.44
1.1706.1 Available under license :
MIT License

Copyright (c) 2014-2018 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1707 yu p 0.29.3 0.29.3
1.1707.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Jason Quense

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1708 p-locate 4.1.0
1.1708.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1709 quartz 2.3.2

1.1709.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices.
that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. 

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be
 enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and
description of purpose be included on the same "printed page" as the copyright notice for easier identification within
third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1710 request 2.88.2 2.88.2

1.1710.1 Available under license :
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of
fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source
code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   You must give any other recipients of the Work or Derivative Works a copy of this License; and

   You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.
1.1711 batik-dom 1.13

1.1711.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2005 World Wide Web Consortium,
 *
 * (Massachusetts Institute of Technology, European Research Consortium for
 * Informatics and Mathematics, Keio University). All Rights Reserved. This
 * work is distributed under the W3C(r) Software License [1] in the hope that
 * it will be useful, but WITHOUT ANY WARRANTY; without even the implied
 * warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 *
 *
 * Modifications:
 *
 * September 10, 2005
 * Placed interface in org.apache.batik.dom.xbl for the time being.
 * Added javadocs.
 */

Found in path(s):
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-
  jar/org/apache/batik/dom/xbl/ShadowTreeEvent.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-
  jar/org/apache/batik/dom/xbl/NodeXBL.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-
  jar/org/apache/batik/dom/xbl/XBLShadowTreeElement.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-
  jar/org/apache/batik/dom/xbl/OriginalEvent.java

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-
  jar/org/apache/batik/dom/resources/Messages.properties

No license file was found, but licenses were detected in source scan.
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/GenericProcessingInstruction.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/events/DOMKeyboardEvent.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/ExtendedNode.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/events/DOMMouseEvent.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/AbstractCharacterData.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/events/DOMEvent.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/AbstractElement.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/events/DOMMutationNameEvent.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/StyleSheetProcessingInstruction.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/ExtensibleDOMImplementation.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/GenericDocument.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/traversal/DOMNodeIterator.java
* /opt/cola/permits/1146878549_1616835234.92/0/batik-dom-1-13-sources-1-jar/org/apache/batik/dom/util/XLinkSupport.java
Open Source Used In DNAC Platform Antelope 5830
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

1.1712 pathval 1.1.1
1.1712.1 Available under license :
MIT License

Copyright (c) 2011-2013 Jake Luer jake@alogicalparadox.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO
THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
 WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1713 chalk 0.4.0 0.4.0

1.1713.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1714 buffer-from 1.1.1
1.1714.1 Available under license:
MIT License

Copyright (c) 2016, 2018 Linus Unnebeck

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1715 glob-parent 2.0.0
1.1715.1 Available under license:
The ISC License

Copyright (c) 2015 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1716 jersey-common 2.5.1
1.1716.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty
percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.
3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of
this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as
"Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a
Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes
any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if
the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall,
upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day
notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software
against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or
indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation
of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections
2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted
by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall
survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING
NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER
CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF
SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR
CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES
FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY
AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN
INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT
APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S
NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME
JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR
CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of
"commercial computer software" (as that term is defined at 48 C.F.R. " 252.227-7014(a)(1)) and "commercial
computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48
Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and
supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer
software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is
held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.
When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.
To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

""CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that
accompanied this code.”

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.1717 colord 1.2.12-1ubuntu1

1.1717.1 Available under license:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.
For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

**GNU GENERAL PUBLIC LICENSE**

**TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices
      stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in
      whole or in part contains or is derived from the Program or any
      part thereof, to be licensed as a whole at no charge to all third
      parties under the terms of this License.

   c) If the modified program normally reads commands interactively
      when run, you must cause it, when started running for such
      interactive use in the most ordinary way, to print or display an
      announcement including an appropriate copyright notice and a
      notice that there is no warranty (or else, saying that you provide
      a warranty) and that users may redistribute the program under
      these conditions, and telling the user how to view a copy of this
      License. (Exception: if the Program itself is interactive but
      does not normally print such an announcement, your work based
      on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.
The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.
Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

&l;one line to give the program's name and a brief idea of what it does.&g;  
Copyright (C) &l;year&g;  &l;name of author&g;  

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

&l;signature of Ty Coon&g;, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

1.1718 jna 3.5.1

1.1718.1 Available under license:

Found license 'GNU Lesser General Public License' in "/* Copyright (c) 2007 Wayne Meissner, All Rights Reserved
This library * is free software; you can redistribute it and/or modify it under the * terms of the GNU Lesser General Public License as published by the * Free Software Foundation; either version 2.1 of the License, or (at * your option) any later version. <p> This library is distributed in * PARTICULAR PURPOSE. See the GNU Lesser General Public License for * more details.'

Found license 'GNU Lesser General Public License' in "/* Copyright (c) 2007 Wayne Meissner, All Rights Reserved
* Copyright (c) 2007, 2008, 2009 Timothy Wall, All Rights Reserved * This library is free software; you can redistribute it and/or * modify it under the terms of the GNU Lesser General Public * License as published by the Free Software Foundation; either * version 2.1 of the License, or (at your option) any later version. * This library is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU * Lesser General Public License for more details.'

Found license 'GNU Lesser General Public License' in "/* Copyright (c) 2007 Timothy Wall, All Rights Reserved * This library is free software; you can redistribute it and/or * modify it under the terms of the GNU Lesser General Public * License as published by the Free Software Foundation; either * version 2.1 of the License, or (at your option) any later version. * This library is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU * Lesser General Public License for more details.'

Found license 'GNU Lesser General Public License' in "/* Copyright (c) 2012 Timothy Wall, All Rights Reserved *
This library is free software; you can redistribute it and/or * modify it under the terms of the GNU Lesser General Public * License as published by the Free Software Foundation; either * version 2.1 of the License, or (at your option) any later version. * This library is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU * Lesser General Public License for more details.'

Found license 'GNU Lesser General Public License' in "/* Copyright (c) 2012 Timothy Wall, All Rights Reserved *
This library is free software; you can redistribute it and/or * modify it under the terms of the GNU Lesser General Public * License as published by the Free Software Foundation; either * version 2.1 of the License, or (at your option) any later version. * This library is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU * Lesser General Public License for more details.'
1.1719 bl 2.2.1 2.2.1

1.1719.1 Available under license:

The MIT License (MIT)

==================================

Copyright (c) 2013-2018 bl contributors

==================================

*bl contributors listed at <https://github.com/rvagg/bl#contributors>*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1720 kind-of 6.0.3
1.1720.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1721 babel-register 6.26.0 6.26.0
1.1721.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1722 proxy-addr 1.0.10

1.1722.1 Available under license:
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1723 estraverse 1.9.3

1.1723.1 Available under license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1724 drill-kudu-storage 1.19.0

1.1724.1 Available under license:

Drill : Contrib : Storage : Kudu
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1725 pupa 2.1.1
1.1725.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1726 ini 1.3.7
1.1726.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1727 should-equal 2.0.0 2.0.0
1.1727.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1728 react-intl 2.4.0 2.4.0

1.1728.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1729 jackson-xc 2.9.10

1.1729.1 Available under license:

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions): but is always available from the source code management (SCM) system project uses. This copy of Jackson JSON processor `jackson-module-jaxb-annotations` module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.1730 @babel/helper-define-polyfill-provider 0.1.5

1.1730.1 Available under license:

MIT License

Copyright (c) 2014-present Nicol Ribaudo and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1731 drill-rdbms-metastore 1.19.0

1.1731.1 Available under license:

Drill : Metastore : RDBMS
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1732 hbase-logging 2.4.2
1.1732.1 Available under license:

Apache HBase - Logging
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1733 node-forge-flash 0.0.0

1.1733.1 Available under license:

You may use the Forge project under the terms of either the BSD License or the GNU General Public License (GPL) Version 2.

The BSD License is recommended for most projects. It is simple and easy to understand and it places almost no restrictions on what you can do with the Forge project.

If the GPL suits your project better you are also free to use Forge under that license.

You don't have to do anything special to choose one license or the other and you don't have to notify anyone which license you are using. You are free to use this project in commercial projects as long as the copyright header is left intact.

If you are a commercial entity and use this set of libraries in your commercial software then reasonable payment to Digital Bazaar, if you can afford it, is not required but is expected and would be appreciated. If this library saves you time, then it's saving you money. The cost of developing the Forge software was on the order of several hundred hours and tens of thousands of dollars. We are attempting to strike a balance between helping the development community while not being taken advantage of by lucrative commercial entities for our efforts.

New BSD License (3-clause)

Copyright (c) 2010, Digital Bazaar, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Digital Bazaar, Inc. nor the
  names of its contributors may be used to endorse or promote products
  derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL DIGITAL BAZAAR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

----------------------------------------------------------------------------------
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.
For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.1734 fast-safe-stringify 2.0.7 2.0.7
1.1734.1 Available under license :
The MIT License (MIT)

Copyright (c) 2016 David Mark Clements
Copyright (c) 2017 David Mark Clements & Matteo Collina
Copyright (c) 2018 David Mark Clements, Matteo Collina & Ruben Bridgewater

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1735 jline-style 3.12.1
1.1735.1 Available under license :
No license file was found, but licenses were detected in source scan.

2016, the original author or authors.

This software is distributable under the BSD license. See the terms of the BSD license in the documentation provided with this software.

https://opensource.org/licenses/bsd
Found in path(s):
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/META-INF/maven/org.jline/jline-style/pom.xml
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2018, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/Styler.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyleSource.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyleExpression.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyleFactory.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyleColor.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/NopStyleSource.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyleBundle.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyledWriter.java

/*
 * Copyright (c) 2002-2017, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opencv.org/licenses/BSD-3-Clause
 */

Found in path(s):
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/InterpolationHelper.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/MemoryStyleSource.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyleBundleInvocationHandler.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyleResolver.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/NopStyleSource.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyleBundle.java
* /opt/cola/permits/1114600388_1607534029.29/0/jline-style-3-12-1-sources-jar/org/jline/style/StyledWriter.java
1.1736 string-prototype-trimend 1.0.1

1.1736.1 Available under license:
MIT License

Copyright (c) 2017 Khaled Al-Ansari

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1737 gtk 3.18.9-1ubuntu3.3

1.1737.1 Available under license:
GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.
This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is
the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.
The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a
table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.
This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or
link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.
Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.
GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

      (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.
3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.
If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library ‘Frob’ (a library for tweaking knobs) written by James Random Hacker.
<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.1738 @babel/plugin-transform-computed-properties 7.13.0
1.1738.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1739 source-map-url 0.4.0
1.1739.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1740 qs 4.0.0 4.0.0

1.1740.1 Available under license:

Copyright (c) 2014 Nathan LaFreniere and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors
1.1741 immutability-helper 2.7.1 2.7.1

1.1741.1 Available under license:
MIT License

Copyright (c) 2017 Moshe Kolodny

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1742 tmp 0.0.33

1.1742.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 KARASZI Istvn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1743 guava 19.0

1.1744 extsprintf 1.3.0 1.3.0

1.1744.1 Available under license :

Copyright (c) 2012, Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

1.1745 is-stream 2.0.0

1.1745.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1746 node-extend 2.0.2

1.1746.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Stefan Thomas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1747 readline2 1.0.1

1.1747.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Simon Boudrias

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1748 sparse-bitfield 3.0.3 3.0.3

1.1748.1 Available under license :
   The MIT License (MIT)

   Copyright (c) 2016 Mathias Buus

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1749 xalan 2.7.2

1.1749.1 Available under license :
   SUN PUBLIC LICENSE Version 1.0

   1. Definitions.

   1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.
1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof and corresponding documentation released with the source code.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or
hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated documentation, interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1 The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2)
separate from the Original Code; or 3) for infringements caused by: 
i) the modification of the Original Code or ii) the combination of the 
Original Code with other software or devices.

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor 
hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or 
trademark) Licensable by Contributor, to use, reproduce, modify, 
display, perform, sublicense and distribute the Modifications created 
by such Contributor (or portions thereof) either on an unmodified 
basis, with other Modifications, as Covered Code and/or as part of a 
Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of 
Modifications made by that Contributor either alone and/or in 
combination with its Contributor Version (or portions of such 
combination), to make, use, sell, offer for sale, have made, and/or 
otherwise dispose of: 1) Modifications made by that Contributor (or 
portions thereof); and 2) the combination of Modifications made by 
that Contributor with its Contributor Version (or portions of such 
combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective 
on the date Contributor first makes Commercial Use of the Covered 
Code.

(d) notwithstanding Section 2.2(b) above, no patent license is 
granted: 1) for any code that Contributor has deleted from the 
Contributor Version; 2) separate from the Contributor Version; 3) for 
infringements caused by: i) third party modifications of Contributor 
Version or ii) the combination of Modifications made by that 
Contributor with other software (except as part of the Contributor 
Version) or other devices; or 4) under Patent Claims infringed by 
Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.

The Modifications which You create or to which You contribute are 
governed by the terms of this License, including without limitation 
Section 2.2. The Source Code version of Covered Code may be 
distributed only under the terms of this License or a future version 
of this License released under Section 6.1, and You must include a 
copy of this License with every copy of the Source Code You
distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.


(a) Third Party Claims.

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.
(b) Contributor APIs.

If Contributor's Modifications include an application programming interface ("API") and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may
contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.

Sun Microsystems, Inc. ("Sun") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the
License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Sun. No one other than Sun has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must: (a) rename Your license so that the phrases "Sun," "Sun Public License," or "SPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Sun Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:
(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY
RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.


11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as
Multiple-Licensed? means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

Exhibit A - Sun Public License Notice.

The contents of this file are subject to the Sun Public License Version 1.0 (the "License"); you may not use this file except in compliance with the License. A copy of the License is available at http://www.sun.com/

The Original Code is ________________. The Initial Developer of the Original Code is ___________. Portions created by _____ are Copyright (C)_________. All Rights Reserved.

Contributor(s): _____________________________________________________________________.

Alternatively, the contents of this file may be used under the terms of the _____ license (the '?'[___] License?), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [___] License and not to allow others to use your version of this file under the SPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the SPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

This license came from: http://www.w3.org/Consortium/Legal/copyright-documents-20021231

W3C DOCUMENT LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231

Public documents on the W3C site are provided by the copyright holders under the following license. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:
Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright [Sdate-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

--------------------------------------------------------------------------

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on
style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle $Date: 2005/06/03 22:49:13 $

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Geronimo
Copyright 2003-2006 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache OpenEJB
Copyright 1999-2009 The Apache OpenEJB development community

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This license came from: http://www.w3.org/Consortium/Legal/copyright-software-20021231

W3C SOFTWARE NOTICE AND LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.
Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle $Date: 2005/06/03 22:49:13 $
Apache Xalan (Xalan XSLT processor)
Copyright 1999-2014 The Apache Software Foundation

Apache Xalan (Xalan serializer)
Copyright 1999-2012 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software was originally based on the following:
- software copyright (c) 1999-2002, Lotus Development Corporation.,
- software copyright (c) 2001-2002, Sun Microsystems.,
- software copyright (c) 2003, IBM Corporation.,

The binary distribution package (ie. jars, samples and documentation) of
this product includes software developed by the following:

- The Apache Software Foundation
  - Xerces Java - see LICENSE.txt
  - JAXP 1.3 APIs - see LICENSE.txt
  - Bytecode Engineering Library - see LICENSE.txt
  - Regular Expression - see LICENSE.txt

- Scott Hudson, Frank Flannery, C. Scott Ananian
  - CUP Parser Generator runtime (javacup\runtime) - see LICENSE.txt

The source distribution package (ie. all source and tools required to build
Xalan Java) of this product includes software developed by the following:

- The Apache Software Foundation
  - Xerces Java - see LICENSE.txt
  - JAXP 1.3 APIs - see LICENSE.txt
  - Bytecode Engineering Library - see LICENSE.txt
  - Regular Expression - see LICENSE.txt
  - Ant - see LICENSE.txt
  - Stylebook doc tool - see LICENSE.txt

- Elliot Joel Berk and C. Scott Ananian
  - Lexical Analyzer Generator (JLex) - see LICENSE.txt
Apache Xerces Java
Copyright 1999-2006 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of Apache Xerces Java in xercesImpl.jar and xml-apis.jar
were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

Apache xml-commons xml-apis (redistribution of xml-apis.jar)

Apache XML Commons

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

The license above applies to this Apache Xalan release of:
Xalan-Java 2 - XSLT Processor
Xalan-Java 2 - Serializer

The license above also applies to the jar files
xalan.jar and xsltc.jar - Xalan-Java 2 - XSLT Processor from
Source: http://xalan.apache.org/

The license above also applies to the jar file
serializer.jar - Xalan-Java 2 - Serializer
Source: http://xalan.apache.org/
Used by: Xalan-Java 2 and Xerces-Java 2

The license above also applies to the jar file
xercesImpl.jar - Xerces-Java 2 XML Parser.
Source: http://xerces.apache.org/
Used by: Xalan-Java 2
The license above also applies to the jar file xml-apis.jar - Xerces-Java 2 XML Parser.
Source: http://xerces.apache.org/
Used by: Xalan-Java 2 and release copy of Xerces-Java 2

The following license applies to the included files:
tools/ant.jar
tools/antRun
tools/antRun.bat
Source: http://ant.apache.org/
Used By: Xalan's build process: java/build.xml and test/build.xml

<<<<<<<<<<< The Apache Software License, Version 1.1
/*
 * Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 *
 * Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 *
 * The end-user documentation included with the redistribution, if any, must
 * include the following acknowledgment: "This product includes software
 * developed by the Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgment may appear in the software itself, if
 * and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Ant" and "Apache Software Foundation" must not be used to
 * endorse or promote products derived from this software without prior
 * written permission. For written permission, please contact
 * apache@apache.org.
 */
* 5. Products derived from this software may not be called "Apache", nor may
* "Apache" appear in their name, without prior written permission of the
* Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
* INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
* FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
* INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUD-}
* DING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
* OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
* ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* This software consists of voluntary contributions made by many individuals
* on behalf of the Apache Software Foundation. For more information on the
* Apache Software Foundation, please see <http://www.apache.org/>.
* *
* /*
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

The following license, Apache Software License, Version 1.1,
applies to the included BCEL.jar from Apache Jakarta
(Byte Code Engineering Library).
Source:  http://jakarta.apache.org/bcel
Used By: XSLTC component of xml-xalan/java

The following license, Apache Software License, Version 1.1,
also applies to the included regexp.jar,
jakarta-regexp-1.2.jar from Apache Jakarta.
Source: http://jakarta.apache.org/regexp
Used By: BCEL.jar which is used by XSLTC component of xml-xalan/java

<<<<<<<<<
/*
*
* Copyright (c) 2001 The Apache Software Foundation. All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.
* 
* 3. The end-user documentation included with the redistribution,
* if any, must include the following acknowledgment:
* "This product includes software developed by the
* Apache Software Foundation (http://www.apache.org/)."
* Alternately, this acknowledgment may appear in the software itself,
* if and wherever such third-party acknowledgments normally appear.
* 
* 4. The names "Apache" and "Apache Software Foundation" and
* "Apache BCEL" must not be used to endorse or promote products
* derived from this software without prior written permission. For
* written permission, please contact apache@apache.org.
* 
* 5. Products derived from this software may not be called "Apache",
* "Apache BCEL", nor may "Apache" appear in their name, without
* prior written permission of the Apache Software Foundation.
* 
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* 
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
*/

Open Source Used In DNAC Platform Antelope 5942
The following license applies to the DOM documentation
for the org.w3c.dom.* packages:

W3C DOCUMENT LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231
Public documents on the W3C site are provided by the copyright holders
under the following license. By using and/or copying this document,
or the W3C document from which this statement is linked, you (the licensee)
agree that you have read, understood, and will comply with the following
terms and conditions:

Permission to copy, and distribute the contents of this document, or the
W3C document from which this statement is linked, in any medium for any
purpose and without fee or royalty is hereby granted, provided that you include
the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it
doesn't exist, a notice (hypertext is preferred, but a textual representation
is permitted) of the form: "Copyright [date-of-document] World Wide Web
Consortium, (Massachusetts Institute of Technology, European Research
Consortium for Informatics and Mathematics, Keio University). All Rights
Reserved. http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided.
We request that authorship attribution be provided in any software, documents,
or other items or products that you create pursuant to the implementation of the
contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant
to this license. However, if additional requirements (documented in the Copyright FAQ)
are satisfied, the right to create modifications or derivatives is sometimes granted
by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS
OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE;
THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE
IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS.
COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

--------------------------------------------------------------------------------

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <mailto:site-policy@w3.org>
Last revised by Reagle $Date: 2005-07-19 12:33:09 -0400 (Tue, 19 Jul 2005) $

>>>------------------------------------------------------------------------

The following license applies to the DOM software,
for the org.w3c.dom.* packages in jar file xml-apis.jar:

<<<------------------------------------------------------------------------

W3C SOFTWARE NOTICE AND LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231
This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this
work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

____________________________________
This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.
The following license applies to the SAX software,
for the org.xml.sax.* packages in jar file xml-apis.jar:

This module, both source code and documentation, is in the Public Domain,
and comes with NO WARRANTY. See http://www.saxproject.org for further information.

The following license applies to the jar file
java_cup.jar - LALR Parser Generator for Java(TM).
Source:  http://www.cs.princeton.edu/~appel/modern/java/CUP
Used By: XSLTC component of xml-xalan/java

CUP Parser Generator Copyright Notice, License, and Disclaimer

Copyright 1996-1999 by Scott Hudson, Frank Flannery, C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted, provided
that the above copyright notice appear in all copies and that both
the copyright notice and this permission notice and warranty disclaimer
appear in supporting documentation, and that the names of the authors
or their employers not be used in advertising or publicity pertaining
to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to
this software, including all implied warranties of merchantability
and fitness. In no event shall the authors or their employers be liable
for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

The following license applies to the jar file runtime.jar - Component of JavaCup: LALR Parser Generator for Java(TM).
Source: http://www.cs.princeton.edu/~appel/modern/java/CUP
Used By: XSLTC component of xml-xalan/java

CUP Parser Generator Copyright Notice, License, and Disclaimer (runtime.jar component)

Copyright 1996-1999 by Scott Hudson, Frank Flannery, C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.
The following license applies to the JLEX jar file  
JLex.jar - A Lexical Analyzer Generator for Java(TM).  
Source: http://www.cs.princeton.edu/~appel/modern/java/JLex  
Used By: XSLTC component of xml-xalan/java

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

JLEX COPYRIGHT NOTICE, LICENSE AND DISCLAIMER.

Copyright 1996-2000 by Elliot Joel Berk and C. Scott Ananian

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Java is a trademark of Sun Microsystems, Inc. References to the Java programming language in relation to JLex are not meant to imply that Sun endorses this product.

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

The following license applies to the jar file  
stylebook-1.0-b3_xalan-2.jar - Tool for generating Xalan documentation.  
Integrated with Xalan-Java 2 and Xerces 2.  
Source: http://svn.apache.org/viewvc/xml/stylebook/  
Used by: Xalan-Java 2, Xalan-C++

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

/*
* The Apache Software License, Version 1.1
Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:
   "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)."
   Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Xalan", "Xerces", and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation and was
* originally based on software copyright (c) 1999, International
* Business Machines, Inc., http://www.apache.org. For more
* information on the Apache Software Foundation, please see
*/

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered
under http://saxproject.org/?selected=pd

This page is now out of date -- see the new SAX site at
http://www.saxproject.org/ for more up-to-date
releases and other information. Please change your bookmarks.

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for
XML), and release all of the SAX 2.0 source code, compiled code, and
documentation contained in this distribution into the Public Domain.
SAX comes with NO WARRANTY or guarantee of fitness for any
purpose.

David Megginson, david@megginson.com
2000-05-05

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE Version 1.0 (CDDL-1.0)
(text)

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a
Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files
containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this
License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes
or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.
3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the
terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANDABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF
SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

1.1750 babel-plugin-transform-es2015-for-of
6.23.0
1.1750.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>

Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# Database of contributors to Apache Calcite.
# Pages such as developer.md use this data.
# List must be sorted by first name, last name.

- name: Alan Gates
  emeritus: 2018/05/04
  apacheId: gates
  githubId: alanfgates
  org: Hortonworks
  role: PMC
- name: Aman Sinha
  apacheId: amansinha
  githubId: amansinha100
  org: MapR
  role: PMC
- name: Andrei Sereda
  apacheId: sereda
  githubId: asereda-gs
  org: ""
  role: Committer
- name: Ashutosh Chauhan
  apacheId: hashutosh
  githubId: ashutoshc
  org: Hortonworks
  role: PMC
- name: Chris Baynes
- name: Christian Beikov
  apacheId: cbeikov
  githubId: beikov
  org: Blazebit
  role: Committer
- name: Chunwei Lei
  apacheId: chunwei
  githubId: chunweilei
  org: Alibaba
  role: Committer
- name: Danny Chan
  apacheId: danny0405
  githubId: danny0405
  org: Alibaba
  role: Committer
- name: Edmon Begoli
  apacheId: ebegoli
  githubId: ebegoli
  org: Oak Ridge National Laboratory
  role: Committer
- name: Francis Chuang
  apacheId: francischuang
  githubId: F21
  org: Boostport
  role: PMC Chair
- name: Gian Merlino
  apacheId: gian
  githubId: gianm
  org: Imply
  role: Committer
- name: Haisheng Yuan
  apacheId: hyuan
  githubId: hsyuan
  org: Alibaba
  role: Committer
- name: Hongze Zhang
  apacheId: hongze
  githubId: zhztheplayer
  org: Tencent
  role: Committer
- name: James R. Taylor
  apacheId: jamesstaylor
  githubId: JamesRTaylor
  org: Salesforce
role: PMC
- name: Jacques Nadeau
  apacheId: jacques
  githubId: jacques-n
  org: Dremio
  role: PMC
- name: Jess Balint
  apacheId: jbalint
  githubId: jbalint
  org: Stardog
  role: Committer
- name: Jess Camacho Rodriguez
  apacheId: jcamacho
  githubId: jcamachor
  org: Hortonworks
  role: PMC
- name: Jinfeng Ni
  apacheId: jni
  githubId: jinfengni
  org: MapR
  role: PMC
- name: John Pullokkaran
  apacheId: jpullokk
  githubId: jpullokkaran
  org: Hortonworks
  role: PMC
- name: Josh Elser
  apacheId: elserj
  githubId: joshelser
  org: Hortonworks
  role: PMC
- name: Julian Hyde
  apacheId: jhyde
  githubId: julianhyde
  org: Looker
  role: PMC
  homepage: http://people.apache.org/~jhyde
- name: Kevin Liew
  apacheId: kliew
  githubId: kliewkliew
  org:
  role: Committer
- name: Kevin Risden
  apacheId: krisden
  githubId: risdenk
  org:
  role: PMC
- name: Laurent Goujon
- name: Maryann Xue
  apacheId: maryannxue
  githubId: maryannxue
  org: Intel
  role: PMC
- name: Michael Mior
  apacheId: mmior
  githubId: michaelmior
  org: Rochester Institute of Technology
  role: PMC
  homepage: https://michael.mior.ca/
- name: Milinda Pathirage
  apacheId: milinda
  githubId: milinda
  org: Indiana University
  role: Committer
  homepage: http://milinda.pathirage.org/
- name: MinJi Kim
  apacheId: minji
  githubId: minji-kim
  org: Oracle
  role: Committer
- name: Nick Dimiduk
  apacheId: ndimiduk
  githubId: ndimiduk
  role: PMC
  homepage: http://www.n10k.com
- name: Nishant Bangarwa
  apacheId: nishant
  githubId: nishantmonu51
  org: Hortonworks
  role: Committer
- name: Ruben Quesada Lopez
  apacheId: rubenql
  githubId: rubenada
  org: TIBCO
  role: Committer
- name: Sergey Nuyanzin
  apacheId: snuyanzin
  githubId: snuyanzin
  org: EPAM
  role: Committer
- name: Shuyi Chen
apacheId: shuyichen
githubId: suez1224
org: Uber
role: Committer
- name: Slim Bouguerra
  apacheId: bslim
githubId: b-slim
org: Hortonworks
role: Committer
- name: Stamatis Zampetakis
  apacheId: zabetak
githubId: zabetak
org: TIBCO
role: PMC
- name: Steven Noels
  apacheId: stevenn
githubId: stevenn
org: NGData
role: PMC
- name: Ted Dunning
  apacheId: tdunning
githubId: tdunning
org: MapR
role: PMC
  avatar: https://mapr.com/blog/author/ted-dunning/assets/tdunning-panama.jpg
- name: Vladimir Sitnikov
  apacheId: vladimirsitnikov
githubId: vlsi
org: NetCracker
role: PMC
- name: Volodymyr Vysotskyi
  apacheId: volodymyr
githubId: vvysotskyi
org:
role: PMC
- name: Zhen Wang
  apacheId: zhenw
githubId: zinking
org:
role: Committer
- name: Zhiqiang He
  apacheId: zhiqianghe
githubId: Zhiqiang-He
org: Huawei
role: Committer
- name: Zhiwei Peng
  apacheId: zhiwei
githubId: pengzhiwei2018
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-----------------------------------------------------------------------

APACHE CALCITE SUBCOMPONENTS:

The Apache Calcite project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

-----------------------------------------------------------------------

The MIT License

-----------------------------------------------------------------------

The Apache Calcite project bundles the following files under the MIT License:

- site
  Parts of the web site generated by Jekyll (http://jekyllrb.com/)
  Copyright (c) 2008-2015 Tom Preston-Werner
- site/_sass/_font-awesome.scss
  Font-awesome css files v4.1.0 (http://fortawesome.github.io/Font-Awesome/)
  Copyright (c) 2013  Dave Gandy
- site/_sass/_normalize.scss
  normalize.css v3.0.2 | git.io/normalize
  Copyright (c) Nicolas Gallagher and Jonathan Neal
- site/_sass/_gridism.scss
Gridism: A simple, responsive, and handy CSS grid by @cobyism
https://github.com/cobyism/gridism
Copyright (c) 2013 Coby Chapple
- site/js/html5shiv.min.js
   HTML5 Shiv 3.7.2 | @afarkas @jdalton @jon_neal @rem
- site/js/respond.min.js
   Respond.js v1.4.2: min/max-width media query polyfill
   Copyright 2013 Scott Jehl

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----------------------------------------------------------------------
The Open Font License
-----------------------------------------------------------------------

The Apache Calcite project bundles the following fonts under the
SIL Open Font License (OFL) - http://scripts.sil.org/OFL/

- site/fonts/fontawesome-webfont.*
  Font-awesome font files v4.0.3 (http://fortawesome.github.io/Font-Awesome/)
  Apache Calcite
  Copyright 2012-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product is based on source code originally developed
by DynamoBI Corporation, LucidEra Inc., SQLstream Inc. and others
under the auspices of the Eigenbase Foundation
and released as the LucidDB project.

The web site includes files generated by Jekyll.
1.1752 content-disposition 0.5.1 0.5.1

1.1752.1 Available under license:

(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1753 emoji-regex 8.0.0 8.0.0

1.1753.1 Available under license:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
1.1754 avro 1.7.6

1.1754.1 Available under license:

Apache Avro
Copyright 2009-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1755 object-assign 3.0.0 3.0.0

1.1755.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1756 cypress 1.0.0 1.0.0

1.1756.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Cypress.io, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```javascript
import cs from 'classnames'
import React, { Component } from 'react'
import { observer } from 'mobx-react'
import ipc from '../lib/ipc'
import errors from '../lib/errors'
import projectsStore from '../projects/projects-store'

const SUCCESS = 'SUCCESS'
const FAILURE = 'FAILURE'

@observer
class PermissionMessage extends Component {
  constructor (props) {
    super(props)

    this.state = {
      error: null,
      isSubmitting: false,
      result: null,
    }
  }
```

```javascript
```
render () {
    return (
        <div
            className={cs('request-access', {
                'is-submitting': this.state.isSubmitting,
            })}
        >
            {this._message()}
        </div>
    )
}

_message () {
    const membershipRequested = projectsStore.wasMembershipRequested(this.props.project.id)

    if (this.state.result === SUCCESS || membershipRequested) {
        return this._success()
    } else if (this.state.result === FAILURE) {
        return this._failure()
    } else {
        return this._noResult()
    }
}

_button () {
    return (
        <button
                className='btn btn-primary'
                disabled={this.state.isSubmitting}
                onClick={this._requestAccess}
                onClick={this._requestAccess}
        >
            <span>
                <i className='fa fa-paper-plane'></i> Request Access
            </span>
            <i className='fa fa-spinner fa-spin'></i>
        </button>
    )
}

_success () {
    return (
        <div className='empty'>
            <h4>
                <i className='fa fa-check passed'></i> Request Sent
            </h4>
        </div>
    )
}
The project owner will be notified with your request.

// if they're denied or the request has already been made,
// tell them it's all good
if (errors.isDenied(error) || errors.isAlreadyRequested(error)) {
    return this._success()
} else {
    return (
        <div className='empty'>
            <h4><i className='fa fa-exclamation-triangle failed'></i> Request Failed</h4>
            <p>An unexpected error occurred while requesting access:</p>
            <pre className='alert alert-danger'>{this.state.error.message}</pre>
            <p>Try again.</p>
            {this._button()}
        </div>
    )
}

_noResult () {
    return (  
        <div className="empty">
            <h4><i className='fa fa-lock'></i> Request access to see the runs</h4>
            <p>This is a private project created by someone else.</p>
            <p>The project owner must give you access to see the runs.</p>
            {this._button()}
        </div>
    )
}

_requestAccess = () => {
    this.setState({

isSubmitting: true,
})

const id = this.props.project.id

ipc.requestAccess(id)
.then(() => {
    projectsStore.membershipRequested(id)
    this._setResult()
})
.catch(ipc.isUnauthed, ipc.handleUnauthed)
.catch((error) => {
    this._setResult(error)
})

_setResult (error) {
    if (errors.isAlreadyMember(error)) {
        this.props.onRetry()
        return
    }

    this.setState({
        error,
        isSubmitting: false,
        result: error ? FAILURE : SUCCESS,
    })
}

export default PermissionMessage

1.1757 jersey-container-servlet 2.5.1
1.1757.1 Available under license :
    CDDL+GPL License
    http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.1758 glob 6.0.4
1.1758.1 Available under license :
    The ISC License

    Copyright (c) Isaac Z. Schlueter and Contributors

    Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1759 proxy-addr 2.0.4 2.0.4

1.1759.1 Available under license :
(The MIT License)

Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1760 thrift 0.7.0

1.1760.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

The following files contain some portions of code contributed under the Thrift Software License (see doc/old-thrift-license.txt), and relicensed under the Apache 2.0 License:

compiler/cpp/Makefile.am
compiler/cpp/src/generate/t_cocoa_generator.cc
compiler/cpp/src/generate/t_cpp_generator.cc
compiler/cpp/src/generate/t_csharp_generator.cc
compiler/cpp/src/generate/t_erl_generator.cc
compiler/cpp/src/generate/t_hs_generator.cc
compiler/cpp/src/generate/t_java_generator.cc
compiler/cpp/src/generate/t_ocaml_generator.cc
compiler/cpp/src/generate/t_perl_generator.cc
compiler/cpp/src/generate/t_php_generator.cc
compiler/cpp/src/generate/t_py_generator.cc
compiler/cpp/src/generate/t_rb_generator.cc
compiler/cpp/src/generate/t_st_generator.cc
compiler/cpp/src/generate/t_xsd_generator.cc
compiler/cpp/src/main.cc
compiler/cpp/src/parse/t_field.h
compiler/cpp/src/parse/t_program.h
compiler/cpp/src/platform.h
compiler/cpp/src/thriftl.ll
compiler/cpp/src/thifty.yy
lib/csharp/src/Protocol/TBinaryProtocol.cs
lib/csharp/src/Protocol/TField.cs
lib/csharp/src/Protocol/TLList.cs
This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch
ghost@aladdin.com

*/

---------------------------------------------------
For the lib/rb/setup.rb: Copyright (c) 2000-2005 Minero Aoki.
lib/ocaml/OCamlMakefile and lib/ocaml/README-OCamlMakefile components:
   Copyright (C) 1999 - 2007 Markus Mottl

Licensed under the terms of the GNU Lesser General Public License 2.1
(see doc/lgpl-2.1.txt for the full terms of this license)
This package was debianized by Thrift Developer's <dev@thrift.apache.org>.

This package and the Debian packaging is licensed under the Apache License,
see `/usr/share/common-licenses/Apache-2.0`.

The following informations was copied from Apache Thrift LICENSE file.

---------------------------------------------------
SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

---------------------------------------------------
Portions of the following files are licensed under the MIT License:
lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

--------------------------------------------------

The following files contain some portions of code contributed under
the Thrift Software License (see doc/old-thrift-license.txt), and relicensed
under the Apache 2.0 License:

compiler/cpp/Makefile.am
compiler/cpp/src/generate/t_cocoa_generator.cc
compiler/cpp/src/generate/t_cpp_generator.cc
compiler/cpp/src/generate/t_csharp_generator.cc
compiler/cpp/src/generate/t_erl_generator.cc
compiler/cpp/src/generate/t_hs_generator.cc
compiler/cpp/src/generate/t_java_generator.cc
compiler/cpp/src/generate/t_ocaml_generator.cc
compiler/cpp/src/generate/t_perl_generator.cc
compiler/cpp/src/generate/t_php_generator.cc
compiler/cpp/src/generate/t_py_generator.cc
compiler/cpp/src/generate/t_rb_generator.cc
compiler/cpp/src/generate/t_st_generator.cc
compiler/cpp/src/generate/t_xsd_generator.cc
compiler/cpp/src/main.cc
compiler/cpp/src/parser/t_field.h
compiler/cpp/src/parser/t_program.h
compiler/cpp/src/platform.h
compiler/cpp/src/thriftl.1l
compiler/cpp/src/thrifty.yy
lib/csharp/src/Protocol/TBinaryProtocol.cs
lib/csharp/src/Protocol/TField.cs
lib/csharp/src/Protocol/TList.cs
lib/csharp/src/Protocol/TMap.cs
lib/csharp/src/Protocol/TMessage.cs
lib/csharp/src/Protocol/TMessageType.cs
lib/csharp/src/Protocol/TProtocol.cs
lib/csharp/src/Protocol/TProtocolException.cs
lib/csharp/src/Protocol/TProtocolFactory.cs
lib/csharp/src/Protocol/TProtocolUtil.cs
lib/csharp/src/Protocol/TSet.cs
lib/csharp/src/Protocol/TStruct.cs
lib/csharp/src/Protocol/TTType.cs
lib/csharp/src/Server/TServer.cs
lib/csharp/src/Server/TSimpleServer.cs
lib/csharp/src/Server/ThreadPoolServer.cs
lib/csharp/src/TApplicationException.cs
lib/csharp/src/Thrift.csproj
For the aclocal/ax_boost_base.m4 and contrib/fb303/aclocal/ax_boost_base.m4 components:

```
#  Copyright (c) 2007 Thomas Porschberg <thomas@randspringer.de>
#
#  Copying and distribution of this file, with or without
#  modification, are permitted in any medium without royalty provided
#  the copyright notice and this notice are preserved.
```

For the compiler/cpp/src/md5.[ch] components:

```
/*
   Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved.

   This software is provided 'as-is', without any express or implied
   warranty. In no event will the authors be held liable for any damages
   arising from the use of this software.

   Permission is granted to anyone to use this software for any purpose,
   including commercial applications, and to alter it and redistribute it
   freely, subject to the following restrictions:

   1. The origin of this software must not be misrepresented; you must not
      claim that you wrote the original software. If you use this software
      in a product, an acknowledgment in the product documentation would be
      appreciated but is not required.
   2. Altered source versions must be plainly marked as such, and must not be
      misrepresented as being the original software.
```
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch
ghost@aladdin.com

/*

---------------------------------------------------
For the lib/rb/setup.rb: Copyright (c) 2000-2005 Minero Aoki,
lib/ocaml/OCamlMakefile and lib/ocaml/README-OCamlMakefile components:
  Copyright (C) 1999 - 2007 Markus Mottl

Licensed under the terms of the GNU Lesser General Public License 2.1
(see doc/lgpl-2.1.txt for the full terms of this license)
Tue Oct 24 12:28:44 CDT 2006

Copyright (c) <2006> <Martin J. Logan, Erlware>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software (OTP Base, fslib, G.A.S) and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE
OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache Thrift
Copyright 2006-2010 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.1761 is-buffer 1.1.6
1.1761.1 Available under license:
The MIT License (MIT)

Copyright (c) Feross Aboukhadieh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1762 resolve 1.18.1

1.1762.1 Available under license:
MIT License

Copyright (c) 2012 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1763 sprintf-js 1.1.0

1.1763.1 Available under license:

Copyright (c) 2007-2016, Alexandru Marasteanu <hello [at] alexei [dot] ro>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of this software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1764 jna 5.8.0

1.1764.1 Available under license:

/*
 * The contents of this file are subject to the Mozilla Public License Version 1.1
 * (the "License"); you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at http://www.mozilla.org/MPL/
 *
 * Software distributed under the License is distributed on an "AS IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
 * for the specific language governing rights and limitations under the License.
 *
 * The Original Code is 'iText, a free JAVA-PDF library'.
 *
 * The Initial Developer of the Original Code is Bruno Lowagie. Portions created by
 * the Initial Developer are Copyright (C) 1999, 2000, 2001, 2002 by Bruno Lowagie.
 * All Rights Reserved.
* Co-Developer of the code is Paulo Soares. Portions created by the Co-Developer
* are Copyright (C) 2000, 2001, 2002 by Paulo Soares. All Rights Reserved.

* Contributor(s): all the names of the contributors are added in the source code
* where applicable.

* Alternatively, the contents of this file may be used under the terms of the
* LGPL license (the "GNU LIBRARY GENERAL PUBLIC LICENSE"), in which case the
* provisions of LGPL are applicable instead of those above. If you wish to
* allow use of your version of this file only under the terms of the LGPL
* License and not to allow others to use your version of this file under
* the MPL, indicate your decision by deleting the provisions above and
* replace them with the notice and other provisions required by the LGPL.
* If you do not delete the provisions above, a recipient may use your version
* of this file under either the MPL or the GNU LIBRARY GENERAL PUBLIC LICENSE.

* This library is free software; you can redistribute it and/or modify it
* under the terms of the MPL as stated above or under the terms of the GNU
* Library General Public License as published by the Free Software Foundation;
* either version 2 of the License, or any later version.

* This library is distributed in the hope that it will be useful, but WITHOUT
* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
* FOR A PARTICULAR PURPOSE. See the GNU Library general Public License for more
* details.

* If you didn't download this code from the following link, you should check if
* you aren't using an obsolete version:
* http://www.lowagie.com/iText/

* This class is generated based on a grammar file provided by SUN, and updated
* by Carsten Hammer. SUN's license agreement can be found at this URL:
* http://java.sun.com/products/java-media/2D/samples/samples-license.html
* See also the file sun.txt in directory com.lowagie.text.pdf
*/

Maven Ant Tasks

Copyright 2002-2011 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

=======================================================================
=======================================================================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from 'org.jfree' to 'clover.org.jfree'. This was necessary to avoid potential name
conflicts during instrumentation of a code using the original library when using Clover. No source code of
the original library was modified.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library": The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:
a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.
7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you...
may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Ant
Copyright 1999-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.

/*
   Apache License
   Version 2.0, January 2004
   http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
*/
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions...
* for use, reproduction, or distribution of Your modifications, or
* for any such Derivative Works as a whole, provided Your use,
* reproduction, and distribution of the Work otherwise complies with
* the conditions stated in this License.
*
* 5. Submission of Contributions. Unless You explicitly state otherwise,
* any Contribution intentionally submitted for inclusion in the Work
* by You to the Licensor shall be under the terms and conditions of
* this License, without any additional terms or conditions.
* Notwithstanding the above, nothing herein shall supersede or modify
* the terms of any separate license agreement you may have executed
* with Licensor regarding such Contributions.
*
* 6. Trademarks. This License does not grant permission to use the trade
* names, trademarks, service marks, or product names of the Licensor,
* except as required for reasonable and customary use in describing the
* origin of the Work and reproducing the content of the NOTICE file.
*
* 7. Disclaimer of Warranty. Unless required by applicable law or
* agreed to in writing, Licensor provides the Work (and each
* Contributor provides its Contributions) on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied, including, without limitation, any warranties or conditions
* of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
* PARTICULAR PURPOSE. You are solely responsible for determining the
* appropriateness of using or redistributing the Work and assume any
* risks associated with Your exercise of permissions under this License.
*
* 8. Limitation of Liability. In no event and under no legal theory,
* whether in tort (including negligence), contract, or otherwise,
* unless required by applicable law (such as deliberate and grossly
* negligent acts) or agreed to in writing, shall any Contributor be
* liable to You for damages, including any direct, indirect, special,
* incidental, or consequential damages of any character arising as a
* result of this License or out of the use or inability to use the
* Work (including but not limited to damages for loss of goodwill,
* work stoppage, computer failure or malfunction, or any and all
* other commercial damages or losses), even if such Contributor
* has been advised of the possibility of such damages.
*
* 9. Accepting Warranty or Additional Liability. While redistributing
* the Work or Derivative Works thereof, You may choose to offer,
* and charge a fee for, acceptance of support, warranty, indemnity,
* or other liability obligations and/or rights consistent with this
* License. However, in accepting such obligations, You may act only
* on Your own behalf and on Your sole responsibility, not on behalf
* of any other Contributor, and only if You agree to indemnify,
* defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

/*

W3C SOFTWARE NOTICE AND LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms

Open Source Used In DNAC Platform Antelope 6009
and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.

3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

____________________________________

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered under http://saxproject.org/?selected=pd

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.
The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from 'com.keypoint/org.jfree' to 'clover.com.keypoint/clover.org.jfree'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.
To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that,
in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.
4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).
To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Lang
Copyright 2001-2011 The Apache Software Foundation
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
/*
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000-2003 The Apache Software Foundation. All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution, if
 *    any, must include the following acknowlegement:
 *    "This product includes software developed by the
 *    Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowlegement may appear in the software itself,
 *    if and wherever such third-party acknowlegements normally appear.
 *
 * 4. The names "Ant" and "Apache Software
 *    Foundation" must not be used to endorse or promote products derived
 *    from this software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache"
 *    nor may "Apache" appear in their names without prior written
 *    permission of the Apache Group.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.
1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

MOZILLA PUBLIC LICENSE
Version 1.1

-------------

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.
1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:
   A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

   B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available.
for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:
   (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and
   (b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).
   (c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.
   (d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license
   (a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an
unmodified basis, with other Modifications, asCovered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12)
months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
(a) Third Party Claims.
If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.
If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.
Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source
Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.
If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS,
WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant
alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.


11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions.
With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is ______________________________________.

The Initial Developer of the Original Code is _________________________.
Portions created by ______________________ are Copyright (C) ______
_____________________. All Rights Reserved.
Contributor(s): ______________________________________.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License.”

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

/*
Copyright (c) 2000, Derek Petillo
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Praxis Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright-Only Dedication (based on United States law) or Public Domain Certification

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Open Source License for the overlibmws Package

1. License coverage

Note that this license only covers the script library (javascript core and plugin modules) and not any supporting material such as the overlibmws website or its online documentation and support files. You may not reproduce the website or its online material without explicit written permission from the author, but can freely incorporate scripts and procedures which are demonstrated in that material into your own HTML or XML documents.

2. License (Artistic)

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

use the modified Package only within your corporation or organization.

rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

make other distribution arrangements with the Copyright Holder.

You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

accompany the distribution with the machine-readable source of the Package with your modifications.

accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

make other distribution arrangements with the Copyright Holder.

You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
==============================================================================

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from 'it.unimi.dsi.fastutil' to 'clover.it.unimi.dsi.fastutil'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

==============================================================================
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.
To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of
Free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:
a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.
If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.
NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANDISABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Javolution - Java(TM) Solution for Real-Time and Embedded Systems
Copyright (c) 2006, Javolution (http://javolution.org)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The libffi source distribution contains certain code that is not part
of libffi, and is only used as tooling to assist with the building and
testing of libffi. This includes the msvcc.sh script used to wrap the
Microsoft compiler with GNU compatible command-line options,
make_sunver.pl, and the libffi test code distributed in the
testsuite/libffi.bhaible directory. This code is distributed with libffi for the purpose of convenience only, and libffi is in no way derived from this code.

msvcc.sh an testsuite/libffi.bhaible are both distributed under the terms of the GNU GPL version 2, as below.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:
Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

libffi - Copyright (c) 1996-2020 Anthony Green, Red Hat, Inc and others. See source files for details.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Commons BeanUtils
Copyright 2000-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

===================================================================

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from 'org.apache.velocity' to 'clover.org.apache.velocity'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

===================================================================
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS.
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. 
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software 
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.

/*
$Id: LICENSE.txt,v 1.1.1.1 2004/07/01 13:59:13 jvanzy Exp$

Copyright 2002 (C) The Codehaus. All Rights Reserved.

Redistribution and use of this software and associated documentation 
("Software"), with or without modification, are permitted provided 
that the following conditions are met:

1. Redistributions of source code must retain copyright 
   statements and notices. Redistributions must also contain a 
   copy of this document.

2. Redistributions in binary form must reproduce the 
   above copyright notice, this list of conditions and the 
   following disclaimer in the documentation and/or other 
   materials provided with the distribution.

3. The name "classworlds" must not be used to endorse or promote 
   products derived from this Software without prior written 
   permission of The Codehaus. For written permission, please 
   contact bob@codehaus.org.

4. Products derived from this Software may not be called "classworlds" 
   nor may "classworlds" appear in their names without prior written 
   permission of The Codehaus. "classworlds" is a registered 
   trademark of The Codehaus.

5. Due credit should be given to The Codehaus. 
   (http://classworlds.codehaus.org/).

THIS SOFTWARE IS PROVIDED BY THE CODEHAUS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE CODEHAUS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/
=================================================================================================
==============================================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from the 'antlr.*' to the 'clover.antlr.*'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

=================================================================================================
==============================================
SOFTWARE RIGHTS

ANTLR 1989-2006 Developed by Terence Parr
Partially supported by University of San Francisco & jGuru.com

We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commerical software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

The primary ANTLR guy:

Terence Parr
The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'com.google.common' to 'clover.com.google.common'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

==================================================================================
==========================
Modified by Atlassian
The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'com.google.json' to 'clover.com.google.json'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

==================================================================================
==========================
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed...
as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

==================================================================================

Modified by Atlassian
The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'com.lowagie' to 'clover.com.lowagie'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

==================================================================================
==========================
MOZILLA PUBLIC LICENSE
Version 1.1

-----------
I. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous
Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or
as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims
infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
(a) Third Party Claims.
If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the
party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.
If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.
Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the
obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.
6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLA PL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom
You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN
INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.


11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.
Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A - Mozilla Public License.

```
The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is ______________________________________.

The Initial Developer of the Original Code is ______________________.
Portions created by ______________________ are Copyright (C) ______ _______. All Rights Reserved.

Contributor(s): ______________________________________.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [_____ ] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

Apache Velocity

Copyright (C) 2000-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.

4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.

5. Due credit should be given to the DOM4J Project - http://www.dom4j.org

THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*--
$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp $
Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
derived from this software without prior written permission. For
written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:

"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/
From: http://www.json.org/license.html
==================================================================================
======================================
Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SPDX-License-Identifier: Apache-2.0 OR LGPL-2.1

Java Native Access (JNA) is licensed under the LGPL, version 2.1 or later, or (from version 4.0 onward) the Apache License, version 2.0.

You can freely decide which license you want to apply to the project.

You may obtain a copy of the LGPL License at:

http://www.gnu.org/licenses/licenses.html

A copy is also included in the downloadable source code package containing JNA, in file "LGPL2.1", under the same directory as this file.

You may obtain a copy of the Apache License at:

http://www.apache.org/licenses/
A copy is also included in the downloadable source code package containing JNA, in file "AL2.0", under the same directory as this file.

Commercial support may be available, please e-mail twall[at]users[dot]sf[dot]net.

==================================================================================

                             Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'org.apache.commons' to 'clover.org.apache.commons'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

==================================================================================

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Open Source Used In DNAC Platform Antelope 6095
Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2005 - 2009 Taras Puchko
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

********************************************************************************
* CruiseControl, a Continuous Integration Toolkit
* Copyright (c) 2001-2003, ThoughtWorks, Inc.
* 651 W Washington Ave. Suite 500
* Chicago, IL 60661 USA
* All rights reserved.
* * Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* * + Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* * + Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following
* disclaimer in the documentation and/or other materials provided
* with the distribution.
* * + Neither the name of ThoughtWorks, Inc., CruiseControl, nor the
* names of its contributors may be used to endorse or promote
* products derived from this software without specific prior
* written permission.
* * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
********************************************************************************/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright
the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.
The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact...
all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above,

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Library General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

/*--

$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp$

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

/*
Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana Univeristy" and "Indiana Univeristy Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact http://www.extreme.indiana.edu/.

5. Products derived from this software may not use "Indiana Univeristy"
name nor may "Indiana Univeristy" appear in their name, without prior
written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==================================================================================
==========================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from the 'org.apache.commons' to the 'clover.org.apache.commons'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.

==================================================================================
==========================
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*
File: Core.js

Description:

Provides common utility functions and the Class object used internally by the library.

Also provides the <TreeUtil> object for manipulating JSON tree structures


Author:
Nicolas Garcia Belmonte

Copyright:

Copyright 2008-2009 by Nicolas Garcia Belmonte.

Homepage:

<http://thejit.org>

Version:

1.1.2

License:

BSD License

> Redistribution and use in source and binary forms, with or without
> modification, are permitted provided that the following conditions are met:
> * Redistributions of source code must retain the above copyright
>   notice, this list of conditions and the following disclaimer.
> * Redistributions in binary form must reproduce the above copyright
>   notice, this list of conditions and the following disclaimer in the
>   documentation and/or other materials provided with the distribution.
> * Neither the name of the organization nor the
>   names of its contributors may be used to endorse or promote products
>   derived from this software without specific prior written permission.
> *
> * THIS SOFTWARE IS PROVIDED BY Nicolas Garcia Belmonte ``AS IS'' AND ANY
>   EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
>   WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
>   DISCLAIMED. IN NO EVENT SHALL Nicolas Garcia Belmonte BE LIABLE FOR ANY
>   DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
>   (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
>   LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
>   ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
>   (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
>   SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.1765 pcapngdecoder 1.2

1.1765.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * The MIT License (MIT)
 * <p/>
 * Copyright (c) 2015-2016 Bertrand Martel
 * <p/>
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the Software is
 * furnished to do so, subject to the following conditions:
 * <p/>
 * The above copyright notice and this permission notice shall be included in
 * all copies or substantial portions of the Software.
 * <p/>
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
 * THE SOFTWARE.
 */

Found in path(s):
* /opt/cola/permits/1202446644_1631567950.01/0/pcapngdecoder-1-2-sources-jar/fr/bmartel/pcapdecoder/structure/options/impl/OptionsEnhancedHeader.java
* /opt/cola/permits/1202446644_1631567950.01/0/pcapngdecoder-1-2-sources-jar/fr/bmartel/pcapdecoder/structure/types/impl/NameResolutionHeader.java
* /opt/cola/permits/1202446644_1631567950.01/0/pcapngdecoder-1-2-sources-jar/fr/bmartel/pcapdecoder/structure/options/OptionParser.java
* /opt/cola/permits/1202446644_1631567950.01/0/pcapngdecoder-1-2-sources-jar/fr/bmartel/pcapdecoder/structure/options/inter/IOptionsEnhancedPacketHeader.java
* /opt/cola/permits/1202446644_1631567950.01/0/pcapngdecoder-1-2-sources-jar/fr/bmartel/pcapdecoder/structure/options/inter/IOptionsDescriptionHeader.java
* /opt/cola/permits/1202446644_1631567950.01/0/pcapngdecoder-1-2-sources-jar/fr/bmartel/pcapdecoder/network/DnsEntryObject.java
* /opt/cola/permits/1202446644_1631567950.01/0/pcapngdecoder-1-2-sources-
1.1766 expand-range 1.8.2
1.1766.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1767 babel-plugin-syntax-class-properties
6.13.0
1.1767.1 Available under license:
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.
MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1768 okio 1.13.0

1.1769 apr 1.6.5
1.1770 istanbul-lib-report 3.0.0

1.1770.1 Available under license:

MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1771 mail 1.4.7

1.1771.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or
otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the
Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients' rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it
clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributos under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT
8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software, (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation, as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.
If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY
TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERNED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at
This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.1772 pirates 4.0.1
1.1772.1 Available under license :
MIT License

Copyright (c) 2016-2018 Ari Porad

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1773 libxi 1.7.6-1
1.1773.1 Available under license:

Copyright 1989, 1998 The Open Group

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

Copyright 1989 by Hewlett-Packard Company, Palo Alto, California.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

HEWLETT-PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
EVENT SHALL HEWLETT-PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2008 Peter Hutterer

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the author shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the author.

Copyright 2009 Red Hat, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1774 mute-stream 0.0.7

1.1774.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1775 commons-lang3 3.8.1

1.1775.1 Available under license:

Apache Commons Lang
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1776 calcite-example-function 1.21.0
1.1776.1 Available under license :

Calcite Example Function
Copyright 2012-2019 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a ”NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the Open Source Used In DNAC Platform Antelope 6153
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1777 json-simple 1.1.1

1.1778 amqp-client 3.6.6

1.1778.1 Available under license:

Found license 'General Public License 2.0' in // Copyright (c) 2007-Present Pivotal Software, Inc. All rights reserved. // This software, the RabbitMQ Java client library, is triple-licensed under the // Mozilla Public License 1.1 ("MPL"), the GNU General Public License version 2 ("GPL") and the Apache License version 2 ("ASL"). For the MPL, please see Copyright (c) 2007-2016 Pivotal Software, Inc. All Rights Reserved. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 distributed under the License is distributed on an "AS IS" BASIS, * Based on org.stringtree.json.JSONWriter, licensed under APL and

Found license 'General Public License 2.0' in // Copyright (c) 2007-Present Pivotal Software, Inc. All rights reserved. // This software, the RabbitMQ Java client library, is triple-licensed under the // Mozilla Public License 1.1 ("MPL"), the GNU General Public License version 2 ("GPL") and the Apache License version 2 ("ASL"). For the MPL, please see Copyright (c) 2007-2016 Pivotal Software, Inc. All Rights Reserved. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 distributed under the License is distributed on an "AS IS" BASIS, * Based on org.stringtree.json.JSONReader, licensed under APL and

Found license 'General Public License 2.0' in // Copyright (c) 2007-Present Pivotal Software, Inc. All rights reserved. // This software, the RabbitMQ Java client library, is triple-licensed under the // Mozilla Public License 1.1 ("MPL"), the GNU General Public License version 2 ("GPL") and the Apache License version 2 ("ASL"). For the MPL, please see * Expert Group and released to the public domain, as explained at

Found license 'General Public License 2.0' in // Copyright (c) 2007-Present Pivotal Software, Inc. All rights reserved. // This software, the RabbitMQ Java client library, is triple-licensed under the // Mozilla Public License 1.1 ("MPL"), the GNU General Public License version 2 ("GPL") and the Apache License version 2 ("ASL"). For the MPL, please see final static String LICENSE="Licensed under the MPL. See http://www.rabbitmq.com/";

Found license 'General Public License 2.0' in // Copyright (c) 2007-Present Pivotal Software, Inc. All rights reserved. // This software, the RabbitMQ Java client library, is triple-licensed under the // Mozilla Public License 1.1 ("MPL"), the GNU General Public License version 2 ("GPL") and the Apache License version 2 ("ASL"). For the MPL, please see
1.1779 fast-json-stable-stringify 2.1.0

1.1779.1 Available under license:

This software is released under the MIT license:

Copyright (c) 2017 Evgeny Poberezkin
Copyright (c) 2013 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1780 babel-plugin-transform-es2015-block-scoped-functions 6.22.0 6.22.0

1.1780.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.1781 gson 2.8.6

1.1781.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2010 The Android Open Source Project
 * Copyright (C) 2012 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
  jar/com/google/gson/internal/LinkedTreeMap.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
  jar/com/google/gson/internal/LinkedHashTreeMap.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 The Gson authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
*/
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2017 The Gson authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/CollectionTypeAdapterFactory.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/TypeAdapters.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/JsonTreeReader.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/ArrayTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/ObjectTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/MapTypeAdapterFactory.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/TypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/JsonReaderInternalAccess.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/JsonTreeWriter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/TypeAdapterFactory.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/ReflectiveTypeAdapterFactory.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
jar/com/google/gson/internal/bind/TypeAdapterRuntimeTypeWrapper.java
No license file was found, but licenses were detected in source scan.

/
* Copyright (C) 2008 Google Inc.
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

No license file was found, but licenses were detected in source scan.

*/

Copyright (C) 2008 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

No license file was found, but licenses were detected in source scan.

*/

Copyright (C) 2010 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/JsonToken.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/JsonWriter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/JsonScope.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/MalformedJsonException.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/stream/JsonReader.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Google Inc.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/LazilyParsedNumber.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/SqlDateTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/TimeTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/ConstructorConstructor.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/SqlDateTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/TimeTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/TimeTypeAdapter.java
jar/com/google/gson/internal/bind/DateTypeAdapter.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/UnsafeAllocator.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/bind/TreeTypeAdapter.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2010 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonSyntaxException.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/UnsafeAllocator.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/internal/Streams.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/JsonSyntaxException.java
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/Internal/Streams.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 Google Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1108597571_1606176527.27/0/gson-2-8-6-sources-1-jar/com/google/gson/LongSerializationPolicy.java
1.1782 passport 0.4.0
1.1782.1 Available under license:

The MIT License (MIT)

Copyright (c) 2011-2015 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1783 debug 4.3.1
1.1783.1 Available under license:

(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
1.1784 activation-api 1.2.1

1.1784.1 Available under license:

## Notices for Eclipse Project for JAF

This content is produced and maintained by the Eclipse Project for JAF project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaf

### Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

### Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0, which is available at http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

### Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaf

### Third-party Content

This project leverages the following third party content.

JUnit (4.12)

* License: Eclipse Public License

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1785 t-digest 3.2

1.1786 levn 0.3.0

1.1786.1 Available under license:

Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1787 es6-promise 3.2.1
1.1787.1 Available under license :
Copyright (c) 2014 Yehuda Katz, Tom Dale, Stefan Penner and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1788 jline-terminal 3.12.1
1.1788.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2017, the original author or authors.
 * This software is distributable under the BSD license. See the terms of the BSD license in the documentation provided with this software.
 * https://opensource.org/licenses/BSD-3-Clause
 */
/*

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/NonBlockingPumpInputStream.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/terminal/impl/AbstractWindowsConsoleWriter.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/WriterOutputStream.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/NonBlockingPumpReader.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/PumpReader.java

No license file was found, but licenses were detected in source scan.

#
# Copyright (c) 2002-2016, the original author or authors.
#
# This software is distributable under the BSD license. See the terms of the
# BSD license in the documentation provided with this software.
#
# https://opensource.org/licenses/BSD-3-Clause
#

auto_left_margin, bw, bw
auto_right_margin, am, am
back_color_erase, bce, ut
can_change, ccc, cc
ceol_standout_glitch, xhp, xs
col_addr_glitch, xhpa, YA
cpi_changes_res, cpix, YF
cr_cancels_micro_mode, crxm, YB
dest_tabs_magic_smso, xt, xt
eat_newline_glitch, xenl, xn
erase_overstrike, eo, eo
generic_type, gn, gn
hard_copy, hc, hc
hard_cursor, chts, HC
has_meta_key, km, km
has_print_wheel, daisy, YC
has_status_line, hs, hs
hue_lightness_saturation, hls, hl
insert_null_glitch, in, in
lpi_changes_res, lpix, YG
memory_above, da, da
memory_below, db, db
move_insert_mode, mir, mi
move_standout_mode, msgr, ms
needs_xon_xoff, nxon, nx
no_esc_ctlc, xsb, xb
no_pad_char, npc, NP
non_dest_scroll_region, ndscr, ND
non_rev_rmcup, nrmc, NR
over_strike, os, os
prtr_silent, mc5i, 5i
row_addr_glitch, xvpa, YD
semi_auto_right_margin, sam, YE
status_line_esc_ok, eslok, es
tilde_glitch, hz, hz
transparent_underline, ul, ul
xon_xoff, xon, xo
columns, cols, co
init_tabs, it, it
label_height, lh, lh
label_width, lw, lw
lines, lines, li
lines_of_memory, lm, lm
magic_cookie_glitch, xmc, sg
max_attributes, ma, ma
max_colors, colors, Co
max_pairs, pairs, pa
maximum_windows, wnum, MW
no_color_video, ncv, NC
num_labels, nlab, Nl
padding_baud_rate, pb, pb
virtual_terminal, vt, vt
width_status_line, wsl, ws
bit_image_entwining, bitwin, Yo
bit_image_type, bitype, Yp
buffer_capacity, bufsz, Ya
buttons, btns, BT
dot_horz_spacing, spinh, Yc
dot_vert_spacing, spinv, Yb
max_micro_address, maddr, Yd
max_micro_jump, mjump, Ye
micro_col_size, mcs, Yf
micro_line_size, mls, Yg
number_of_pins, npins, Yh
output_res_char, orc, Yi
output_res_horz_inch, orhi, Yk
output_res_line, orl, Yj
output_res_vert_inch, orvi, Yl
print_rate, cps, Ym
wide_char_size, widcs, Yn
acs_chars, acsc, ac
back_tab, cbt, bt
bell, bel, bl
carriage_return, cr, cr
change_char_pitch, cpi, ZA
change_line_pitch, lpi, ZB
change_res_horz, chr, ZC
change_res_vert, cvr, ZD
change_scroll_region, csr, cs
char_padding, rmp, rP
clear_all_tabs, tbc, ct
clear_margins, mge, MC
clear_screen, clear, cl
clr_bol, el1, cb
clr_eol, el, ce
clr_eos, ed, cd
column_address, hpa, ch
command_character, cmdch, CC
create_window, cwin, CW
cursor_address, cup, cm
cursor_down, cud1, do
cursor_home, home, ho
cursor_invisible, civis, vi
cursor_left, cub1, le
cursor_mem_address, mrcup, CM
cursor_normal, cnorm, ve
cursor_right, cuf1, nd
cursor_to_ll, ll, ll
cursor_up, cuu1, up
cursor_visible, cvvis, vs
define_char, defc, ZE
delete_character, dch1, dc
delete_line, dl1, dl
dial_phone, dial, DI
dis_status_line, dsl, ds
display_clock, dclk, DK
down_half_line, hd, hd
ena_acs, enacs, eA
enter_alt_charset_mode, smacs, as
enter_am_mode, smam, SA
enter_blink_mode, blink, mb
enter_bold_mode, bold, md
enter_ca_mode, smcup, ti
enter_delete_mode, smdc, dm
enter_dim_mode, dimc, mh
enter_doublewide_mode, swidm, ZF
enter_draft_quality, sdrfq, ZG
enter_insert_mode, snmir, im
enter_italics_mode, sitm, ZH
enter_leftward_mode, slm, ZI
key_f3, kf3, k3
key_f30, kf30, FK
key_f31, kf31, FL
key_f32, kf32, FM
key_f33, kf33, FN
key_f34, kf34, FO
key_f35, kf35, FP
key_f36, kf36, FQ
key_f37, kf37, FR
key_f38, kf38, FS
key_f39, kf39, FT
key_f4, kf4, k4
key_f40, kf40, FU
key_f41, kf41, FV
key_f42, kf42, FW
key_f43, kf43, FX
key_f44, kf44, FY
key_f45, kf45, FZ
key_f46, kf46, Fa
key_f47, kf47, Fb
key_f48, kf48, Fc
key_f49, kf49, Fd
key_f5, kf5, k5
key_f50, kf50, Fe
key_f51, kf51, Ff
key_f52, kf52, Fg
key_f53, kf53, Fh
key_f54, kf54, Fi
key_f55, kf55, Fj
key_f56, kf56, Fk
key_f57, kf57, Fl
key_f58, kf58, Fm
key_f59, kf59, Fn
key_f6, kf6, k6
key_f60, kf60, Fo
key_f61, kf61, Fp
key_f62, kf62, Fq
key_f63, kf63, Fr
key_f7, kf7, k7
key_f8, kf8, k8
key_f9, kf9, k9
key_find, kfnd, @0
key_help, khlp, %1
key_home, khome, kh
key_ic, kich1, kI
key_il, kil1, kA
key_left, kcub1, kl
key_ll, kll, kH
key_stab, khts, KT
key_sundo, kUND, !3
key_suspend, kspd, &7
key_undo, kund, &8
key_up, kcuu1, ku
keypad_local, rmkx, ke
keypad_xmit, smkx, ks
lab_f0, lf0, 10
lab_f1, lf1, 11
lab_f10, lf10, la
lab_f2, lf2, 12
lab_f3, lf3, 13
lab_f4, lf4, 14
lab_f5, lf5, 15
lab_f6, lf6, 16
lab_f7, lf7, 17
lab_f8, lf8, 18
lab_f9, lf9, 19
label_format, fln, Lf
label_off, rmln, LF
label_on, smln, LO
meta_off, rmm, mo
meta_on, smm, mm
micro_column_address, mhpa, ZY
micro_down, mcud1, ZZ
micro_left, mcub1, Za
micro_right, mcuf1, Zb
micro_row_address, mvpa, Zc
micro_up, mcuu1, Zd
newline, nel, nw
order_of_pins, porder, Ze
orig_colors, oc, oc
orig_pair, op, op
pad_char, pad, pc
parm_dch, dch, DC
parm_delete_line, dl, DL
parm_down_cursor, cud, DO
parm_down_micro, mcud, Zf
parm_ich, ich, IC
parm_index, indn, SF
parm_insert_line, il, AL
parm_left_cursor, cub, LE
parm_left_micro, mcub, Zg
parm_right_cursor, cuf, RI
parm_right_micro, mcuf, Zh
parm_rindex, rin, SR
parm_up_cursor, cuu, UP
parm_up_micro, mcuu, Zi
pkey_key, pkey, pk
pkey_local, pfloc, pl
pkey_xmit, px, px
plab_norm, pln, pn
print_screen, mc0, ps
prtr_non, mc5p, pO
prtr_off, mc4, pf
prtr_on, mc5, po
pulse, pulse, PU
quick_dial, qdial, QD
remove_clock, rmclk, RC
repeat_char, rep, rp
req_for_input, rfi, RF
reset_1string, rs1, r1
reset_2string, rs2, r2
reset_3string, rs3, r3
reset_file, rf, rf
restore_cursor, rc, rc
row_address, vpa, cv
save_cursor, sc, sc
scroll_forward, ind, sf
scroll_reverse, ri, sr
select_char_set, scs, Zj
set_attributes, sgr, sa
set_background, setb, Sb
set_bottom_margin, smgb, Zk
set_bottom_margin_parm, smgbp, Zl
set_clock, sclk, SC
set_color_pair, scp, sp
set_foreground, setf, Sf
set_left_margin, smgl, ML
set_left_margin_parm, smglp, Zm
set_right_margin, smgr, MR
set_right_margin_parm, smgrp, Zn
set_tab, hts, st
set_top_margin, smgt, Zo
set_top_margin_parm, smgtp, Zp
set_window, wind, wi
start_bit_image, sbim, Zq
start_char_set_def, scsd, Zr
stop_bit_image, rbim, Zs
stop_char_set_def, rcsd, Zt
subscript_characters, subcs, Zu
superscript_characters, supcs, Zv
tab, ht, ta
these_cause_cr, docr, Zw
to_status_line, tsl, ts
tone, tone, TO
underline_char, uc, uc
down_half_line, hu, hu
user0, u0, 0
user1, u1, 1
user2, u2, 2
user3, u3, 3
user4, u4, 4
user5, u5, 5
user6, u6, 6
user7, u7, 7
user8, u8, 8
user9, u9, 9
wait_tone, wait, WA
xon_character, xonc, XN
zero_motion, zerom, Zx
alt_scancode_esc, scesa, S8
bit_image_carriage_return, bicr, Yv
bit_image_newline, binel, Zz
bit_image_repeat, birep, Xy
char_set_names, csnm, Zy
code_set_init, csin, ci
color_names, colornm, Yw
define_bit_image_region, defbi, Yx
device_type, devt, dv
display_pc_char, dispcc, S1
end_bit_image_region, endbi, Yy
enter_pc_charset_mode, smpch, S2
enter_scancode_mode, smsc, S4
exit_pc_charset_mode, rmpch, S3
exit_scancode_mode, rmsec, S5
get_mouse, getm, Gm
key_mouse, kmous, Km
mouse_info, minfo, Mi
pc_term_options, pctrm, S6
pkey_plab, pxl, xl
req_mouse_pos, reqm, RQ
scancode_escape, scesc, S7
set0_des_seq, st0ds, 0
set1_des_seq, st1ds, 1
set2_des_seq, st2ds, 2
set3_des_seq, st3ds, 3
set_a_background, setab, AB
set_a_foreground, setaf, AF
set_color_band, setcolor, Yz
set_lr_margin, smgll, ML
set_page_length, slines, YZ
set_tb_margin, smgtb, MT
enter_horizontal_hl_mode, ehhlm, Xh
enter_left_hl_mode, ehhlm, XI
enter_low_hl_mode, elo hl m, Xo
enter_right_hl_mode, erhlm, Xr
enter_top_hl_mode, ethlm, Xt
enter_vertical_hl_mode, evhl m, Xv
set_a_attributes, sgr1, sA
set_pgl_en Inch, slength, sL

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/utils/capabilities.txt
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2016, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/terminal/impl/ExecPty.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/terminal/impl/MouseSupport.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/terminal/Attributes.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/terminal/Cursor.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/terminal/impl/CursorSupport.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/terminal/impl/AbstractPosixTerminal.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/terminal/impl/NativeSignalHandler.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/utils/package-info.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/utils/DiffHelper.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/utils/InputStreamReader.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-
dar/jline/terminal/MouseEvent.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/WCWidth.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/ClosedException.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/Levenshtein.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/ExecHelper.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/OSUtils.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/ShutdownHooks.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/terminal/spi/Pty.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/terminal/impl/package-info.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/Signals.java
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/AttributedString.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009-2018 the original author(s).
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1057262552_1592255590.19/0/jline-terminal-3-12-1-sources-jar/org/jline/utils/AnsiWriter.java

No license file was found, but licenses were detected in source scan.

#
# Copyright (c) 2002-2018, the original author or authors.
#
# This software is distributable under the BSD license. See the terms of the
# BSD license in the documentation provided with this software.
#
# https://opensource.org/licenses/BSD-3-Clause
#
black
maroon
green
olive
navy
purple
teal
silver
grey
red
lime
yellow
blue
fuchsia
aqua
white
grey0
navyblue
darkblue
blue3
blue3a
blue1
darkgreen
deepskyblue4
deepskyblue4a
deepskyblue4b
dodgerblue3
dodgerblue2
green4
springgreen4
turquoise4
deepskyblue3
deepskyblue3a
dodgerblue1
green3
springgreen3
darkcyan
lightseagreen
deepskyblue2
deepskyblue1
green3a
springgreen3a
springgreen2
cyan3
mediumpurple2a
mediumpurple1
gold3
darkkhaki
navajowhite3
grey69
lightsteelblue3
lightsteelblue
eyellow3
darkolivegreen3b
darkseagreen3a
darkseagreen2
lightcyan3
lightskyblue1
greenyellow
darkolivegreen2
palegreen1a
darkseagreen2a
darkseagreen1
paleturquoise1
red3a
deeppink3
deeppink3a
magenta3a
magenta3b
magenta2
darkorange3a
indianreda
hotpink3a
hotpink2
orchid
mediumorchid1
orange3
lightsalmon3a
lightpink3
pink3
plum3
violet
gold3a
lightgoldenrod3
tan
mistyrose3
thistle3
plum2
yellow3a
khaki3
lightgoldenrod2
lightyellow3
No license file was found, but licenses were detected in source scan.

2016, the original author or authors.

This software is distributable under the BSD license. See the terms of the BSD license in the documentation provided with this software.

https://opensource.org/licenses/BSD

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2018, the original author or authors.
 * This software is distributable under the BSD license. See the terms of the BSD license in the documentation provided with this software.
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2019, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 */
1.1789 jetty-util 9.4.17.v20190418

1.1789.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates'
from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright
3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial
product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the
parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1790 log4j-api 2.8.2

1.1790.1 Available under license:

Apache Log4j 1.x Compatibility API
Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1791 de.huxhorn.sulky.io 0.9.17

1.1791.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from
the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.
You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this
c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.1792 ee-first 1.1.1
1.1792.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1793.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1794 eslint-plugin-import 2.22.1
1.1794.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Ben Mosher

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1795 lodash-merge 4.6.2

1.1795.1 Available under license:

Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====
Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1796 libx11 1.6.3-1ubuntu2.1

1.1796.1 Available under license:

The following is the 'standard copyright' agreed upon by most contributors, and is currently the canonical license preferred by the X.Org Foundation. This is a slight variant of the common MIT license form published by the Open Source Initiative at http://www.opensource.org/licenses/mit-license.php

Copyright holders of new code should use this license statement where possible, and insert their name to this list. Please sort by surname for people, and by the full name for other entities (e.g. Juliusz Chroboczek sorts before Intel Corporation sorts before Daniel Stone).

See each individual source file or directory for the license that applies to that file.

Copyright (C) 2003-2006,2008 Jamey Sharp, Josh Triplett
Copyright 2009 Red Hat, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

The following licenses are 'legacy' - usually MIT/X11 licenses with the name
of the copyright holder(s) in the license statement:

Copyright 1984-1994, 1998 The Open Group

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation.

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be
used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization from The Open Group.

X Window System is a trademark of The Open Group.

Copyright 2000 The XFree86 Project, Inc.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.


 Portions Copyright 1990, 1991 by Tektronix, Inc.

Permission to use, copy, modify and distribute this documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in all copies, and that the names of Digital and Tektronix not be used in in advertising or publicity pertaining to this documentation without specific, written prior permission. Digital and Tektronix makes no representations about the suitability of this documentation for any purpose. It is provided "as is" without express or implied warranty.

----------------------------------------

Copyright (c) 1999-2000 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
FREE SOFTWARE FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the Free Software Foundation shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the Free Software Foundation.

----------------------------------------

Code and supporting documentation (c) Copyright 1990 1991 Tektronix, Inc. All Rights Reserved

This file is a component of an X Window System-specific implementation of Xcms based on the TekColor Color Management System. TekColor is a trademark of Tektronix, Inc. The term "TekHVC" designates a particular color space that is the subject of U.S. Patent No. 4,985,853 (equivalent foreign patents pending). Permission is hereby granted to use, copy, modify, sell, and otherwise distribute this software and its documentation for any purpose and without fee, provided that:

1. This copyright, permission, and disclaimer notice is reproduced in all copies of this software and any modification thereof and in supporting documentation;

2. Any color-handling application which displays TekHVC color coordinates identifies these as TekHVC color coordinates in any interface that displays these coordinates and in any associated documentation;

3. The term "TekHVC" is always used, and is only used, in association with the mathematical derivations of the TekHVC Color Space, including those provided in this file and any equivalent pathways and mathematical derivations, regardless of digital (e.g., floating point or integer) representation.

Tektronix makes no representation about the suitability of this software for any purpose. It is provided "as is" and with all faults.

TEKTRONIX DISCLAIMS ALL WARRANTIES APPLICABLE TO THIS SOFTWARE, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL TEKTRONIX BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA, OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE, OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR THE PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
(c) Copyright 1995 FUJITSU LIMITED
This is source code modified by FUJITSU LIMITED under the Joint Development Agreement for the CDE/Motif PST.

----------------------------------------
Copyright 1992 by Oki Technosystems Laboratory, Inc.
Copyright 1992 by Fuji Xerox Co., Ltd.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Oki Technosystems Laboratory and Fuji Xerox not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
Oki Technosystems Laboratory and Fuji Xerox make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

OKI TECHNOSYSTEMS LABORATORY AND FUJI XEROX DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OKI TECHNOSYSTEMS LABORATORY AND FUJI XEROX BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FUJITSU LIMITED not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
FUJITSU LIMITED makes no representations about the suitability of this software for any purpose.
It is provided "as is" without express or implied warranty.

FUJITSU LIMITED DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL FUJITSU LIMITED BE LIABLE FOR ANY SPECIAL, INDIRECT OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright (c) 1995 David E. Wexelblat. All rights reserved

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL DAVID E. WEXELBLAT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of David E. Wexelblat shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from David E. Wexelblat.

----------------------------------------

Copyright 1990, 1991 by OMRON Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name OMRON not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. OMRON makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.
OMRON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OMRON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------


Portions Copyright 1990, 1991 by Tektronix, Inc

Rewritten for X.org by Chris Lee <clee@freedesktop.org>

Permission to use, copy, modify, distribute, and sell this documentation for any purpose and without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

Chris Lee makes no representations about the suitability for any purpose of the information in this document. It is provided "as-is" without express or implied warranty.

----------------------------------------

Copyright 1993 by Digital Equipment Corporation, Maynard, Massachusetts,
Copyright 1994 by FUJITSU LIMITED
Copyright 1994 by Sony Corporation

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Digital, FUJITSU LIMITED and Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL, FUJITSU LIMITED AND SONY CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL, FUJITSU LIMITED AND SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1991 by the Open Software Foundation

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the name of Open Software Foundation
not be used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission. Open Software
Foundation makes no representations about the suitability of this
software for any purpose. It is provided "as is" without express or
implied warranty.

OPEN SOFTWARE FOUNDATION DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL OPEN SOFTWARE FOUNDATION BE
LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1993, 1994 by Sony Corporation

Permission to use, copy, modify, distribute, and sell this software and
its documentation for any purpose is hereby granted without fee, provided
that the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the name of FUJITSU LIMITED and Sony Corporation
not be used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission. FUJITSU LIMITED and
Sony Corporation makes no representations about the suitability of this
software for any purpose. It is provided "as is" without express or
implied warranty.

FUJITSU LIMITED AND SONY CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD
TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL FUJITSU LIMITED OR SONY CORPORATION BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE
ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1992, 1993 by FUJITSU LIMITED
Copyright 1993 by Fujitsu Open Systems Solutions, Inc.
Copyright 1994 by Sony Corporation

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FUJITSU LIMITED, Fujitsu Open Systems Solutions, Inc. and Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
FUJITSU LIMITED, Fujitsu Open Systems Solutions, Inc. and Sony Corporation make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

FUJITSU LIMITED, FUJITSU OPEN SYSTEMS SOLUTIONS, INC. AND SONY CORPORATION DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL FUJITSU OPEN SYSTEMS SOLUTIONS, INC., FUJITSU LIMITED AND SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1987, 1988, 1990, 1993 by Digital Equipment Corporation, Maynard, Massachusetts,

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Digital not be used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1993 by SunSoft, Inc.
Copyright 1999-2000 by Bruno Haible

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of SunSoft, Inc. and Bruno Haible not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. SunSoft, Inc. and Bruno Haible make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

SunSoft Inc. AND Bruno Haible DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SunSoft, Inc. OR Bruno Haible BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1991 by the Open Software Foundation
Copyright 1993 by the TOSHIBA Corp.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Open Software Foundation and TOSHIBA not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Open Software Foundation and TOSHIBA make no representations about the suitability of this software for any purpose. It is provided "as is" without express or
implied warranty.

OPEN SOFTWARE FOUNDATION AND TOSHIBA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OPEN SOFTWARE FOUNDATION OR TOSHIBA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1988 by Wyse Technology, Inc., San Jose, Ca.,
All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name Wyse not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

WYSE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1991 by the Open Software Foundation
Copyright 1993, 1994 by the Sony Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Open Software Foundation and Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Open Software Foundation and Sony Corporation make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.
OPEN SOURCE FOUNDATION AND SONY CORPORATION DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL OPEN SOFTWARE FOUNDATION OR SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1992, 1993 by FUJITSU LIMITED
Copyright 1993 by Fujitsu Open Systems Solutions, Inc.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FUJITSU LIMITED and Fujitsu Open Systems Solutions, Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
FUJITSU LIMITED and Fujitsu Open Systems Solutions, Inc. makes no representations about the suitability of this software for any purpose.
It is provided "as is" without express or implied warranty.

FUJITSU LIMITED AND FUJITSU OPEN SYSTEMS SOLUTIONS, INC. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL FUJITSU OPEN SYSTEMS SOLUTIONS, INC. AND FUJITSU LIMITED BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1993, 1994 by Sony Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
Sony Corporation makes no representations about the suitability of
this software for any purpose. It is provided "as is" without express or implied warranty.

SONY CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 1986, 1998 The Open Group
Copyright (c) 2000 The XFree86 Project, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM OR THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium or of the XFree86 Project shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium and the XFree86 Project.

----------------------------------------
Copyright 1990, 1991 by OMRON Corporation, NTT Software Corporation, and Nippon Telegraph and Telephone Corporation
Copyright 1991 by the Open Software Foundation
Copyright 1993 by the FUJITSU LIMITED

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the names of OMRON, NTT Software, NTT, and
Open Software Foundation not be used in advertising or publicity
pertaining to distribution of the software without specific,
written prior permission. OMRON, NTT Software, NTT, and Open Software
Foundation make no representations about the suitability of this
software for any purpose. It is provided "as is" without express or
implied warranty.

OMRON, NTT SOFTWARE, NTT, AND OPEN SOFTWARE FOUNDATION
DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT
SHALL OMRON, NTT SOFTWARE, NTT, OR OPEN SOFTWARE FOUNDATION BE
LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 1988 by Wyse Technology, Inc., San Jose, Ca,
Copyright 1987 by Digital Equipment Corporation, Maynard, Massachusetts,

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name Digital not be
used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

DIGITAL AND WYSE DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO
EVENT SHALL DIGITAL OR WYSE BE LIABLE FOR ANY SPECIAL, INDIRECT OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF
USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 1991, 1992 by Fuji Xerox Co., Ltd.
Copyright 1992, 1993, 1994 by FUJITSU LIMITED

Permission to use, copy, modify, distribute, and sell this software
and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Fuji Xerox, FUJITSU LIMITED not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Fuji Xerox, FUJITSU LIMITED make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

FUJI XEROX, FUJITSU LIMITED DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL FUJI XEROX, FUJITSU LIMITED BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 2006 Josh Triplett

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

----------------------------------------
(c) Copyright 1996 by Sebastien Marineau and Holger Veit
<marineau@genie.uottawa.ca>
    <Holger.Veit@gmd.de>
Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
HOLGER VEIT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Except as contained in this notice, the name of Sebastien Marineau or Holger Veit
shall not be used in advertising or otherwise to promote the sale, use or other
dealings in this Software without prior written authorization from Holger Veit or
Sebastien Marineau.

------------------------------

Copyright 1990, 1991 by OMRON Corporation, NTT Software Corporation,
and Nippon Telegraph and Telephone Corporation
Copyright 1991 by the Open Software Foundation
Copyright 1993 by the TOSHIBA Corp.
Copyright 1993, 1994 by Sony Corporation
Copyright 1993, 1994 by the FUJITSU LIMITED

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the names of OMRON, NTT Software, NTT, Open
Software Foundation, and Sony Corporation not be used in advertising
or publicity pertaining to distribution of the software without specific,
written prior permission. OMRON, NTT Software, NTT, Open Software
Foundation, and Sony Corporation make no representations about the
suitability of this software for any purpose. It is provided "as is"
without express or implied warranty.

OMRON, NTT SOFTWARE, NTT, OPEN SOFTWARE FOUNDATION, AND SONY
CORPORATION DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT
SHALL OMRON, NTT SOFTWARE, NTT, OPEN SOFTWARE FOUNDATION, OR SONY
Copyright 2000 by Bruno Haible

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Bruno Haible not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Bruno Haible makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Bruno Haible DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL Bruno Haible BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

--------------------------------------------------------

Copyright 2003 Keith Packard

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Keith Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Keith Packard makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

KEITH PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL KEITH PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright (c) 2007-2009, Troy D. Hanson
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTicular PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGligence OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

----------------------------------------

Copyright 1992, 1993 by TOSHIBA Corp.

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted, provided
that the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the name of TOSHIBA not be used in advertising
or publicity pertaining to distribution of the software without specific,
written prior permission. TOSHIBA make no representations about the
suitability of this software for any purpose. It is provided "as is"
without express or implied warranty.

TOSHIBA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
TOSHIBA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.
Copyright IBM Corporation 1993

All Rights Reserved

License to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name of IBM not be
used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

IBM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS, AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS, IN NO EVENT SHALL
IBM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
 ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

Copyright 1990, 1991 by OMRON Corporation, NTT Software Corporation,
and Nippon Telegraph and Telephone Corporation

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the names of OMRON, NTT Software, and NTT
not be used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission. OMRON, NTT Software,
and NTT make no representations about the suitability of this
software for any purpose. It is provided "as is" without express or
implied warranty.

OMRON, NTT SOFTWARE, AND NTT, DISCLAIM ALL WARRANTIES WITH REGARD
TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS, IN NO EVENT SHALL OMRON, NTT SOFTWARE, OR NTT, BE
LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

This notice applies to the files in this directory. They are taken from
the libiconv-1.1 package, which is covered by the LGPL license. The files
Copyright (c) 1999-2000 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE FREE SOFTWARE FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the Free Software Foundation shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the Free Software Foundation.

Notes:

1. This copyright applies only to the files in this directory, and not to the remaining files in libiconv.

2. The Free Software Foundation does not encourage the use of the above license for newly written software.

1.1797 form-data 2.5.0
1.1797.1 Available under license:
MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE

1.1798 axios 0.19.2
1.1798.1 Available under license:
Copyright (c) 2014-present Matt Zabriskie

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1799 metrics-servlets 4.1.0

1.1800 node-fetch-h 2.3.0 2.3.0
1.1800.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 David Frank

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1801 formidable 1.2.1

1.1801.1 Available under license:

Copyright (C) 2011 Felix Geisendrfer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1802 commons-dbcp 2.6.0
1.1802.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons DBCP
Copyright 2001-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.1803 jsreport-tags 2.5.0
1.1803.1 Available under license :
MIT License

Copyright (c) 2017 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (C) 2001-2006 Bart Massey, Jamey Sharp, and Josh Triplett. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the names of the authors or their institutions shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the authors.

https-proxy-agent

### An HTTP(s) proxy `http.Agent` implementation for HTTPS

[![Build Status](https://travis-ci.org/TooTallNate/node-https-proxy-agent.svg?branch=master)](https://travis-ci.org/TooTallNate/node-https-proxy-agent)

This module provides an `http.Agent` implementation that connects to a specified...
HTTP or HTTPS proxy server, and can be used with the built-in `https` module.

Specifically, this `Agent` implementation connects to an intermediary "proxy" server and issues the [CONNECT HTTP method][CONNECT], which tells the proxy to open a direct TCP connection to the destination server.

Since this agent implements the CONNECT HTTP method, it also works with other protocols that use this method when connecting over proxies (i.e. WebSockets). See the "Examples" section below for more.

Installation
-------------

Install with `npm`:

```bash
$ npm install https-proxy-agent
```

Examples
--------

#### `https` module example

```js
var url = require('url');
var https = require('https');
var HttpsProxyAgent = require('https-proxy-agent');

// HTTP/HTTPS proxy to connect to
var proxy = process.env.http_proxy || 'http://168.63.76.32:3128';
console.log('using proxy server %j', proxy);

// HTTPS endpoint for the proxy to connect to
console.log('attempting to GET %j', endpoint);
var options = url.parse(endpoint);

// create an instance of the `HttpsProxyAgent` class with the proxy server information
var agent = new HttpsProxyAgent(proxy);
options.agent = agent;

https.get(options, function (res) {
    console.log("response" event!, res.headers);
    res.pipe(process.stdout);
});
```
### `ws` WebSocket connection example

```javascript
var url = require('url');
var WebSocket = require('ws');
var HttpsProxyAgent = require('https-proxy-agent');

// HTTP/HTTPS proxy to connect to
var proxy = process.env.http_proxy || 'http://168.63.76.32:3128';
console.log('using proxy server %j', proxy);

// WebSocket endpoint for the proxy to connect to
var endpoint = process.argv[2] || 'ws://echo.websocket.org';
var parsed = url.parse(endpoint);
console.log('attempting to connect to WebSocket %j', endpoint);

// create an instance of the `HttpsProxyAgent` class with the proxy server information
var options = url.parse(proxy);

var agent = new HttpsProxyAgent(options);

// finally, initiate the WebSocket connection
var socket = new WebSocket(endpoint, { agent: agent });

socket.on('open', function () {
    console.log("open" event!);
    socket.send('hello world');
});

socket.on('message', function (data, flags) {
    console.log("message" event! %j %j, data, flags);
    socket.close();
});
```

---

### API

---

#### new HttpsProxyAgent(Object options)

The `HttpsProxyAgent` class implements an `http.Agent` subclass that connects to the specified "HTTP(s) proxy server" in order to proxy HTTPS and/or WebSocket requests. This is achieved by using the [HTTP `CONNECT` method][CONNECT].

The `options` argument may either be a string URI of the proxy server to use, or an "options" object with more specific properties:
* `host` - String - Proxy host to connect to (may use `hostname` as well). Required.
* `port` - Number - Proxy port to connect to. Required.
* `protocol` - String - If `https:`, then use TLS to connect to the proxy.
* `headers` - Object - Additional HTTP headers to be sent on the HTTP CONNECT method.
* Any other options given are passed to the `net.connect()`/`tls.connect()` functions.

License
-------

(The MIT License)

Copyright (c) 2013 Nathan Rajlich &lt;nathan@tootallnate.net&gt;

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[CONNECT]: http://en.wikipedia.org/wiki/HTTP_tunnel#HTTP_CONNECT_Tunneling

Found in path(s):
* /opt/cola/permits/1112087480_1607035434.28/0/https-proxy-agent-2-2-4-6-tgz/package/README.md

1.1806 commons-io 2.6

1.1806.1 Available under license:

Apache Commons IO
Copyright 2002-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1807 ansicolors 0.2.1 0.2.1
1.1807.1 Available under license :
Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1808 aws4 1.8.0 1.8.0
1.1808.1 Available under license :
Copyright 2013 Michael Hart (michael.hart.au@gmail.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1809 buffer-indexof-polyfill 1.0.2
1.1809.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Sarosia

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1810 commons-dbcp 1.4
1.1810.1 Available under license :
Apache Commons DBCP

Copyright 2001-2010 The Apache Software Foundation
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1811 follow-redirects 1.14.5 1.14.5

1.1811.1 Available under license :
Copyright 2014-present Olivier Lalonde <olalonde@gmail.com>, James Talmage <james@talmage.io>, Ruben Verborgh

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR
IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1812 drill-protocol 1.19.0

1.1812.1 Available under license :
Drill : Protocol
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software 
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and 
limitations under the License.

1.1813 apib-include-directive 0.1.0 0.1.0

1.1813.1 Available under license :
MIT License

Copyright (c) 2016 Daniel G. Taylor  
Copyright (c) 2017 Keisuke Minami

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1814 readdirp 2.2.1

1.1814.1 Available under license :
This software is released under the MIT license:

Copyright (c) 2012-2015 Thorsten Lorenz

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1815 cookies 0.7.3
1.1815.1 Available under license :
(The MIT License)

Copyright (c) 2014 Jed Schmidt, http://jed.is/
Copyright (c) 2015-2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1816 httpdlog-parser 5.7
1.1816.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
* Apache HTTPD & NGINX Access log parsing made easy
* Copyright (C) 2011-2019 Niels Basjes
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/nginxHttpLogFormatDissector.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/HttpdLogFormatDissector.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/HttpUriDissector.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/ResponseSetCookieListDissector.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/translate/ConvertNumberIntoCLF.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/StrfTimeToDateTimeFormatter.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/Token.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/tokenformat/TokenOutputField.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/Version.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/nginxmodules/SslModule.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/StrfTimeToDateTimeFormatter.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/HttpdLoglineParser.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/tokenformat/NamedTokenParser.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-jar/nl/basjes/parse/httpdlog/dissectors/TimeStampDissector.java
* /opt/cola/permits/1202446997_1631567998.12/0/httpdlog-parser-5-7-sources-
No license file was found, but licenses were detected in source scan.

*/
* Apache HTTPD logparsing made easy
* Copyright (C) 2011-2019 Niels Basjes
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

1.1817 escape-html 1.0.3

1.1817.1 Available under license :
(The MIT License)

Copyright (c) 2012-2013 TJ Holowaychuk
Copyright (c) 2015 Andreas Lubbe
Copyright (c) 2015 Tiancheng "Timothy" Gu

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1818 icu 63

1.1818.1 Available under license :

```makefile
## -*-makefile-*-
## Copyright (C) 2016 and later: Unicode, Inc. and others.
## License & terms of use: http://www.unicode.org/copyright.html
## BSD-specific setup (FreeBSD, OpenBSD, NetBSD, *BSD)
## Copyright (c) 1999-2013, International Business Machines Corporation and
## others. All Rights Reserved.

## Commands to generate dependency files
GEN_DEPS.c=$(CC) -E -MM $(DEFS) $(CPPFLAGS)
GEN_DEPS.cc=$(CXX) -E -MM $(DEFS) $(CPPFLAGS)

## Flags for position independent code
SHAREDLIBCFLAGS = -fPIC
SHAREDLCXXFLAGS = -fPIC
SHAREDLIBCPPFLAGS = -DPIC

## Additional flags when building libraries and with threads
THREADSCPPFLAGS = -D_REENTRANT
LIBCPPFLAGS =

## Compiler switch to embed a runtime search path
LD_RPATH=
LD_RPATH_PRE=-Wl,-rpath,

## Compiler switch to embed a library name
LD_SONAME = -Wl,-soname -Wl,$(notdir $(MIDDLE_SO_TARGET))

## Shared library options
LD_SOOPTIONS= -Wl,-Bsymbolic

## Shared object suffix
```
SO = so  
## Non-shared intermediate object suffix  
STATIC_O = ao  
## Compilation rules  
%.$(STATIC_O): $(srcdir)/%.c  
$(COMPILE.c) $(STATICCPPFLAGS) $(STATICCFLAGS) -o $@ $<  
%.o: $(srcdir)/%.c  
$(COMPILE.c) $(DYNAMICCPPFLAGS) $(DYNAMICCFLAGS) -o $@ $<  
%.$(STATIC_O): $(srcdir)/%.cpp  
$(COMPILE.cc) $(STATICCPPFLAGS) $(STATICCXXFLAGS) -o $@ $<  
%.o: $(srcdir)/%.cpp  
$(COMPILE.cc) $(DYNAMICCPPFLAGS) $(DYNAMICCXXFLAGS) -o $@ $<  
## Dependency rules  
%.d: $(srcdir)/%.c  
@echo "generating dependency information for $<"  
@$(SHELL) -ec '$(GEN_DEPS.c) $< \n    | sed \''s/$\(.*\).o \([ :]*/\1.o $@ : /g\'' > $@; \n    [ -s $@ ] || rm -f $@'  
%.d: $(srcdir)/%.cpp  
@echo "generating dependency information for $<"  
@$(SHELL) -ec '$(GEN_DEPS.cc) $< \n    | sed \''s/$\(.*\).o \([ :]*/\1.o $@ : /g\'' > $@; \n    [ -s $@ ] || rm -f $@'  
## Versioned libraries rules  
$(RM) $@ & & ln -s ${<F} $@  
%.$(SO): %.$(SO).$(SO_TARGET_VERSION_MAJOR)  
$(RM) $@ & & ln -s {<F}.$(SO).$(SO_TARGET_VERSION) $@  
## Bind internal references  
# LDflags that pkgdata will use  
BIR_LDFLAGS= -Wl,-Bsymbolic  
# Dependencies [i.e. map files] for the final library  
BIR_DEPS=  
## Remove shared library 's'  
STATIC_PREFIX_WHEN_USED =  
STATIC_PREFIX =
COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either
(a) this copyright and permission notice appear with all copies of the Data Files or Software, or
(b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.


# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
# # Redistributions of source code must retain the above copyright notice,
# # this list of conditions and the following disclaimer.
# # Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
#   - https://sourceforge.net/project/?group_id=1519
#   - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#   - Its license terms and conditions are shown below.
#
# --------COPYING.libtabe ---- BEGIN-------------------
#
# /*
# * Copyright (c) 1999 TaBE Project.
# * Copyright (c) 1999 Pai-Hsiang Hsiao.
# * All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions and the following disclaimer in
# the documentation and/or other materials provided with the
# distribution.
# Neither the name of the TaBE Project nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.

# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.

# /*

# Copyright (c) 1999 Computer Systems and Communication Lab,
# Institute of Information Science, Academia Sinica. All rights reserved.

# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:

# Redistributions of source code must retain the above copyright
# notice, this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions and the following disclaimer in
# the documentation and/or other materials provided with the
# distribution.
# Neither the name of the Computer Systems and Communication Lab
# nor the names of its contributors may be used to endorse or
# promote products derived from this software without specific
# prior written permission.

# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

Open Source Used In DNAC Platform Antelope 6266
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */

# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-tsai4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsai4
#
#
# ---------------COPYING.libtable-----END--------------------------------
#
#
# ---------------COPYING.ipadic-----BEGIN--------------------------------
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise. The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
# project or thereafter. Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# ~~~~~~~~~~~~~~~~~COPYING.ipadic-----END~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

3. Lao Word Break Dictionary Data (laodict.txt)

# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
# (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# ----------------------------------------------------------------------
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
# ----------------------------------------------------------------------

4. Burmese Word Break Dictionary Data (burmesedict.txt)

# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
# github.com/kanyawtech/myanmar-karen-word-lists
#
# Copyright (c) 2013, LeRoy Benjamin Sharon
# All rights reserved.
#
5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.
# making a contribution to the database or code waives all rights to
# future claims in that contribution or in the TZ Database.

6. Google double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following
  disclaimer in the documentation and/or other materials provided
  with the distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1819 ip-address 5.8.9

1.1820 react-is 16.12.0 16.12.0

1.1820.1 Available under license :

MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1821 tmp 0.1.0

1.1821.1 Available under license :

Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1822 @babel/helper-annotate-as-pure

7.12.13
1.1822.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1823 escalade 3.1.1
1.1823.1 Available under license:
MIT License

Copyright (c) Luke Edwards <luke.edwards05@gmail.com> (lukeed.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1824 end-of-stream 1.4.4

1.1824.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1825 jackson-module-afterburner 2.12.1

1.1825.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components (as well their dependencies) may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits
A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

This copy of Jackson JSON processor `jackson-module-afterburner` module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Additionally licensing information exists for following 3rd party library dependencies

### ASM

ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.1826 import-lazy 2.1.0
1.1826.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1827 json-path 0.11.2 0.11.2
1.1827.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1828 chokidar 3.3.0

1.1828.1 Available under license:

The MIT License (MIT)

Copyright (c) 2012-2019 Paul Miller (https://paulmillr.com), Elan Shanker

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1829 forever-agent 0.6.1

1.1829.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1830 prop-types 15.7.2
1.1830.1 Available under license :
MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1831 bunyan 1.8.1
1.1831.1 Available under license :
# This is the MIT license

Copyright 2016 Trent Mick
Copyright 2016 Joyent Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, 
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be included 
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS 
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. 
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY 
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, 
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE 
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1832 javolution 5.5.1
1.1832.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**
 * Interface for an element's attribute specifications.
 *
 * &lt;blockquote&gt;
 * &lt;em&gt;This module, both source code and documentation, is in the 
 * Public Domain, and comes with &lt;strong&gt;NO WARRANTY &lt;/strong&gt;.&lt;/em&gt;
 * See &lt;a href='http://www.saxproject.org'>http://www.saxproject.org&lt;/a&gt;
 * for further information.
 * &lt;/blockquote&gt;
 *
 * &lt;p&gt;This is the original SAX1 interface for reporting an element's 
 * attributes. Unlike the new { &lt;link org.xml.sax.Attributes Attributes} 
 * interface, it does not support Namespace-related information.&lt;/p&gt;
 * 
 * &lt;p&gt;When an attribute list is supplied as part of a 
 * { &lt;link org.xml.sax.DocumentHandler#startElement startElement} 
 * event, the list will return valid results only during the 
 * scope of the event; once the event handler returns control 
 * to the parser, the attribute list is invalid. To save a 
 * persistent copy of the attribute list, use the SAX1 AttributeSetImpl 
 * helper class.&lt;/p&gt;
 *
 * &lt;p&gt;An attribute list includes only attributes that have been 
 * specified or defaulted: #IMPLIED attributes will not be included.&lt;/p&gt;
 * 
 * &lt;p&gt;There are two ways for the SAX application to obtain information
from the AttributeList. First, it can iterate through the entire list:

```java
public void startElement (String name, AttributeList atts) {
  for (int i = 0; i < atts.getLength(); i++) {
    String name = atts.getName(i);
    String type = atts.getType(i);
    String value = atts.getValue(i);
    [...]  
  }
}
```

(Note that the result of getLength() will be zero if there are no attributes.)

As an alternative, the application can request the value or type of specific attributes:

```java
public void startElement (String name, AttributeList atts) {
  String identifier = atts.getValue("id");
  String label = atts.getValue("label");
  [...] 
}
```

@since SAX 1.0
@author David Megginson
@version 2.0.1 (sax2r2)
*/

Found in path(s):
* `/opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/AttributeList.java`

No license file was found, but licenses were detected in source scan.

/**
 * Receive notification of general document events.
 *
 * <blockquote>
 * <em>This module, both source code and documentation, is in the Public Domain, and comes with NO WARRANTY.</em>
 * </blockquote>
 *
 * This was the main event-handling interface for SAX1; in
* SAX2, it has been replaced by \{@link org.xml.sax.ContentHandler \ContentHandler\}, which provides Namespace support and reporting
* of skipped entities. This interface is included in SAX2 only
* to support legacy SAX1 applications.</p><p>
* <p>The order of events in this interface is very important, and
* mirrors the order of information in the document itself. For
* example, all of an element's content (character data, processing
* instructions, and/or subelements) will appear, in order, between
* the startElement event and the corresponding endElement event.</p><p>
* <p>Application writers who do not want to implement the entire
* interface can derive a class from HandlerBase, which implements
* the default functionality; parser writers can instantiate
* HandlerBase to obtain a default handler. The application can find
* the location of any document event using the Locator interface
* supplied by the Parser through the setDocumentLocator method.</p><p>
* @since SAX 1.0
* @author David Megginson
* @version 2.0.1 (sax2r2)
* @see org.xml.sax.Parser#setDocumentHandler
* @see org.xml.sax.Locator
* @see org.xml.sax.HandlerBase
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-
jar/org/xml/sax/DocumentHandler.java
No license file was found, but licenses were detected in source scan.

/**
* Basic interface for resolving entities.
* 
* <blockquote>
* <em>This module, both source code and documentation, is in the
* Public Domain, and comes with <strong>NO WARRANTY</strong>.</em>
* See <a href='http://www.saxproject.org'>http://www.saxproject.org</a>
* for further information.
* </blockquote>
* 
* <p>If a SAX application needs to implement customized handling
* for external entities, it must implement this interface and
* register an instance with the SAX driver using the
* \{@link org.xml.sax.XMLReader#setEntityResolver setEntityResolver\}
* method.</p>
* 
* <p>The XML reader will then allow the application to intercept any
Many SAX applications will not need to implement this interface, but it will be especially useful for applications that build XML documents from databases or other specialised input sources, or for applications that use URI types other than URLs.

The following resolver would provide the application with a special character stream for the entity with the system identifier "http://www.myhost.com/today":

```
import org.xml.sax.EntityResolver;
import org.xml.sax.InputSource;

public class MyResolver implements EntityResolver {
    public InputSource resolveEntity (String publicId, String systemId)
    {
        if (systemId.equals("http://www.myhost.com/today")) {
            // return a special input source
            MyReader reader = new MyReader();
            return new InputSource(reader);
        } else {
            // use the default behaviour
            return null;
        }
    }
}
```

The application can also use this interface to redirect system identifiers to local URIs or to look up replacements in a catalog (possibly by using the public identifier).

@since SAX 1.0
@author David Megginson
@version 2.0.1 (sax2r2)
@see org.xml.sax.XMLReader#setEntityResolver
@see org.xml.sax.InputSource
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/EntityResolver.java
No license file was found, but licenses were detected in source scan.

/**
 * Interface for associating a SAX event with a document location.
This module, both source code and documentation, is in the Public Domain, and comes with NO WARRANTY. See [http://www.saxproject.org](http://www.saxproject.org) for further information.

If a SAX parser provides location information to the SAX application, it does so by implementing this interface and then passing an instance to the application using the content handler's `setDocumentLocator` method. The application can use the object to obtain the location of any other SAX event in the XML source document.

Note that the results returned by the object will be valid only during the scope of each callback method: the application will receive unpredictable results if it attempts to use the locator at any other time, or after parsing completes.

SAX parsers are not required to supply a locator, but they are very strongly encouraged to do so. If the parser supplies a locator, it must do so before reporting any other document events. If no locator has been set by the time the application receives the `startDocument` event, the application should assume that a locator is not available.

@since SAX 1.0
@version 2.0.1 (sax2r2)
@see org.xml.sax.ContentHandler#setDocumentLocator

No license file was found, but licenses were detected in source scan.

// NO WARRANTY! This class is in the Public Domain.
/**
 * Exception class for an unsupported operation.
 * 
 * @author David Megginson
 * @version 2.0.1 (sax2r2)
 */

Found in path(s):
//opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/Locator.java

No license file was found, but licenses were detected in source scan.
* <p>An XMLReader will throw this exception when it recognizes a feature or property identifier, but cannot perform the requested operation (setting a state or value). Other SAX2 applications and extensions may use this class for similar purposes.</p>
* 
* @since SAX 2.0
* @author David Megginson
* @version 2.0.1 (sax2r2)
* @see org.xml.sax.SAXNotRecognizedException
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/SAXNotSupportedException.java
No license file was found, but licenses were detected in source scan.

/*
* Javolution - Java(TM) Solution for Real-Time and Embedded Systems
* Copyright (C) 2005 - Javolution (http://javolution.org/)
* All rights reserved.
*
* Permission to use, copy, modify, and distribute this software is freely granted, provided that this notice is preserved.
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/xml/sax/XMLReaderImpl.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/java/lang/EnumConstantNotPresentException.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/util/ReentrantLock.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javax/realtime/MemoryArea.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/java/io/Serializable.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/lang/Immutable.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/java/util/Formattable.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javax/realtime/RealtimeThread.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/util/logging/Logger.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/java/util/logging/Logger.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/java/util/FormatFlagsConversionMismatchException.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-
Open Source Used In DNAC Platform Antelope
No license file was found, but licenses were detected in source scan.

/**
 * Javolution - Java(TM) Solution for Real-Time and Embedded Systems
 * Copyright (C) 2009 - Javolution (http://javolution.org/)
 * All rights reserved.
 *
 * Permission to use, copy, modify, and distribute this software is
 * freely granted, provided that this notice is preserved.
 */

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javadoc/context/PoolContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javadoc/context/PoolContext.java

No license file was found, but licenses were detected in source scan.
* Basic interface for SAX (Simple API for XML) parsers.

*<blockquote>
<em>This module, both source code and documentation, is in the Public Domain, and comes with <strong>NO WARRANTY</strong>.</em>
See [http://www.saxproject.org](http://www.saxproject.org) for further information.
</blockquote>

*<p>This was the main event supplier interface for SAX1; it has been replaced in SAX2 by [XMLReader], which includes Namespace support and sophisticated configurability and extensibility.</p>

*<p>All SAX1 parsers must implement this basic interface: it allows applications to register handlers for different types of events and to initiate a parse from a URI, or a character stream.</p>

*<p>All SAX1 parsers must also implement a zero-argument constructor (though other constructors are also allowed).</p>

*<p>SAX1 parsers are reusable but not re-entrant: the application may reuse a parser object (possibly with a different input source) once the first parse has completed successfully, but it may not invoke the parse() methods recursively within a parse.</p>

*<p>@deprecated This interface has been replaced by the SAX2 XMLReader interface, which includes Namespace support.</p>

*<a href='http://www.saxproject.org'>http://www.saxproject.org</a>

/*
 * Encapsulate an XML parse error or warning.
 */
* This module, both source code and documentation, is in the Public Domain, and comes with NO WARRANTY.
* See [http://www.saxproject.org](http://www.saxproject.org) for further information.
* 

This exception may include information for locating the error in the original XML document, as if it came from a Locator object. Note that although the application will receive a SAXParseException as the argument to the handlers in the ErrorHandler interface, the application is not actually required to throw the exception; instead, it can simply read the information in it and take a different action.

Since this exception is a subclass of SAXException, it inherits the ability to wrap another exception.

@since SAX 1.0  
@author David Megginson  
@version 2.0.1 (sax2r2)  
@see org.xml.sax.SAXException  
@see org.xml.sax.Locator  
@see org.xml.sax.ErrorHandler

Since this exception is a subclass of SAXException, it inherits the ability to wrap another exception.

@since SAX 1.0  
@author David Megginson  
@version 2.0.1 (sax2r2)  
@see org.xml.sax.SAXException  
@see org.xml.sax.Locator  
@see org.xml.sax.ErrorHandler

No license file was found, but licenses were detected in source scan.

// NO WARRANTY!  This class is in the Public Domain.
/**
 * Interface for an XML filter.
 * 
 * This module, both source code and documentation, is in the Public Domain, and comes with NO WARRANTY.
 * See [http://www.saxproject.org](http://www.saxproject.org) for further information.
 * 
 * An XML filter is like an XML reader, except that it obtains its events from another XML reader rather than a primary source like an XML document or database. Filters can modify a stream of events as they pass on to the final application.
 * 
 * The XMLFilterImpl helper class provides a convenient base
* for creating SAX2 filters, by passing on all {@link org.xml.sax.EntityResolver EntityResolver}, {@link org.xml.sax.DTDHandler DTDHandler}, {@link org.xml.sax.ContentHandler ContentHandler} and {@link org.xml.sax.ErrorHandler ErrorHandler} events automatically. </p>

* @since SAX 2.0
* @author David Megginson
* @version 2.0.1 (sax2r2)
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/XMLFilter.java

No license file was found, but licenses were detected in source scan.

/*
* Javolution - Java(TM) Solution for Real-Time and Embedded Systems
* Copyright (C) 2006 - Javolution (http://javolution.org/)
* All rights reserved.
*
* Permission to use, copy, modify, and distribute this software is
* freely granted, provided that this notice is preserved.
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/lang/ClassInitializer.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/xml/sax/Attributes.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/xml/XMLObjectReader.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/context/HeapContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/xml/stream/XMLStreamReaderImpl.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/xml/stream/XMLStreamWriter.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/xml/stream/NamespacesImpl.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/io/AppendableWriter.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/io/Appender.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/context/Allocator.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/xml/XMLReferenceResolver.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/util/FastCollection.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/xml/XMLFormat.java
Open Source Used In DNAC Platform Antelope 6295

* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/text/TextBuilder.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/io/AppendableWriter.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/util/FastIterator.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/text/TextBuilder.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/security/security/StackContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/util/StandardLog.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/context/PersistentContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/xml/XMLObjectWriter.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/xml/sax/ContentHandler.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/context/ConcurrentException.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/context/LogContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/context/ObjectFactory.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/templates/javolution/context/ConcurrentThread.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/io/UTF8StreamWriter.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/io/UTF8StreamReader.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/xml/XMLElementReader.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/xml/XMLObjectReader.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/xml/XMLFormat.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/xml/sax/DefaultHandler.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/xml/stream/XMLStreamWriter.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/xml/stream/XMLStreamReader.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javadoc/javadoc/Reference.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javadoc/javadoc/Reference.java
* /opt/cola/permit...
Open Source Used In DNAC Platform Antelope
No license file was found, but licenses were detected in source scan.

/**
 * Receive notification of basic DTD-related events.
 *
 * <blockquote>
 * <em>This module, both source code and documentation, is in the
 * Public Domain, and comes with <strong>NO WARRANTY</strong>.</em>
 * See <a href='http://www.saxproject.org'>http://www.saxproject.org</a>
 * for further information.
 * </blockquote>
 *
 * <p>If a SAX application needs information about notations and
 * unparsed entities, then the application implements this
 * interface and registers an instance with the SAX parser using
 * the parser's setDTDHandler method. The parser uses the
 * instance to report notation and unparsed entity declarations to
 * the application.</p>
 *
 * <p>Note that this interface includes only those DTD events that
 * the XML recommendation <em>requires</em> processors to report:
 * notation and unparsed entity declarations.</p>
 *
 * The SAX parser may report these events in any order, regardless
 * of the order in which the notations and unparsed entities were
 * declared; however, all DTD events must be reported after the
 * document handler's startDocument event, and before the first
 * startElement event.
 * (If the LexicalHandler is
 * used, these events must also be reported before the endDTD event.)
 * </p>
 *
 * It is up to the application to store the information for
 * future use (perhaps in a hash table or object tree).
 * If the application encounters attributes of type "NOTATION",
 * "ENTITY", or "ENTITIES", it can use the information that it
 * obtained through this interface to find the entity and/or
 * notation corresponding with the attribute value.</p>
 *
 * @since SAX 1.0
 * @author David Megginson
 * @version 2.0.1 (sax2r2)
 */
Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/DTDHandler.java
No license file was found, but licenses were detected in source scan.

/ *
* Javolution - Java(TM) Solution for Real-Time and Embedded Systems
* Copyright (C) 2008 - Javolution (http://javolution.org/)
* All rights reserved.
* *
* Permission to use, copy, modify, and distribute this software is
* freely granted, provided that this notice is preserved.
* */

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/util/FastBitSet.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/util/FastBitSet.java
No license file was found, but licenses were detected in source scan.

/ *
* Javolution - Java(TM) Solution for Real-Time and Embedded Systems
* Copyright (C) 2009 - Javolution (http://javolution.org/)
* All rights reserved.
* *
* Permission to use, copy, modify, and distribute this software is
* freely granted, provided that this notice is preserved.
* */

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/text/Cursor.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/text/Cursor.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/testing/JUnitContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/testing/JUnitContext.java
No license file was found, but licenses were detected in source scan.

// NO WARRANTY! This class is in the public domain.
/**
* Interface for a list of XML attributes.
* *
* <blockquote>
* <em>This module, both source code and documentation, is in the
* Public Domain, and comes with <strong>NO WARRANTY</strong>.</em>
* See <a href="http://www.saxproject.org">http://www.saxproject.org</a>
* for further information.
* </blockquote>
This interface allows access to a list of attributes in three different ways:

- by attribute index;
- by Namespace-qualified name; or
- by qualified (prefixed) name.

The list will not contain attributes that were declared #IMPLIED but not specified in the start tag. It will also not contain attributes used as Namespace declarations (xmlns*) unless the feature is set to true (it is false by default).

Because SAX2 conforms to the original "Namespaces in XML" recommendation, it normally does not give namespace declaration attributes a namespace URI.

Some SAX2 parsers may support using an optional feature flag to request that those attributes be given URIs, conforming to a later backwards-incompatible revision of that recommendation. (The attribute's "local name" will be the prefix, or "xmlns" when defining a default element namespace.) For portability, handler code should always resolve that conflict, rather than requiring parsers that can change the setting of that feature flag.

If the namespace-prefixes feature (see above) is false, access by qualified name may not be available; if the feature is false, access by Namespace-qualified names may not be available.

This interface replaces the now-deprecated SAX1 AttributeList interface, which does not contain Namespace support. In addition to Namespace support, it adds the getIndex methods (below).

The order of attributes in the list is unspecified, and will vary from implementation to implementation.
**Encapsulate a general SAX error or warning.**

* *

* `<em>` This module, both source code and documentation, is in the
* Public Domain, and comes with `<strong>` NO WARRANTY </strong> .</em>*
* See `<a href='http://www.saxproject.org'>http://www.saxproject.org</a>`
* for further information.

* `</em>`

* `<p>` This class can contain basic error or warning information from
* either the XML parser or the application: a parser writer or
* application writer can subclass it to provide additional
* functionality. SAX handlers may throw this exception or
* any exception subclassed from it.</p>

* *

* `<p>` If the application needs to pass through other types of
* exceptions, it must wrap those exceptions in a SAXException
* or an exception derived from a SAXException.</p>

* *

* `<p>` If the parser or application needs to include information about a
* specific location in an XML document, it should use the
* `[ @link org.xml.sax.SAXParseException SAXParseException ] subclass.</p>

* *

* `@since` SAX 1.0
* `@author` David Megginson
* `@version` 2.0.1 (sax2r2)
* `@see` org.xml.sax.SAXParseException

*/

// NO WARRANTY! This class is in the Public Domain.

/**

* Interface for reading an XML document using callbacks.

* *

* `<blockquote>`

* `<em>` This module, both source code and documentation, is in the
* Public Domain, and comes with `<strong>` NO WARRANTY </strong> .</em>*
* See `<a href='http://www.saxproject.org'>http://www.saxproject.org</a>`
* for further information.
<p>Note: despite its name, this interface does not extend the standard Java Reader interface, because reading XML is a fundamentally different activity than reading character data.</p>

<p>XMLReader is the interface that an XML parser's SAX2 driver must implement. This interface allows an application to set and query features and properties in the parser, to register event handlers for document processing, and to initiate a document parse.</p>

<p>All SAX interfaces are assumed to be synchronous: the methods must not return until parsing is complete, and readers must wait for an event-handler callback to return before reporting the next event.</p>

<p>This interface replaces the (now deprecated) SAX 1.0 interface. The XMLReader interface contains two important enhancements over the old Parser interface (as well as some minor ones):</p>

<ul>
  <li>it adds a standard way to query and set features and properties; and</li>
  <li>it adds Namespace support, which is required for many higher-level XML standards.</li>
</ul>

<p>There are adapters available to convert a SAX1 Parser to a SAX2 XMLReader and vice-versa.</p>

@since SAX 2.0
@author David Megginson
@version 2.0.1+ (sax2r3pre1)
@see org.xml.sax.XMLFilter
/

Found in path(s):
/opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/XMLReader.java
No license file was found, but licenses were detected in source scan.

/*
 * Javolution - Java(TM) Solution for Real-Time and Embedded Systems
 * Copyright (C) 2006 - Javolution (http://javolution.org/)
 * All rights reserved.
 */
* Permission to use, copy, modify, and distribute this software is
* freely granted, provided that this notice is preserved.
*/
/
* ****************************************************************************

* Copyright (C) 1993 by Sun Microsystems, Inc. All rights reserved.
*
* Developed at SunSoft, a Sun Microsystems, Inc. business.
* Permission to use, copy, modify, and distribute this
* software is freely granted, provided that this notice
* is preserved.
* ****************************************************************************
*
*/
/
* ****************************************************************************

* Copyright (C) 2004 by Sun Microsystems, Inc. All rights reserved.
*
* Permission to use, copy, modify, and distribute this
* software is freely granted, provided that this notice
* is preserved.
* ****************************************************************************
*
*/
/

---

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/lang/MathLib.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/lang/MathLib.java

No license file was found, but licenses were detected in source scan.

/**
 * Default base class for handlers.
 * 
 * <blockquote>
 * <em>This module, both source code and documentation, is in the
 * Public Domain, and comes with <strong>NO WARRANTY</strong>. </em>
 * See <a href='http://www.saxproject.org'>http://www.saxproject.org</a>
 * for further information.
 * </blockquote>
 *
 * <p>This class implements the default behaviour for four SAX1
 * interfaces: EntityResolver, DTDHandler, DocumentHandler,
 * and ErrorHandler. It is now obsolete, but is included in SAX2 to
 * support legacy SAX1 applications. SAX2 applications should use
 * the DefaultHandler class instead.</p>
 *
 * <p>Application writers can extend this class when they need to
 * implement only part of an interface; parser writers can

---
* instantiate this class to provide default handlers when the
* application has not supplied its own.</p>
* 
* <p>Note that the use of this class is optional.</p>
* 
* @deprecated This class works with the deprecated DocumentHandler
* interface. It has been replaced by the SAX2 DefaultHandler
* class.
* @since SAX 1.0
* @author David Megginson
* @version 2.0.1 (sax2r2)
* @see org.xml.sax.EntityResolver
* @see org.xml.sax.DTDHandler
* @see org.xml.sax.DocumentHandler
* @see org.xml.sax.ErrorHandler
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/HandlerBase.java
No license file was found, but licenses were detected in source scan.

/**
 * Basic interface for SAX error handlers.
 *
 * <blockquote>
 * <em>This module, both source code and documentation, is in the
 * Public Domain, and comes with <strong>NO WARRANTY</strong>.</em>
 * See <a href='http://www.saxproject.org'>http://www.saxproject.org</a>
 * for further information.
 * </blockquote>
 *
 * <p>If a SAX application needs to implement customized error
 * handling, it must implement this interface and then register an
 * instance with the XML reader using the
 * { @link org.xml.sax.XMLReader#setErrorHandler setErrorHandler }
 * method. The parser will then report all errors and warnings
 * through this interface.</p>
 *
 * <p><strong>WARNING:</strong> If an application does <em>not</em>
 * register an ErrorHandler, XML parsing errors will go unreported,
 * except that <em>SAXParseException</em>s will be thrown for fatal errors.
 * In order to detect validity errors, an ErrorHandler that does something
 * with { @link #error error() } calls must be registered.</p>
 *
 * <p>For XML processing errors, a SAX driver must use this interface
 * in preference to throwing an exception: it is up to the application
 * to decide whether to throw an exception for different types of
 * errors and warnings. Note, however, that there is no requirement that
* the parser continue to report additional errors after a call to
  * [@link fatalError fatalError]. In other words, a SAX driver class
  * may throw an exception after reporting any fatalError.
  * Also parsers may throw appropriate exceptions for non-XML errors.
  * For example, [@link XMLReader.parse XMLReader.parse()] would throw
  * an IOException for errors accessing entities or the document.</p>
*  
* @since SAX 1.0  
* @author David Megginson  
* @version 2.0.1+ (sax2r3pre1)  
* @see org.xml.sax.XMLReader#setErrorHandler  
* @see org.xml.sax.SAXParseException
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/ErrorHandler.java

No license file was found, but licenses were detected in source scan.

// NO WARRANTY!  This class is in the Public Domain.
/**
* Exception class for an unrecognized identifier.
*  
* <blockquote>
* This module, both source code and documentation, is in the
* Public Domain, and comes with <strong>NO WARRANTY</strong>.  
* See <a href='http://www.saxproject.org'>http://www.saxproject.org</a>  
* for further information.
* </blockquote>

* An XMLReader will throw this exception when it finds an
* unrecognized feature or property identifier; SAX applications and
* extensions may use this class for other, similar purposes.</p>
*  
* @since SAX 2.0  
* @author David Megginson  
* @version 2.0.1 (sax2r2)  
* @see org.xml.sax.SAXNotRecognizedException
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/SAXNotRecognizedException.java

No license file was found, but licenses were detected in source scan.

/**
* A single input source for an XML entity.
*  
* <blockquote>
This module, both source code and documentation, is in the Public Domain, and comes with NO WARRANTY. See http://www.saxproject.org for further information.

This class allows a SAX application to encapsulate information about an input source in a single object, which may include a public identifier, a system identifier, a byte stream (possibly with a specified encoding), and/or a character stream.

There are two places that the application can deliver an input source to the parser: as the argument to the Parser.parse method, or as the return value of the EntityResolver.resolveEntity method.

The SAX parser will use the InputSource object to determine how to read XML input. If there is a character stream available, the parser will read that stream directly, disregarding any text encoding declaration found in that stream. If there is no character stream, but there is a byte stream, the parser will use that byte stream, using the encoding specified in the InputSource or else (if no encoding is specified) autodetecting the character encoding using an algorithm such as the one in the XML specification. If neither a character stream nor a byte stream is available, the parser will attempt to open a URI connection to the resource identified by the system identifier.

An InputSource object belongs to the application: the SAX parser shall never modify it in any way (it may modify a copy if necessary). However, standard processing of both byte and character streams is to close them on as part of end-of-parse cleanup, so applications should not attempt to re-use such streams after they have been handed to a parser.

@since SAX 1.0
@author David Megginson
@version 2.0.1 (sax2r2)
@see org.xml.sax.XMLReader#parse(org.xml.sax.InputSource)
@see org.xml.sax.EntityResolver#resolveEntity
// NO WARRANTY! This class is in the public domain.
/**
 * Receive notification of the logical content of a document.
 *
 * <blockquote>
 * <em>This module, both source code and documentation, is in the
 * Public Domain, and comes with <strong>NO WARRANTY</strong>. </em>
 * See <a href='http://www.saxproject.org'>http://www.saxproject.org</a>
 * for further information.
 * </blockquote>
 *
 * This is the main interface that most SAX applications
 * implement: if the application needs to be informed of basic parsing
 * events, it implements this interface and registers an instance with
 * the SAX parser using the @link org.xml.sax.XMLReader#setContentHandler
 * setContentHandler method. The parser uses the instance to report
 * basic document-related events like the start and end of elements
 * and character data.</p>
 *
 * The order of events in this interface is very important, and
 * mirrors the order of information in the document itself. For
 * example, all of an element's content (character data, processing
 * instructions, and/or subelements) will appear, in order, between
 * the startElement event and the corresponding endElement event.</p>
 *
 * This interface is similar to the now-deprecated SAX 1.0
 * DocumentHandler interface, but it adds support for Namespaces
 * and for reporting skipped entities (in non-validating XML
 * processors).</p>
 *
 * Implementors should note that there is also a
 * <code>ContentHandler</code> class in the <code>java.net</code> package; that means that it's probably a bad idea to do</p>
 *
 * In fact, "import ...*" is usually a sign of sloppy programming
 * anyway, so the user should consider this a feature rather than a
 * bug.</p>
 *
 * @since SAX 2.0
 * @author David Megginson
 * @version 2.0.1+ (sax2r3pre1)
 * @see org.xml.sax.XMLReader
 * @see org.xml.sax.DTDHandler
 * @see org.xml.sax.ErrorHandler
Open Source Used In DNAC Platform Antelope 6308

*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/org/xml/sax/ContentHandler.java

No license file was found, but licenses were detected in source scan.

*/

* Javolution - Java(TM) Solution for Real-Time and Embedded Systems
* Copyright (C) 2007 - Javolution (http://javolution.org/)
* All rights reserved.
* 
* Permission to use, copy, modify, and distribute this software is
* freely granted, provided that this notice is preserved.
*/

Found in path(s):
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/testing/AssertionException.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/xml/XMLSerializable.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/xml/QName.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/context/Context.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/context/ArrayFactory.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/xml/ws/WebServiceClient.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/context/Context.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/context/Context.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/context/ImmortalContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/testing/TestContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/testing/TimeContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/lang/ValueType.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/testing/TestCase.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/testing/TestSuite.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/lang/Realtime.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/context/ImmortalContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/_templates/javolution/testing/TestContext.java
* /opt/cola/permits/1118987831_1608672578.56/0/javolution-5-5-1-sources-2-jar/javolution/xml/QName.java
1.1833 json-stringify-safe 5.0.1 5.0.1

1.1833.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1834 babel-helper-function-name 6.24.1

6.24.1
1.1834.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1835 groovy-ant 3.0.8
1.1835.1 Available under license:
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.

Found in path(s):
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources-jar/groovy/ant/antlib.xml
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources-jar/org/codehaus/groovy/antlib.xml
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources-
/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/UberCompileTask.java
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/AntProjectPropertiesDelegate.java
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/GenerateStubsTask.java
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/GroovycTask.java
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/FileSystemCompilerFacade.java
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/CompileTaskSupport.java
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/Groovy.java
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/GenerateStubsTask.java
* /opt/cola/permits/1218310310_1634928239.71/0/groovy-ant-3-0-8-sources.jar/org/codehaus/groovy/ant/GroovycTask.java

No license file was found, but licenses were detected in source scan.

1.1836 xmldom 0.1.27
1.1836.1 Available under license:
You can choose any one of those:

The MIT License (MIT):

link:http://opensource.org/licenses/MIT

LGPL:
http://www.gnu.org/licenses/lgpl.html

1.1837 npm-install-package 2.1.0
1.1837.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Yoshua Wuyts

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1838 yauaa 5.20
1.1838.1 Available under license:

No license file was found, but licenses were detected in source scan.

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# distributed under the License is distributed on an AS IS BASIS,

Found in path(s):
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/LG.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Opera.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/GoogleChrome.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/GameConsoles.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Bada.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Alcatel.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Watches.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Rbots-PriceSpiders.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/GoogleFavicon.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Windows.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/MobileBrand-rules.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Tencent.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Lenovo-Tests.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Huawei.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Ubuntu-tests.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Robots.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Robots-Google.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/MobileBrowser-tests.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Maxthon.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/OtherOS.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Linux.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Cubot.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Nokia.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/WindowsCE.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/MSEdge.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/GoogleChrome70.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Apple.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Hackers.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/WindowsPhone.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Amazon.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/GeckoFirefox.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/TV.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/GoannaPaleMoon.yaml
* /opt/cola/permits/120246250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/UUID.yaml
No license file was found, but licenses were detected in source scan.

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/OperatingSystemTests.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/AppleTypes.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/WindowsOEMCodes.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/WindowsPhoneLookups.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/UbuntuVersionLookups.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/OperatingSystemDeviceNames.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/MobileBrands.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/Ubuntu-18.10-tests.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/CPUTypes.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/AmazonDevices.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/RobotBaseRules.yaml
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/UserAgents/AndroidDevice Overrides.yaml

No license file was found, but licenses were detected in source scan.

2020 Niels Basjes
~
~ Licensed under the Apache License, Version 2.0 (the "License");
~ you may not use this file except in compliance with the License.
~ You may obtain a copy of the License at
~
~ https://www.apache.org/licenses/LICENSE-2.0

~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS,
~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
~ See the License for the specific language governing permissions and
~ limitations under the License.

Found in path(s):
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/META-INF/maven/nl.basjes.parse.useragent/yauaa/pom.xml

No license file was found, but licenses were detected in source scan.

/*
* Yet Another UserAgent Analyzer
* Copyright (C) 2013-2020 Niels Basjes
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/debug/FlattenPrinter.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/UserAgent.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/annotate/UserAgentAnnotationMapper.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/analyze/treewalker/steps/walk/StepUp.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/analyze/treewalker/steps/walk/StepDown.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/analyze/treewalker/steps/compare/StepEndsWith.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/analyze/treewalker/steps/compare/StepDefaultIfNull.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/analyze/treewalker/steps/compare/StepIsNull.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/classify/DeviceClass.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/calculate/CalculateDeviceBrand.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/calculate/ConcatNONDuplicatedCalculator.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/calculate/MajorVersionCalculator.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/calculate/CalculateAgentEmail.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/analyze/InvalidParserConfigurationException.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/utils/YauaaVersion.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/analyze/treewalker/steps/lookup/StepLookupPrefix.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/calculate/DeviceClass.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/UserAgentAnalyzer.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/UserAgent.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/annotate/UserAgentAnnotationMapper.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/analyze/treewalker/steps/walk/StepUp.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/analyze/treewalker/steps/walk/StepDown.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/analyze/treewalker/steps/compare/StepEndsWith.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/analyze/treewalker/steps/compare/StepDefaultIfNull.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/analyze/treewalker/steps/compare/StepIsNull.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/parser/analyze/treewalker/steps/compare/StepDefaultIfNull.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/classify/DeviceClass.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-jar/nl/basjes/parse/useragent/calculate/CalculateAgentEmail.java
* /opt/cola/permits/1202446250_1631568086.76/0/yauaa-5-20-sources-
1.1839 isobject 2.1.0
1.1839.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights...
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1840 es6-promise 4.2.8
1.1840.1 Available under license :
Copyright (c) 2014 Yehuda Katz, Tom Dale, Stefan Penner and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1841 hadoop 3.2.2
1.1841.1 Available under license :
Copyright (c) 2003-2014 Christopher M. Kohlhoff (chris at kohlhoff dot com)

Distributed under the Boost Software License, Version 1.0. (See accompanying file LICENSE_1_0.txt or copy at http://www.boost.org/LICENSE_1_0.txt)
Use of this software is granted under one of the following two licenses, to be chosen freely by the user.


Copyright (c) 2006, 2007 Marcin Kalicinski

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2. The MIT License

Copyright (c) 2006, 2007 Marcin Kalicinski

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
   2. You must cause any modified files to carry prominent notices stating that You changed the files; and
   3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
   4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

   5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any
additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of
org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the
following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of
AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java),
which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-
2010 AOL Inc.

The binary distribution of this product bundles binaries of
Gson 2.2.4,
which has the following notices:

The Netty Project
=================
Please visit the Netty web site for more information:

  * http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source
Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google’s data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
  * HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
  * HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
  * HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
  * HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
  * HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4j', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.alto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===============================================================================

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.

Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:
* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices:
* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of "Java Concurrency in Practice" book annotations 1.0, which has the following notices:
* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
  Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of Jetty :: Http Utility 9.3.19., Jetty :: IO Utility 9.3.19., Jetty :: Security 9.3.19., Jetty :: Server Core 9.3.19., Jetty :: Servlet Handling 9.3.19., Jetty :: Utilities 9.3.19., Jetty :: Utilities :: Ajax, Jetty :: Webapp Application Support 9.3.19., Jetty :: XML utilities 9.3.19., which has the following notices:
* ==================================================================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
  ==================================================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and
* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2
The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----

Assorted
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies./

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common http://hadoop.apache.org/ (Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by "GCC RUntime Library Exception"

== Contributors ==
* Tatu Saloranta
  * Providing benchmark suite
* Alec Wysoker
  * Performance and memory usage improvement

The binary distribution of this product bundles binaries of Xerces2 Java Parser 2.9.1, which has the following notices:
* == NOTICE file corresponding to section 4(d) of the Apache License, ==
  == Version 2.0, in this case for the Apache Xerces Java distribution. ==

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
  - voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.
The binary distribution of this product bundles binaries of
Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0

--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting
it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was
relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.
Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of
Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library
distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5
(https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1,
which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9
which has the following notices:
* = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=============================================================================
This product includes software developed by
Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/
uriparser - RFC 3986 URI parsing library

Copyright (C) 2007, Weijia Song <songweijia@gmail.com>
Copyright (C) 2007, Sebastian Pipping <webmaster@hartwork.org>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 * Copyright (c) 2005, European Commission project OneLab under contract 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 * - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL nor the names of its contributors may be used to endorse or promote products derived from this software without specific written permission.
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008, 2009, 2010 Massachusetts Institute of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*++
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 *
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 *
 * Abstract: The main routine
 *
--*/

For src/main/native/src/org/apache/hadoop/io/compress/lz4/{lz4.h,lz4.c,lz4hc.h,lz4hc.c},

/*
LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at:
- LZ4 source repository: http://code.google.com/p/lz4/
- LZ4 public forum: https://groups.google.com/forum/#!forum/lz4c

For hadoop-common-project/hadoop-common/src/main/native/gtest

---------------------------------------------------------------------

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:

re2j 1.1

---------------------------------------------------------------------
(license)
This is a work derived from Russ Cox’s RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni
(https://github.com/fusesource/leveldbjni), which is available under the
following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.
* Neither the name of FuseSource Corp. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

TI_________________S SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===============

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
em-helpers v0.5.13
em-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

---------------------------------------------------------------------
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)
--------------------------------------------------------------------------------
Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js
--------------------------------------------------------------------------------
The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js

The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
oj! Algorithms - version 43.0


This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are
externally maintained libraries used by this software which have their
own licenses; we recommend you read them, as their terms may differ from
the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js
--------------------------------------------------------------------------------
Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

D3 is available under a 3-clause BSD license. For details, see:
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
HSQLDB Database 2.3.4

(HSQL License)
"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ""AS IS"" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:
Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the
Hypersonic SQL Group."

The binary distribution of this product bundles these dependencies under the
following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
1. Definitions.

1.1.Contributormeans each individual or entity
that creates or contributes to the creation of
Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or
1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications,
as Covered Software and/or as part of a Larger Work; and

(b) under Patent
Claims infringed by the making, using, or selling of Modifications made by that
Contributor either alone and/or in combination with its Contributor Version (or
portions of such combination), to make, use, sell, offer for sale, have made,
and/or otherwise dispose of: (1) Modifications made by that Contributor (or
portions thereof); and (2) the combination of Modifications made by that
Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are
effective on the date Contributor first distributes or otherwise makes the
Modifications available to a third party.

(d) Notwithstanding Section 2.2(b)
above, no patent license is granted: (1) for any code that Contributor has
deleted from the Contributor Version; (2) for infringements caused by:
(i) third party modifications of Contributor Version, or (ii) the combination
of Modifications made by that Contributor with other software (except as part of
the Contributor Version) or other devices; or (3) under Patent Claims infringed
by Covered Software in the absence of Modifications made by that
Contributor.

3. Distribution Obligations.

3.1. Availability of Source
Code. Any Covered Software that You distribute or otherwise make available in
Executable form must also be made available in Source Code form and that Source
Code form must be distributed only under the terms of this License. You must
include a copy of this License with every copy of the Source Code form of the
Covered Software You distribute or otherwise make available. You must inform
recipients of any such Covered Software in Executable form as to how they can
obtain such Covered Software in Source Code form in a reasonable manner on or
through a medium customarily used for software exchange.

3.2.
Modifications. The Modifications that You create or to which You contribute are
governed by the terms of this License. You represent that You believe Your
Modifications are Your original creation(s) and/or You have sufficient rights to
grant the rights conveyed by this License.

3.3. Required Notices. You must
include a notice in each of Your Modifications that identifies You as the
Contributor of the Modification. You may not remove or alter any copyright,
patent or trademark notices contained within the Covered Software, or any
notices of licensing or any descriptive text giving attribution to any
Contributor or the Initial Developer.
3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered
Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.
6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International
Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
grizzly-http 2.2.21
grizzly-http-server 2.2.21
grizzly-http-servlet 2.2.21
grizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
   (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
   (b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have
made, use, practice, sell, and offer for sale, and/or otherwise dispose of the
Original Software (or portions thereof).
(c) The licenses granted in Sections
2.1(a) and (b) are effective on the date Initial Developer first distributes or
otherwise makes the Original Software available to a third party under the terms
of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is
granted: (1) for code that You delete from the Original Software, or (2) for
infringements caused by: (i) the modification of the Original Software, or (ii)
the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and
subject to third party intellectual property claims, each Contributor hereby
grants You a world-wide, royalty-free, non-exclusive license:
(a) under
intellectual property rights (other than patent or trademark) Licensable by
Contributor to use, reproduce, modify, display, perform, sublicense and
distribute the Modifications created by such Contributor (or portions thereof),
either on an unmodified basis, with other Modifications, as Covered Software
and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the
making, using, or selling of Modifications made by that Contributor either alone
and/or in combination with its Contributor Version (or portions of such
combination), to make, use, sell, offer for sale, have made, and/or otherwise
dispose of: (1) Modifications made by that Contributor (or portions thereof);
and (2) the combination of Modifications made by that Contributor with its
Contributor Version (or portions of such combination).
(c) The licenses granted
in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications available to a third
party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2)
for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with
other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or
otherwise make available in Executable form must also be made available in
Source Code form and that Source Code form must be distributed only under the
terms of this License. You must include a copy of this License with every copy
of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.
4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory
judgment actions) against Initial Developer or a Contributor (the Initial
Developer or Contributor against whom You assert such claim is referred to as
Participant) alleging that the Participant Software (meaning the
Contributor Version where the Participant is a Contributor or the Original
Software where the Participant is the Initial Developer) directly or indirectly
infringes any patent, then any and all rights granted directly or indirectly to
You by such Participant, the Initial Developer (if the Initial Developer is not
the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this
License shall, upon 60 days notice from Participant terminate prospectively and
automatically at the expiration of such 60 day notice period, unless if within
such 60 day period You withdraw Your claim with respect to the Participant
Software against such Participant either unilaterally or pursuant to a written
agreement with Participant.

6.3. If You assert a patent infringement claim
against Participant alleging that the Participant Software directly or
indirectly infringes any patent where such claim is resolved (such as by license
or settlement) prior to the initiation of patent infringement litigation, then
the reasonable value of the licenses granted by such Participant under Sections
2.1 or 2.2 shall be taken into account in determining the amount or value of any
payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2
above, all end user licenses that have been validly granted by You or any
distributor hereunder prior to termination (excluding licenses granted to You by
any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT
(INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL
DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR
ANY
SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT,
SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING,
WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOOGAGE, COMPUTER
FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN
IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS
LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL
INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW
PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR
LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND
LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered
Software is a commercial item, as that term is defined in 48 C.F.R. 2.101
(Oct. 1995), consisting of commercial computer software (as that term is
defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form.
This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE,
6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a
distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.
(2-clause BSD)
 Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0
(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions
"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital
performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work
if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same
terms and conditions as the license granted to You under this License. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

jmmon-runtime 2.4.1

(Mozilla Public License
Version 2.0)

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means
that the initial Contributor has attached the notice described in Exhibit B to
the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or
earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate
file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the
time of the initial grant or subsequently, any and all of the rights conveyed by
this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or
modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and
apparatus claims, in any patent Licensable by such Contributor that would be
infringed, but for the grant of the License, by the making, using, selling,
offering for sale, having made, import, or transfer of either its Contributions
or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General
Public License, Version 2.1, the GNU Affero General Public License, Version 3.0,
or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For
legal entities, You includes any entity that controls, is controlled by,
or is under common control with You. For purposes of this definition,
control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses
No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.
3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated.
(a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.
8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice
This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

/#--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the
following license:
Hbase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is
available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE
LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN
ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS
INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES
REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR
DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE
COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY

Open Source Used In DNAC Platform Antelope 6390
COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the
organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to
exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the
Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or
Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as
stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special,
incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.


Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2010 Ben Noordhuis

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013, Michael Bostock
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* The name Michael Bostock may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL MICHAEL BOSTOCK BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1842 firstline 1.3.1
1.1842.1 Available under license :
The MIT License (MIT)

Copyright (c) 2018 Alessandro Zanardi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1843 kerby-pkix 1.0.0
1.1843.1 Available under license:

Kerby PKIX Project
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1844 extsprintf 1.3.0

1.1844.1 Available under license :

Copyright (c) 2012, Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1845 fast-json-stable-stringify 2.1.0 2.1.0

1.1845.1 Available under license:

This software is released under the MIT license:

Copyright (c) 2017 Evgeny Poberezkin
Copyright (c) 2013 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1846 commons-el 5.5.23

1.1846.1 Available under license:

Apache Commons CLI
Copyright 2001-2007 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1847 batik 1.13
1.1847.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate.
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1848 is-symbol 1.0.3

1.1848.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1849 ci-info 2.0.0

1.1849.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016-2018 Thomas Watson Steen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1850 y18n 5.0.8 5.0.8

1.1850.1 Available under license:

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.1851 uri-js 4.4.0

1.1851.1 Available under license:

Copyright 2011 Gary Court. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY GARY COURT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL GARY COURT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of Gary Court.

1.1852 mv 2.1.1

1.1852.1 Available under license:

Copyright (c) 2014 Andrew Kelley

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1853 unset-value 1.0.0

1.1853.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015, 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1854 kite-data-core 1.1.0

1.1854.1 Available under license:
No license file was found, but licenses were detected in source scan.

#  
# Copyright 2013 Cloudera Inc.  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at  
#  
# http://www.apache.org/licenses/LICENSE-2.0  
#
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# org.kitesdk.data.spi.filesystem.Loader

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/META-INF/services/org.kitesdk.data.spi.Loadable
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 Cloudera.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/StorageKey.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/View.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/filesystem/MultiLevelIterator.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/AbstractDatasetRepository.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/AbstractMetadataProvider.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/Marker.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/AbstractRefinableView.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/UnknownFormatException.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/MarkerRange.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 Cloudera.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 joey.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

No license file was found, but licenses were detected in source scan.

/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/spi/filesystem/AvroFSInput.java
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2015 Cloudera Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/spi/filesystem/SignalManager.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 Cloudera.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 Cloudera, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1.jar/org/kitesdk/data/spi/DataModelUtil.java
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2014 Cloudera Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1.jar/org/kitesdk/data/spi/FilteredRecordReader.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1.jar/org/kitesdk/data/spi/PartitionStrategyParser.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1.jar/org/kitesdk/data/spi/ColumnMappingParser.java
/*
 * Copyright 2015 Cloudera Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 Cloudera Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 Cloudera Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
   * /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/filesystem/FileSystemViewKeyInputFormat.java
   * /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/DescriptorUtil.java

No license file was found, but licenses were detected in source scan.
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2013 Cloudera Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/filesystem/MultiFileDatasetReader.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/SchemaValidationUtil.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/filesystem/ParquetAppender.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/package-info.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/partition/IntRangeFieldPartitioner.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/partition/HashFieldPartitioner.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/PartitionListener.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/partition/MinuteFieldPartitioner.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/DatasetIOException.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/ColumnMapping.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/partition/PartitionFunctions.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/FieldMapping.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/partition/ListFieldPartitioner.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/package-info.java
jar/org/kitesdk/data/spi/PartitionKey.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/FieldPartitioner.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/Key.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/TypeNotFoundException.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/partition/YearFieldPartitioner.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/DatasetRepository.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/Formats.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/DatasetDescriptor.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/filesystem/AccessorImpl.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/filesystem/ParquetFileSystemDatasetReader.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/spi/filesystem/SchemaManager.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-jar/org/kitesdk/data/impl/package-info.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 Cloudera Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/spi/PartitionedDataset.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2013 Cloudera Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/spi/filesystem/FileSystemUtil.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/spi/TemporaryDatasetRepositoryAccessor.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/spi/TemporaryFileSystemDatasetRepository.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/spi/SchemaUtil.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/spi/Pair.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
jar/org/kitesdk/data/Flushable.java
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 Cloudera.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446424_1631568199.51/0/kite-data-core-1-1-0-sources-1-
1.1855 is-my-ip-valid 1.0.0

1.1855.1 Available under license :

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1856 excel-streaming-reader 3.0.4

1.1856.1 Available under license :

No license file was found, but licenses were detected in source scan.

/* ====================================================================
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
==================================================================== */
1.1857 drill-storage-hive-core 1.19.0

1.1857.1 Available under license:

Drill : Contrib : Storage : Hive : Core
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1858 ajv 5.5.2 5.5.2
1.1858.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1859 babel-traverse 6.26.0
1.1859.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>>> is protected by copyright and is
being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR

INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from
THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1860 deep-sort-object 1.0.2 1.0.2

1.1860.1 Available under license:

Copyright (c) 2014 IndigoUnited

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1861 arr-diff 2.0.0

1.1861.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1862 jersey 2.25.1
1.1862.1 Available under license:

/*
 * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
 * 
 * Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
 * 
 * The contents of this file are subject to the terms of either the GNU
 * General Public License Version 2 only ("GPL") or the Common Development
 * and Distribution License("CDDL") (collectively, the "License"). You
 * may not use this file except in compliance with the License. You can
 * obtain a copy of the License at
 * http://glassfish.java.net/public/CDDL+GPL_1_1.html
 * or packager/legal/LICENSE.txt. See the License for the specific
 * language governing permissions and limitations under the License.
 * 
 * When distributing the software, include this License Header Notice in each
 * file and include the License file at packager/legal/LICENSE.txt.
 * 
 * GPL Classpath Exception:
 * Oracle designates this particular file as subject to the "Classpath"
 * exception as provided by Oracle in the GPL Version 2 section of the License
 * file that accompanied this code.
 * 
 * Modifications:
 * If applicable, add the following below the License Header, with the fields
 * enclosed by brackets [ ] replaced by your own identifying information:
 * "Portions Copyright [year] [name of copyright owner]"
 * 
 * Contributor(s):
 * If you wish your version of this file to be governed by only the CDDL or
 * only the GPL Version 2, indicate your decision by adding "[Contributor]
 * elects to include this software in this distribution under the [CDDL or GPL
 * Version 2] license." If you don't indicate a single choice of license, a
 * recipient has the option to distribute your version of this file under
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has
deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or
such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms
herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

----------
NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.
When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to
achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2
Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

DO NOT TRANSLATE OR LOCALIZE.

%% The following software may be included in this product: ASM
Use of any of this software is governed by the terms of the license below:

Copyright (c) 2000-2005 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
Open Source Used In DNAC Platform Antelope 6452

CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

%%% The following software may be included in this product: Jettison
Use of any of this software is governed by the terms of the license below:

Copyright 2006 Envoi Solutions LLC

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

------------------------------------------------------------------------
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and

2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

---

The following software may be included in this product: Jackson

Use of any of this software is governed by the terms of the license below:

Jackson is dual-licensed under two alternative popular Open Source licenses: Apache (AL 2.0) and Gnu Lesser GPL (LGPL 2.1). You choose one or the other, as necessary (if you want to redistribute the code for use, you do not need license), and abide by the license rules as defined by the respective license agreement (and only that one).

Apartment License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
1. You must give any other recipients of the Work or Derivative Works a copy of this License; and

2. You must cause any modified files to carry prominent notices stating that You changed the files; and

3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may
choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by
brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and
description of purpose be included on the same "printed page" as the copyright notice for easier identification within
third-party archives.

Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you
may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
limitations under the License.

1.1863 groovy-sql 3.0.8
1.1863.1 Available under license :

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.

Found in path(s):
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/package.html
/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/ResultSetOutParameter.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/ExtractIndexAndSql.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/Sql.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/ResultSetMetaDataWrapper.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/GroovyResultSetExtension.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/BatchingPreparedStatementWrapper.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/CallResultSet.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/DataSet.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/GroovyRowResult.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/GroovyResultSet.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/SqlWithParams.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/ExpandedVariable.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/GroovyResultSetProxy.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/BatchingPreparedStatementWrapper.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/InOutParameter.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/InParameter.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/groovy/sql/SqlWhereVisitor.java
* /opt/cola/permits/1218311877_1634925046.69/0/groovy-sql-3-0-8-sources-jar/org/apache/groovy/sql/extensions/SqlExtensions.java

No license file was found, but licenses were detected in source scan.
1.1864 minimatch 3.0.4 3.0.4
1.1864.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1865 css-what 2.1.3
1.1865.1 Available under license:
Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS.
1.1866 httpasyncclient 4.1.4

1.1866.1 Available under license:

Apache HttpAsyncClient
Copyright 2010-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1867 cliui 3.2.0
1.1867.1 Available under license :

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1868 moment 2.27.0
1.1868.1 Available under license:
Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1869 firstline 1.3.1 1.3.1

1.1869.1 Available under license:
The MIT License (MIT)

Copyright (c) 2018 Alessandro Zanardi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1870 react-copy-to-clipboard 5.0.2

1.1870.1 Available under license:

MIT License

Copyright (c) 2017 Mauro Bringolf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2016 John Hiesey

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
Copyright (c) Feross Aboukhadijeh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2008, Fair Oaks Labs, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Fair Oaks Labs, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
Copyright JS Foundation and other contributors, https://js.foundation
Copyright (C) 2012-2013 Yusuke Suzuki (twitter: @Constellation) and other contributors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A Particular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

$/LicenseInfo$
// Original can be found at:
//   https://bitbucket.org/lindenlab/llsd
// Modifications by Joshua Bell inexorabletash@gmail.com
//   https://github.com/inexorabletash/polyfill

// ES3/ES5 implementation of the Khronos Typed Array Specification
// Ref: http://www.khronos.org/registry/typedarray/specs/latest/
// Date: 2011-02-01
// Variations:
// * Allows typed_array.get/set() as alias for subscripts (typed_array[])  
Copyright (c) 2013 Dulin Marat

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2017 CoderPuppy

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
MIT License

Copyright (c) 2014-2018 Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
ISC License

Copyright (c) npm, Inc.

Permission to use, copy, modify, and/or distribute this software for
any purpose with or without fee is hereby granted, provided that the
above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE COPYRIGHT HOLDER DISCLAIMS
ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE
COPYRIGHT HOLDER BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE
USE OR PERFORMANCE OF THIS SOFTWARE.
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2018 Kilian Valkhof

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
The MIT License

Copyright (c) 2012-2014 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schlueter
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License (MIT)

Copyright (c) 2012 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
(The MIT License)

Copyright (c) 2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/*! *****************************************************************************
Copyright (c) Microsoft Corporation. All rights reserved.
Licensed under the Apache License, Version 2.0 (the "License"); you may not use
this file except in compliance with the License. You may obtain a copy of the
License at http://www.apache.org/licenses/LICENSE-2.0

THIS CODE IS PROVIDED ON AN *AS IS* BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED
WARRANTIES OR CONDITIONS OF TITLE, FITNESS FOR A PARTICULAR PURPOSE,
MERCHANTABILITY OR NON-INFRINGEMENT.

See the Apache Version 2.0 License for specific language governing permissions
and limitations under the License.
***************************************************************************/

The MIT License (MIT)

Copyright (c) 2011 Troy Goode <troygoode@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014 Tim Schaub

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
(The MIT License)

Copyright (c) 2012-2013 TJ Holowaychuk
Copyright (c) 2015 Andreas Lubbe
Copyright (c) 2015 Tiancheng "Timothy" Gu

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

You may use the Forge project under the terms of either the BSD License or the
GNU General Public License (GPL) Version 2.

The BSD License is recommended for most projects. It is simple and easy to
understand and it places almost no restrictions on what you can do with the
Forge project.

If the GPL suits your project better you are also free to use Forge under
that license.

You don't have to do anything special to choose one license or the other and
you don't have to notify anyone which license you are using. You are free to
use this project in commercial projects as long as the copyright header is
left intact.

If you are a commercial entity and use this set of libraries in your
commercial software then reasonable payment to Digital Bazaar, if you can afford it, is not required but is expected and would be appreciated. If this library saves you time, then it's saving you money. The cost of developing the Forge software was on the order of several hundred hours and tens of thousands of dollars. We are attempting to strike a balance between helping the development community while not being taken advantage of by lucrative commercial entities for our efforts.

New BSD License (3-clause)
Copyright (c) 2010, Digital Bazaar, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Digital Bazaar, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL DIGITAL BAZAAR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices
      stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in
      whole or in part contains or is derived from the Program or any
      part thereof, to be licensed as a whole at no charge to all third
      parties under the terms of this License.

   c) If the modified program normally reads commands interactively
      when run, you must cause it, when started running for such
      interactive use in the most ordinary way, to print or display an
      announcement including an appropriate copyright notice and a
      notice that there is no warranty (or else, saying that you provide
      a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

ISC License

Copyright (c) 2018, Andrea Giammarchi, @WebReflection

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 LevelUP contributors

*LevelUP contributors listed at <https://github.com/rvagg/node-levelup#contributors>*
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2019 Luís Couto <hello@luiscouto.pt>

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2017 Sergey Rubanov (https://github.com/chicoxyzzy)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014, 2015, 2016, 2017, 2018 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2014 Nathan Rajlich <nathan@tootallnate.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) Kevin Mrtensson <kevinmartensson@gmail.com> (github.com/kevva)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2013-2017 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 crypto-browserify

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-present, Brian Woodward.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2016 Nik Butenko

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2013 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) Kevin Martensson <kevinmartensson@gmail.com> (github.com/kevva)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2015-2018 Google, Inc., Netflix, Inc., Microsoft Corp. and contributors
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
The MIT License (MIT)

Copyright (c) 2015 Roy Riojas

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
The MIT License (MIT)

Copyright (c) 2011 Einar Otto Stangvik <einaros@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright 2014 Andrey Sitnik <andrey@sitnik.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017 sudodoki <smd.deluzion@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
MIT License

Copyright (c) 2017 webpack-contrib

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Toposort - Topological sorting for node.js
Copyright (c) 2012 by Marcel Klehr <mklehr@gmx.net>
MIT LICENSE

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
The MIT License (MIT)

Copyright 2013 Andrey Sitnik <andrey@sitnik.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

MIT License (MIT)

Copyright (c) 2013 [Ramesh Nair](http://www.hiddentao.com/)

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017 Gustaf Andersson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 Tim Caswell (https://github.com/creationix) and other contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
Some files from the tests folder are from joyent/node and mscedex/io.js, a fork of nodejs/io.js:

- tests/iojs/test-http-parser-durability.js


---

Copyright io.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

- tests/fixtures/*
tests/parallel/*
tests/testpy/*
tests/common.js
tests/test.py
tests/utils.py

These files are from https://github.com/nodejs/node with changes by Jan Schr (jscissr).

Node.js is licensed for use as follows:
Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014 Forbes Lindesay
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2014 Marten de Vries

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2011-2016 Heather Arthur <fayearthur@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
MIT License

Copyright (c) 2012 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com
Copyright (c) 2016 Douglas Christopher Wilson doug@somethingdoug.com
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2013-2018 Ben Alman <cowboy@rj3.net>, Blaine Bublitz <blaine.bublitz@gmail.com>, and Eric Schoffstall <yo@contra.io>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UglifyJS is released under the BSD license:

Copyright 2012-2018 (c) Mihai Bazon <mihai.bazon@gmail.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This package contains code originally written by Isaac Z. Schlueter.

Used with permission.

Copyright (c) Meryn Stol ("Author")
All rights reserved.

The BSD License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2012 - 2015 Tobias Koppers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2013 JosF. Romaniello

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright 2015 Kiko Beats

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015, Wes Todd

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) Bogdan Chadkin <trysound@yandex.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Javier Blanco

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Espree
Copyright JS Foundation and other contributors, https://js.foundation

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2017 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Andres Suarez <zertosh@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The ISC License

Copyright (c) npm, Inc. and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
The MIT License (MIT)

Copyright (c) 2016 Kevin Gravier

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
The MIT License

Copyright (c) Tobias Koppers @sokra

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ISC License (ISC)
Copyright 2018 Glen Maddern

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)
Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2016-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
Copyright 2017 JS Foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014, Rebecca Turner <me@rebecca.org>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License

Copyright (c) 2013 James Halliday (mail@substack.net)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (C) 2012 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 JD Ballard

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015, 2017-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2017 Calvin Metcalf, Fedor Indutny & contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# spdx-license-ids


A list of [SPDX license](https://spdx.org/licenses/) identifiers

## Installation

[Download JSON directly](https://raw.githubusercontent.com/shinnn/spdx-license-ids/master/index.json), or [use](https://docs.npmjs.com/cli/install) [npm](https://docs.npmjs.com/about-npm):

```bash
npm install spdx-license-ids
```

## [Node.js](https://nodejs.org/) API

### require('spdx-license-ids')

Type: `string[]`

All license IDs except for the currently deprecated ones.

```javascript
const ids = require('spdx-license-ids');
//=> ['0BSD', 'AAL', 'ADSL', 'AFL-1.1', 'AFL-1.2', 'AFL-2.0', 'AFL-2.1', 'AFL-3.0', 'AGPL-1.0-only', ...]

ids.includes('BSD-3-Clause'); //=> true
ids.includes('CC-BY-1.0'); //=> true

ids.includes('GPL-3.0'); //=> false
```

### require('spdx-license-ids/deprecated')

Type: `string[]`

```javascript
```
Deprecated license IDs.

```javascript
const deprecatedIds = require('spdx-license-ids/deprecated');
//=> ['AGPL-1.0', 'AGPL-3.0', 'GFDL-1.1', 'GFDL-1.2', 'GFDL-1.3', 'GPL-1.0', 'GPL-2.0', ...]

deprecatedIds.includes('BSD-3-Clause'); //=> false

deprecatedIds.includes('CC-BY-1.0'); //=> false

deprecatedIds.includes('GPL-3.0'); //=> true
```

## License

[Creative Commons Zero v1.0 Universal](https://creativecommons.org/publicdomain/zero/1.0/deed)
The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014 Nathan LaFreniere and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote
products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*   *   *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors

Copyright 2015 Calvin Metcalf

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

MIT License

Copyright (c) 2012, 2013, 2014 James Halliday <mail@substack.net>, 2009 Thomas Robinson <280north.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2013 Ryan Day

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Zeit, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-2018 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright JS Foundation and other contributors, https://js.foundation/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2016 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
The MIT License (MIT)

Copyright (c) 2014 Daniel Cousens

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Copyright 2014 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

* Neither the name of the Yahoo! Inc. nor the
  names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License, Version 2.0

Copyright (c) 2011 Dominic Tarr

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Copyright (c) 2014-2016 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Fork, Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 Simon Boudrias

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2015 Devon Govett <devongovett@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This project contains parts of Node.js. Node.js is licensed for use as follows:

----

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

----

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

----

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

License

(The MIT License)

Copyright (c) 2014 The cheeriojs contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ‘Software’), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Ethan Cohen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2011 Alexander Shtuchkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2012 Ben Ripkens http://bripkens.de

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2017 crypto-browserify contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Jon Harris

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Calvin Metcalf and browserify-sign contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
The MIT License (MIT)

Copyright (c) Ben Drucker <bvdrucker@gmail.com> (bendrucker.me)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Jed Watson <jed.watson@me.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{ }" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
The MIT Licence (Expat).

Copyright (c) 2018 Michael Mclaughlin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License

Copyright (c) 2013 Max Ogden

Permission is hereby granted, free of charge,
to any person obtaining a copy of this software and
associated documentation files (the "Software"), to
deal in the Software without restriction, including
without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom
the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice
shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2013-2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

MIT License

Copyright (c) Sam Verschueren <sam.verschueren@gmail.com> (github.com/SamVerschueren)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
Open Source Used In DNAC Platform Antelope  6543

(.an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2016 Matthew Davidson

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

The MIT License

Copyright (c) 2015 Kyle E. Mitchell & other authors listed in AUTHORS

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
Copyright (c) 2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) Sam Verschueren <sam.verschueren@gmail.com> (github.com/SamVerschueren)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (C) 2012-2018 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

node-http-proxy

Copyright (c) 2010-2016 Charlie Robbins, Jarrett Cruger & the Contributors.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, 
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND 
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE 
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION 
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION 
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2010-2018 Caolan McMahon

Permission is hereby granted, free of charge, to any person obtaining a copy 
of this software and associated documentation files (the "Software"), to deal 
in the Software without restriction, including without limitation the rights 
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom the Software is 
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in 
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR 
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, 
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE 
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER 
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, 
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN 
THE SOFTWARE.

Copyright (c) 2012 Charlie Robbins

Permission is hereby granted, free of charge, to any person obtaining 
a copy of this software and associated documentation files (the 
"Software"), to deal in the Software without restriction, including 
without limitation the rights to use, copy, modify, merge, publish, 
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2018 Toru Nagashima

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013, Dominic Tarr
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2013 Mikola Lysenko

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2019 Andres Suarez

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2012 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2016-2017 Douglas Christopher Wilson <doug@somethingdoug.com>
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017 JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright 2014-2019 Benjamin Tan <https://bnjmnt4n.now.sh/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Nikita Gusakov

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

At your option you may choose either of the following licenses:

* The MIT License (MIT)
* The Apache License 2.0 (Apache-2.0)

The MIT License (MIT)

Copyright (c) 2015 AJ O'Neal

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2015 AJ ONeal

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
The MIT License (MIT)
Copyright (c) 2017 Samuel Reed <samuel.trace.reed@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (C) 2018 Dmitry Shirokov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2011 Dominic Tarr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2010-2019 James Coglan

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the
specific language governing permissions and limitations under the License.
The MIT License (MIT)

Copyright (c) 2013 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This software is released under the MIT license:

Copyright (c) 2012-2015 Thorsten Lorenz

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright npm, Inc

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above
Copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

(The MIT License)

Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2018 Sven Sauleau <sven@sauleau.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (C) 2017 JakubPawlowicz.com

Permission is hereby granted, free of charge, to any person obtaining a copy
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (C) 2011-2015 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2012-2014 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2015, Rebecca Turner <me@re-becca.org>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
(The MIT License)

Copyright (c) 2014 Component contributors <dev@component.io>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
# timers-browserify

This project uses the [MIT](http://jryans.mit-license.org/) license:

Copyright 2012 J. Ryan Stinnett <jryans@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the Software),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

# lib/node

The `lib/node` directory borrows files from joyent/node which uses the following license:

Copyright Joyent, Inc. and other Node contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

Copyright (c) 2017, crypto-browserify contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (c) 2013-2018, Viacheslav Lotsmanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2016 Jordan Harband
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Parts of the code are derived from various open source projects.

For code derived from Socket.IO by Guillermo Rauch see https://github.com/LearnBoost/socket.io/tree/0.6.17#readme.

All other code is released on MIT license, see LICENSE.
The MIT License (MIT)

Copyright (c) 2016 Thomas Watson Steen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Minizlib was created by Isaac Z. Schlueter.
It is a derivative work of the Node.js project.
Copyright Isaac Z. Schlueter and Contributors
Copyright Node.js contributors. All rights reserved.
Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

(The MIT License)

Copyright (c) 2011 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2012 Tobias Koppers
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015, Ilya Radchenko <ilya@burstcreations.com>

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
(The MIT License)

Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2018 NAN contributors

* NAN contributors listed at <https://github.com/nodejs/nan#contributors>*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions
a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.
c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography;
a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a
modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You
must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor’s copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author’s honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author’s honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the
applicable national law, to enable You to reasonably exercise Your
right under Section 3(b) of this License (right to make Adaptations)
but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR
OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY
KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE,
INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF
LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS,
WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION
OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE
LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR
ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES
ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS
BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate
automatically upon any breach by You of the terms of this License.
Individuals or entities who have received Adaptations or Collections
from You under this License, however, will not have their licenses
terminated provided such individuals or entities remain in full
compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will
survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is
perpetual (for the duration of the applicable copyright in the Work).
Notwithstanding the above, Licensor reserves the right to release the
Work under different license terms or to stop distributing the Work at
any time; provided, however that any such election will not serve to
withdraw this License (or any other license that has been, or is
required to be, granted under the terms of this License), and this
License will continue in full force and effect unless terminated as
stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection,
the Licensor offers to the recipient a license to the Work on the same
terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor
offers to the recipient a license to the original Work on the same
terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

---

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in
compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at http://creativecommons.org/.

(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

## Glob Logo

Glob’s logo created by Tanya Brassie <http://tanyabrassie.com/>, licensed under a Creative Commons Attribution-ShareAlike 4.0 International License
https://creativecommons.org/licenses/by-sa/4.0/

The MIT License

Copyright (c) EventSource GitHub organisation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) Kevin Mrtensson <kevinmartensson@gmail.com> (github.com/kevva)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Kirill Fomichev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2013-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015-2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2019 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent
infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

The ISC License

Copyright (c) 2015, 2019 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-2018 Suguru Motegi
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2015 Glen Maddern

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (C) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2012 Barnesandnoble.com, llc, Donavon West, and Domenic Denicola

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT Expat Licence.

Copyright (c) 2012 Michael Mclaughlin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2016, 2018 Linus Unnebeck

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Ben Mosher
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013 Thiago de Arruda

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright npm, Inc

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.
The MIT License (MIT)

Copyright (c) 2018 Made With MOXY Lda <hello@moxy.studio>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
Copyright 2014-present Olivier Lalonde <olalonde@gmail.com>, James Talmage <james@talmage.io>, Ruben
Verborgh

Open Source Used In DNAC Platform Antelope 6600
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2011:
Tim Koschitzki (tim@debuggable.com)
Felix Geisendirfer (felix@debuggable.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
The MIT License

Copyright (c) 2017 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (C) 2013 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The MIT License (MIT)
Copyright (c) 2014 Simon Boudrias

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright 2015 Mark Dalgleish <mark.john.dalgleish@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Julian Viereck and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2011-2017 JP Richardson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
The MIT License (MIT)

Copyright (c) 2014 Arnout Kazemier

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
The MIT License (MIT)

Copyright (c) 2016-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This software is dual-licensed under the ISC and MIT licenses.
You may use this software under EITHER of the following licenses.

--------

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

--------

Copyright Isaac Z. Schlueter and Contributors
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2013 Aria Minaei

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (C) 2014-2017 by Vitaly Puzrin and Andrei Tuputcyn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

MIT License

Copyright (c) James Talmage <james@talmage.io> (github.com/jamestalmage)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (C) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
Attribution 4.0 International

Creative Commons Corporation ("Creative Commons") is not a law firm and
does not provide legal services or legal advice. Distribution of
Creative Commons public licenses does not create a lawyer-client or
other relationship. Creative Commons makes its licenses and related
information available on an "as-is" basis. Creative Commons gives no
warranties regarding its licenses, any material licensed under their
terms and conditions, or any related information. Creative Commons
disclaims all liability for damages resulting from their use to the
fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and
conditions that creators and other rights holders may use to share
original works of authorship and other material subject to copyright
and certain other rights specified in the public license below. The
following considerations are for informational purposes only, are not
exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are
intended for use by those authorized to give the public
permission to use material in ways otherwise restricted by
copyright and certain other rights. Our licenses are
irrevocable. Licensors should read and understand the terms
and conditions of the license they choose before applying it.
Licensors should also secure all rights necessary before
applying our licenses so that the public can reuse the
material as expected. Licensors should clearly mark any
material not subject to the license. This includes other CC-
licensed material, or material used under an exception or
limitation to copyright. More considerations for licensors:
wiki.creativecommons.org/Considerations_for_licensors

Considerations for the public: By using one of our public
licenses, a licensor grants the public permission to use the
licensed material under specified terms and conditions. If
the licensor's permission is not necessary for any reason—for
example, because of any applicable exception or limitation to
copyright--then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public:

wiki.creativecommons.org/Considerations_for_licensees

=======================================================================

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar
Rights.

d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

i. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free,
non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

a. reproduce and Share the Licensed Material, in whole or in part; and

b. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a) (4) never produces Adapted Material.

5. Downstream recipients.

a. Offer from the Licensor -- Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

b. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as
b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

   a. retain the following if it is supplied by the Licensor with the Licensed Material:

      i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

      ii. a copyright notice;

      iii. a notice that refers to this Public License;

      iv. a notice that refers to the disclaimer of warranties;
v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.
Section 5 -- Disclaimer of Warranties and Limitation of Liability.

a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations.
of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the
Licensed Material under separate terms or conditions or stop
distributing the Licensed Material at any time; however, doing so
will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public
License.

Section 7 -- Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different
terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the
Licensed Material not stated herein are separate from and
independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

a. For the avoidance of doubt, this Public License does not, and
shall not be interpreted to, reduce, limit, restrict, or impose
conditions on any use of the Licensed Material that could lawfully
be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is
deemed unenforceable, it shall be automatically reformed to the
minimum extent necessary to make it enforceable. If the provision
cannot be reformed, it shall be severed from this Public License
without affecting the enforceability of the remaining terms and
conditions.

c. No term or condition of this Public License will be waived and no
failure to comply consented to unless expressly agreed to by the
Licensor.

d. Nothing in this Public License constitutes or may be interpreted
as a limitation upon, or waiver of, any privileges and immunities
that apply to the Licensor or You, including from the legal
processes of any jurisdiction or authority.

=======================================================================

Creative Commons is not a party to its public
licenses. Notwithstanding, Creative Commons may elect to apply one of
to material it publishes and in those instances
its public licenses to material it publishes and in those instances
will be considered the Licensor. The text of the Creative Commons
will be considered the Licensor. The text of the Creative Commons
public licenses is dedicated to the public domain under the CC0 Public
public licenses is dedicated to the public domain under the CC0 Public
Domain Dedication. Except for the limited purpose of indicating that
Domain Dedication. Except for the limited purpose of indicating that
material is shared under a Creative Commons public license or as
material is shared under a Creative Commons public license or as
otherwise permitted by the Creative Commons policies published at
otherwise permitted by the Creative Commons policies published at
creativec.om/policies, Creative Commons does not authorize the
creativec.om/policies, Creative Commons does not authorize the
use of the trademark "Creative Commons" or any other trademark or logo
use of the trademark "Creative Commons" or any other trademark or logo
of Creative Commons without its prior written consent including,
of Creative Commons without its prior written consent including,
without limitation, in connection with any unauthorized modifications
without limitation, in connection with any unauthorized modifications
to any of its public licenses or any other arrangements,
to any of its public licenses or any other arrangements,
understandings, or agreements concerning use of licensed material. For
understandings, or agreements concerning use of licensed material. For
the avoidance of doubt, this paragraph does not form part of the
the avoidance of doubt, this paragraph does not form part of the
public licenses.
public licenses.

Creative Commons may be contacted at creativecommons.org.
Creative Commons may be contacted at creativecommons.org.
The MIT License (MIT)
The MIT License (MIT)

Copyright (c) 2015 Daniel Cousens
Copyright (c) 2015 Daniel Cousens

Permission is hereby granted, free of charge, to any person obtaining a copy
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
SOFTWARE.
The MIT License (MIT)
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.
Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2011-2018 The sockjs-client Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright 2015 Djordje Lukic, Junle Li

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(MIT License)

Copyright (c) 2012-2018 Aseem Kishore, and [others].

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

[others]: https://github.com/json5/json5/contributors
The MIT License (MIT)

Copyright (c) 2014 Evan Wallace

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)
Copyright (c) 2015 Dmitry Ivanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License
Copyright (c) 2014 Petka Antonov
2015 Sindre Sorhus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (C) 2011-2017 whitequark <whitequark@whitequark.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
MIT License
--------

Copyright (C) 2010-2014 Philipp Dunkel

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The ISC License

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
 LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Copyright (c) 2015, Rebecca Turner

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

# The MIT License

Copyright (c) 2014-2017 James Coglan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013 Fog Creek Software

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the names of the Mozilla Foundation nor the names of project
  contributors may be used to endorse or promote products derived from this
  software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<h1>License</h1>

Unless stated otherwise all works are:

<ol>
  <li>Copyright © 2013+ <a href="http://bevry.me">Bevry Pty Ltd</a></li>
</ol>

and licensed under:

<ol>
  <li><a href="http://spdx.org/licenses/MIT.html">MIT License</a></li>
</ol>

<h2>MIT License</h2>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

<!-- /LICENSEFILE -->

Copyright (c) silverwind
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
MIT License

Copyright (c) 2018 Nikita Skovoroda <chalkerx@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012-2015 Kit Cambridge.
http://kitcambridge.be/

Copyright (c) 2013-2015 Benjamin Tan.
https://d10.github.io/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2015, 2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright contributors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2015, Rebecca Turner

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
ISC License (ISC)
Copyright 2018 Stefan Penner

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
Copyright (C) 2012-2017 by Ingvar Stepanyan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014 Hugh Kennedy

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
Copyright (c) 2013 Dominic Tarr

Permission is hereby granted, free of charge,
to any person obtaining a copy of this software and
associated documentation files (the "Software"), to
deal in the Software without restriction, including
without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom
the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice
shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright Mathias Bynens <http://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
MIT License

Copyright (c) 2012-2016 Aseem Kishore, and [others](https://github.com/aseemk/json5/contributors).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013 Kael Zhang <i@kael.me>, contributors
http://kael.me/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2010 James Halliday (mail@substack.net)
Modified work Copyright 2014 Contributors (ben@npmjs.com)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2018, Gajus Kuizinas (http://gajus.com/)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Gajus Kuizinas (http://gajus.com/) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANUARY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016, 2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
Copyright (c) 2013 Julian Gruber <julian@juliangruber.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Thomas Watson Steen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Aria Minaei

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# The MIT License (MIT)

**Copyright (c) Rod Vagg (the "Original Author") and additional contributors**

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2014-2018 Tyler Kellen <tyler@sleekcode.net>, Blaine Bublitz <blaine.bublitz@gmail.com>, and
Eric Schoffstall <yo@contra.io>

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
MIT License
Copyright (c) DC <threedeecce@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.
Copyright 2012 Irakli Gozalishvili. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

The MIT License (MIT)

Copyright (c) 2013-2018 Petka Antonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2012 Ryan Day

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) 2014 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Dual licensed under WTFPL and MIT:

---

Copyright 2013-2016 Domenic Denicola <d@domenic.me>

This work is free. You can redistribute it and/or modify it under the
terms of the Do What The Fuck You Want To Public License, Version 2,
as published by Sam Hocevar. See below for more details.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified
copies of this license document, and changing it is allowed as long
as the name is changed.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
0. You just DO WHAT THE FUCK YOU WANT TO.

---

The MIT License (MIT)

Copyright 2013-2016 Domenic Denicola <d@domic.me>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Doctrine
Copyright jQuery Foundation and other contributors, https://jquery.org/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2017 createECDH contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (c) 2010-2016 Robert Kieffer and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2012 The Obvious Corporation.
http://obvious.com/

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-------------------------------------------------------------------------

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Copyright 2010, 2011, Chris Winberry <chris@winberry.net>. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter, Ben Noordhuis, and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

(MIT)

Copyright (c) 2013 Julian Gruber &lt;julian@juliangruber.com&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Node.js is licensed for use as follows:

=====

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Brian Woodward

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2018 Tobias Reich

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Yannick Croissant

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015-2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
MIT License

Copyright (c) 2016 depcheck

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)
=====================

Copyright (c) 2014 Rod Vagg
----------------------------

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013, Joel Feenstra
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the ESQuery nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JOEL FEENSTRA BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2014-2017 browserify-aes contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors
Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

This library bundles a version of the `fs.realpath` and `fs.realpathSync` methods from Node.js v0.10 under the terms of the Node.js MIT license.

Node's license follows, also included at the header of `old.js` which contains the licensed code:

Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
(The MIT License)

Copyright (c) 2013 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2012 Airbnb

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-2016 Thomas Watson Steen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
MIT License

Copyright (c) 2017 crypto-browserify

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Jeremiah Senkpiel <fishrock123@rocketmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, Torts OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (C) 2011 VMware, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2013 Roman Shtylman <shtylman@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# This is the MIT license
Copyright (c) 2015 Joyent Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jameson Little

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017 Ilya Kantor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (C) 2011 by Roly Fentanes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) Elan Shanker and Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2007-2014, Alexandru Marasteanu <hello [at] alexei (dot) ro>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of this software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2014 Ben Newman <bn@cs.stanford.edu>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright(c) 2014-2019 Angelos Pikoulas (agelos.pikoulas@gmail.com)

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
Copyright (c) 2015 Aria Minaei

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2017 Calvin Metcalf & contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2015 Calvin Metcalf & contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT

Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2012-2015, JP Richardson <jprichardson@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

BSD 2-Clause License

Copyright (c) 2018, Tadeusz azurski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Steven Chim

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2015 John Hiesey

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2018 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

====

`String.fromCharCode` by Mathias Bynens used according to terms of MIT License, as follows:
Copyright Mathias Bynens <https://mathiasbynens.be/> 

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ISC License (ISC)

Copyright (c) 2015, Glen Maddern

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The ISC License

Copyright (c) 2015 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) Ben Briggs <beneb.info@gmail.com> (http://beneb.info)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2013 Dominic Tarr
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013-2018 sha.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 1998 - 2009, Paul Johnston & Contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2017 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015, 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright JS Foundation and other contributors, https://js.foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jordan Harband
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

* Copyright (c) 2015 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

Copyright (c), Mapbox

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation.
and/or other materials provided with the distribution.
* Neither the name of node-pre-gyp nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2017, Ryan Zimmerman <opensrc@ryanzim.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 KARASZI Istvn

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights

The above copyright notice and this permission notice shall be included in all
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh, and other contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2018 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Nadav Ivgi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016-2017, Brian Woodward.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015-2016 Douglas Christopher Wilson <doug@somethingdoug.com

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
MIT License

Copyright (c) 2016-2018 Ari Porad

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) Vsevolod Strukchinsky <floatdrop@gmail.com> (github.com/floatdrop)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
The MIT License (MIT)

Copyright (c) James Talmage <james@talmage.io> (github.com/jamestalmage)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
Copyright 2008 Fair Oaks Labs, Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote
products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

validate-npm-package-license

Give me a string and I'll tell you if it's a valid npm package license string.

```javascript
var valid = require('validate-npm-package-license');
```

SPDX license identifiers are valid license strings:

```javascript
var assert = require('assert');
var validSPDXExpression = {
validForNewPackages: true,
validForOldPackages: true,
spdx: true
};

assert.deepEqual(valid('MIT'), validSPDXExpression);
assert.deepEqual(valid('BSD-2-Clause'), validSPDXExpression);
assert.deepEqual(valid('Apache-2.0'), validSPDXExpression);
assert.deepEqual(valid('ISC'), validSPDXExpression);
```

The function will return a warning and suggestion for nearly-correct license identifiers:
```javascript
assert.deepEqual(
valid('Apache-2.0'),
{
validForOldPackages: false,
validForNewPackages: false,
warnings: [
'license should be ’ +
'a valid SPDX license expression (without "LicenseRef"), ' +
""UNLICENSED", or ' +
""SEE LICENSE IN <filename>"”,
'license is similar to the valid expression "Apache-2.0”
]
}
);
```

SPDX expressions are valid, too ...

```javascript
// Simple SPDX license expression for dual licensing
assert.deepEqual(
valid('(GPL-3.0-only OR BSD-2-Clause)'),
validSPDXExpression
);
``` ...

... except if they contain `LicenseRef`

```javascript
var warningAboutLicenseRef = {
validForOldPackages: false,
validForNewPackages: false,
spdx: true,
warnings: [
'license should be ’ +
'a valid SPDX license expression (without "LicenseRef"), ' +
""UNLICENSED", or ' +
""SEE LICENSE IN <filename>”",
]
};
``` 

assert.deepEqual(
valid('LicenseRef-Made-Up'),
warningAboutLicenseRef
);
```
assert.deepEqual(
  valid('(MIT OR LicenseRef-Made-Up)'),
  warningAboutLicenseRef
);
```javascript
if (process.env.CI) {
  return invalid('MIT License (MIT)');
}
```

If you can't describe your licensing terms with standardized SPDX identifiers, put the terms in a file in the package and point users there:
```
```javascript
assert.deepEqual(
  valid('SEE LICENSE IN LICENSE.txt'),
  {
    validForNewPackages: true,
    validForOldPackages: true,
    inFile: 'LICENSE.txt'
  }
);
```
```
```javascript
assert.deepEqual(
  valid('SEE LICENSE IN license.md'),
  {
    validForNewPackages: true,
    validForOldPackages: true,
    inFile: 'license.md'
  }
);
```javascript
```

If there aren't any licensing terms, use `UNLICENSED`:
```
```javascript
var unlicensed = {
  validForNewPackages: true,
  validForOldPackages: true,
  unlicensed: true
};
assert.deepEqual(valid('UNLICENSED'), unlicensed);
assert.deepEqual(valid('UNLICENCED'), unlicensed);
```

The MIT License (MIT)

Copyright (c) 2014-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell...
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1871 array prototype flat 1.2.3
1.1871.1 Available under license:
MIT License

Copyright (c) 2017 ECMAScript Shims

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1872 concat-map 0.0.1 0.0.1
1.1872.1 Available under license:
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1873 batik-shared-resources 1.13
1.1873.1 Available under license:

Apache Batik
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the
Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for
Standardization for the definition of character entities used in the software's
documentation.

This product includes images from the Tango Desktop Project
(http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1874 is-buffer 2.0.5

1.1874.1 Available under license:
The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1875 pify 2.3.0

1.1875.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, 
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN 
THE SOFTWARE.

1.1876 commons-codec 1.12

1.1876.1 Available under license:

Apache Commons Codec
Copyright 2002-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===============================================================================

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, 
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by 
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all 
other entities that control, are controlled by, or are under common 
control with that entity. For the purposes of this definition, 
"control" means (i) the power, direct or indirect, to cause the 
direction or management of such entity, whether by contract or 
otherwise, or (ii) ownership of fifty percent (50%) or more of the 
outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1877 p-map 3.0.0

1.1877.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1878 hbase-client 2.4.2
1.1878.1 Available under license:

Apache HBase - Client
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1879 wrappy 1.0.2
1.1879.1 Available under license :

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1880 netty-buffer 4.0.48.Final

1.1880.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
Found in path(s):
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-
  jar/io/netty/buffer/WrappedUnpooledUnsafeDirectByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-
  jar/io/netty/buffer/SimpleLeakAwareCompositeByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-
  jar/io/netty/buffer/AdvancedLeakAwareCompositeByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-
  jar/io/netty/buffer/UnpooledUnsafeNoCleanerDirectByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-
  jar/io/netty/buffer/WrappedCompositeByteBuf.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-
  jar/io/netty/buffer/PooledHeapByteBuf.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4.0.48-final-sources-jar/io/netty/buffer/PooledByteBufAllocatorMetric.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4.0.48-final-sources-jar/io/netty/buffer/ByteBufAllocatorMetricProvider.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4.0.48-final-sources-jar/io/netty/buffer/ByteBufAllocatorMetric.java
jar/io/netty/buffer/ByteBufOutputStream.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/PoolArena.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/ByteBufHolder.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/FixedCompositeByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/ReadOnlyUnsafeDirectByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/AbstractDerivedByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/PooledUnsafeDirectByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/WrappedByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/AdvancedLeakAwareByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/ReadOnlyByteBufferBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/ByteBufProcessor.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/AbstractReferenceCountedByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/EmptyByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources.jar/io/netty/buffer/UnreleasableByteBuf.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-jar/io/netty/buffer/PoolChunkMetric.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-jar/io/netty/buffer/PoolArenaMetric.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-jar/io/netty/buffer/SlicedAbstractByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-jar/io/netty/buffer/PooledUnsafeHeapByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-jar/io/netty/buffer/PoolSubpageMetric.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-jar/io/netty/buffer/AbstractUnsafeSwappedByteBuf.java
* /opt/cola/permits/1202447202_1631568144.52/0/netty-buffer-4-0-48-final-sources-jar/io/netty/buffer/UnsafeByteBufUtil.java

1.1881 fs-constants 1.0.0
1.1881.1 Available under license:
The MIT License (MIT)

Copyright (c) 2018 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1882 debug 3.1.0

1.1882.1 Available under license:

Copyright (c) 2004-2016 Fabien Potencier

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
   Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.
Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified
Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   
   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   
   iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
   
   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial
Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.
5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses
granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program
proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed
only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify

Open Source Used In DNAC Platform Antelope 6734
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

---

## CLASSPATH EXCEPTION
Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

# Notices for Eclipse GlassFish

This content is produced and maintained by the Eclipse GlassFish project.

* Project home: https://projects.eclipse.org/projects/ee4j.glassfish

## Trademarks

Eclipse GlassFish, and GlassFish are trademarks of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:
* https://github.com/eclipse-ee4j/glassfish-ha-api
* https://github.com/eclipse-ee4j/glassfish-logging-annotation-processor
* https://github.com/eclipse-ee4j/glassfish-shoal
* https://github.com/eclipse-ee4j/glassfish-cdi-porting-tck
* https://github.com/eclipse-ee4j/glassfish-jsftemplating
* https://github.com/eclipse-ee4j/glassfish-hk2-extra
* https://github.com/eclipse-ee4j/glassfish-hk2
* https://github.com/eclipse-ee4j/glassfish-fighterfish

## Third-party Content

This project leverages the following third party content.

None

## Cryptography

Content may contain encryption software. The country in which you are currently
may have restrictions on the import, possession, and use, and/or re-export to
another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the import,
possession, or use, and re-export of encryption software, to see if this is
permitted.

### 1.1884 restore-cursor 1.0.1

1.1884.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.1885 jackson 2.2.3
1.1885.1 Available under license:

FasterXML, LLC
Software Grant and Corporate Contributor License Agreement ("Agreement")
http://fasterxml.com/cla
(v1)

Thank you for your interest in FasterXML, LLC ("FasterXML"). In order to clarify the intellectual property license granted with Contributions from any person or entity, FasterXML must have a Contributor License Agreement (CLA) on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of FasterXML and its users; it does not change your rights to use your own Contributions for any other purpose.

This version of the Agreement allows an entity (the "Corporation") to submit Contributions to the FasterXML, to authorize Contributions submitted by its designated employees to FasterXML, and to grant copyright and patent licenses thereto.

If you have not already done so, please complete and sign, then scan and email a pdf file of this Agreement to clas@fasterxml.com. If necessary, send an original signed Agreement to FasterXML, LLC, 600 N 36th Ave, Suite 409, Seattle, WA 98103.

Please read this document carefully before signing and keep a copy for your records.

Corporation name: ________________________________________________

Corporation address: ________________________________________________

________________________________________________

________________________________________________

Point of Contact: ________________________________________________

E-Mail: ________________________________________________

Telephone: _____________________ Fax: _____________________
You accept and agree to the following terms and conditions for Your present and future Contributions submitted to FasterXML. Except for the license granted herein to FasterXML and recipients of software distributed by FasterXML, You reserve all right, title, and interest in and to Your Contributions.

1. Definitions.

"You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with FasterXML. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Contribution" shall mean the code, documentation or other original works of authorship expressly identified in Schedule B, as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to FasterXML for inclusion in, or documentation of, any of the products owned or managed by FasterXML (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to FasterXML or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, FasterXML for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as "Not a Contribution."

2. Grant of Copyright License. Subject to the terms and conditions of this Agreement, You hereby grant to FasterXML and to recipients of software distributed by FasterXML a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

3. Grant of Patent License. Subject to the terms and conditions of this Agreement, You hereby grant to FasterXML and to recipients of software distributed by FasterXML a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable patent license to use, contribute to, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.
of software distributed by FasterXML a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) were submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.

4. You represent that You are legally entitled to grant the above license. You represent further that each employee of the Corporation designated on Schedule A below (or in a subsequent written modification to that Schedule) is authorized to submit Contributions on behalf of the Corporation.

5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others).

6. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

7. Should You wish to submit work that is not Your original creation, You may submit it to FasterXML separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]".

8. It is your responsibility to notify FasterXML when any change is required to the list of designated employees authorized to submit Contributions on behalf of the Corporation, or to the Corporation's Point of Contact with FasterXML.
1.1886 prelude-ls 1.1.2

1.1886.1 Available under license:

Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1887 nopt 1.0.10

1.1887.1 Available under license:

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1888 statuses 1.2.1 1.2.1

1.1888.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1889 hk2-utils 2.5.0-b32
1.1889.1 Available under license :
CDDL + GPLv2 with classpath exception
https://glassfish.java.net/nonav/public/CDDL+GPL_1_1.html
A business-friendly OSS license

1.1890 slash 2.0.0
1.1890.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1891 fast-deep-equal 2.0.1 2.0.1
1.1891.1 Available under license :
MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1892 babel-plugin-transform-es2015-
modules-amd 6.24.1

1.1892.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1893 find-cache-dir 3.3.1

1.1893.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1894 slash 1.0.0

1.1894.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1895 jasper-runtime 5.5.12

1.1895.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of
fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any
medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.1896 reflections 0.9.10
1.1896.1 Available under license :
DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. You just DO WHAT THE FUCK YOU WANT TO.

1.1897 guice-servlet 4.0
1.1897.1 Available under license :

Google Guice - Extensions - Servlet
Copyright 2006-2015 Google, Inc.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1898 istanbul-lib-instrument 4.0.3

1.1898.1 Available under license:

Copyright 2012-2015 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1899 tough-cookie 2.3.4 2.3.4

1.1899.1 Available under license:

Copyright (c) 2015, Salesforce.com, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of Salesforce.com nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGligence OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

The following exceptions apply:

==

`public_suffix_list.dat` was obtained from
<https://publicsuffix.org/list/public_suffix_list.dat> via
<http://publicsuffix.org>. The license for this file is MPL/2.0. The header of
that file reads as follows:

// This Source Code Form is subject to the terms of the Mozilla Public
// License, v. 2.0. If a copy of the MPL was not distributed with this
// file, You can obtain one at http://mozilla.org/MPL/2.0/.

1.1900 map-stream 0.1.0 0.1.0

1.1900.1 Available under license :
Copyright (c) 2011 Dominic Tarr

Permission is hereby granted, free of charge,
to any person obtaining a copy of this software and
associated documentation files (the "Software"), to
deal in the Software without restriction, including
without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom
the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice
shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED. INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1901 checker-compat-qual 2.0.0

1.1902 utils-merge 1.0.0
1.1902.1 Available under license :
(The MIT License)

Copyright (c) 2013 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1903 define-properties 1.1.3
1.1903.1 Available under license :
The MIT License (MIT)

Copyright (C) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1904 read 1.0.7
1.1904.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1905 fs-readdir-recursive 1.1.0
1.1905.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1906 should-type 1.4.0 1.4.0

1.1906.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1907 typedarray-to-buffer 3.1.5

1.1907.1 Available under license :

The MIT License (MIT)

Copyright (c) Feross Aboukhadieh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1908 invert-kv 1.0.0
1.1908.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1909 kerb-client 1.0.0
1.1909.1 Available under license :

Kerby-kerb Client
Copyright 2014-2017 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1910 jboss-logging 3.1.1.GA

1.1910.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with
We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.
The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.
This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.
6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.1911 babel-plugin-transform-es2015-computed-properties 6.24.1
1.1911.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The "Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is
being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.1912 browserslist 4.16.3

1.1912.1 Available under license:
The MIT License (MIT)

Copyright 2014 Andrey Sitnik <andrey@sitnik.ru> and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1913 drill-iceberg-metastore 1.19.0

1.1913.1 Available under license:

Drill : Metastore : Iceberg
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.1914 kafka-schema-serializer 6.1.1

1.1914.1 Available under license:

Confluent Community License Agreement
Version 1.0

This Confluent Community License Agreement Version 1.0 (the Agreement) sets forth the terms on which Confluent, Inc. (Confluent) makes available certain software made available by Confluent under this Agreement (the Software). BY INSTALLING, DOWNLOADING, ACCESSING, USING OR DISTRIBUTING ANY OF THE SOFTWARE, YOU AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO SUCH TERMS AND CONDITIONS, YOU MUST NOT USE THE SOFTWARE. IF YOU ARE RECEIVING THE SOFTWARE ON BEHALF OF A LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE THE ACTUAL AUTHORITY TO AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT ON BEHALF OF SUCH ENTITY. Licensee means you, an individual, or the entity on whose behalf you are receiving the Software.

1. LICENSE GRANT AND CONDITIONS.

1.1 License. Subject to the terms and conditions of this Agreement, Confluent hereby grants to Licensee a non-exclusive, royalty-free, worldwide, non-transferable, non-sublicenseable license during the term of this Agreement to: (a) use the Software; (b) prepare modifications and derivative works of the Software; (c) distribute the Software (including without limitation in source code or object code form); and (d) reproduce copies of the Software (the License). Licensee is not granted the right to, and Licensee shall not, exercise the License for an Excluded Purpose. For purposes of this Agreement, Excluded Purpose means making available any software-as-a-service, platform-as-a-service, infrastructure-as-a-service or other similar online service that competes with Confluent products or services that provide the Software.

1.2 Conditions. In consideration of the License, Licensees distribution of the Software is subject to the following conditions:

(a) Licensee must cause any Software modified by Licensee to carry prominent notices stating that Licensee modified the Software.

(b) On each Software copy, Licensee shall reproduce and not remove or alter all Confluent or third party copyright or other proprietary notices contained in the Software, and Licensee must provide the notice below with each copy.

This software is made available by Confluent, Inc., under the terms of the Confluent Community License Agreement, Version 1.0 located at http://www.confluent.io/confluent-community-license. BY
INSTALLING, DOWNLOADING, ACCESSING, USING OR DISTRIBUTING ANY OF THE SOFTWARE, YOU AGREE TO THE TERMS OF SUCH LICENSE AGREEMENT.

1.3 Licensee Modifications. Licensee may add its own copyright notices to modifications made by Licensee and may provide additional or different license terms and conditions for use, reproduction, or distribution of Licensee's modifications. While redistributing the Software or modifications thereof, Licensee may choose to offer, for a fee or free of charge, support, warranty, indemnity, or other obligations. Licensee, and not Confluent, will be responsible for any such obligations.

1.4 No Sublicensing. The License does not include the right to sublicense the Software, however, each recipient to which Licensee provides the Software may exercise the Licenses so long as such recipient agrees to the terms and conditions of this Agreement.

2. TERM AND TERMINATION. This Agreement will continue unless and until earlier terminated as set forth herein. If Licensee breaches any of its conditions or obligations under this Agreement, this Agreement will terminate automatically and the License will terminate automatically and permanently.

3. INTELLECTUAL PROPERTY. As between the parties, Confluent will retain all right, title, and interest in the Software, and all intellectual property rights therein. Confluent hereby reserves all rights not expressly granted to Licensee in this Agreement. Confluent hereby reserves all rights in its trademarks and service marks, and no licenses therein are granted in this Agreement.

4. DISCLAIMER. CONFLUENT HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE.

5. LIMITATION OF LIABILITY. CONFLUENT WILL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, LOST PROFITS OR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, INDIRECT, OR DIRECT DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING OUT OF THIS AGREEMENT. THE FOREGOING SHALL APPLY TO THE EXTENT PERMITTED BY APPLICABLE LAW.

6. GENERAL.

6.1 Governing Law. This Agreement will be governed by and interpreted in accordance with the laws of the state of California, without reference to its conflict of laws principles. If Licensee is located within the United States, all disputes arising out of this Agreement are subject to the exclusive jurisdiction of courts located in Santa Clara County, California, USA. If Licensee is located outside of the United States,
any dispute, controversy or claim arising out of or relating to this Agreement will be referred to and finally determined by arbitration in accordance with the JAMS International Arbitration Rules. The tribunal will consist of one arbitrator. The place of arbitration will be Palo Alto, California. The language to be used in the arbitral proceedings will be English. Judgment upon the award rendered by the arbitrator may be entered in any court having jurisdiction thereof.

6.2 Assignment. Licensee is not authorized to assign its rights under this Agreement to any third party. Confluent may freely assign its rights under this Agreement to any third party.

6.3 Other. This Agreement is the entire agreement between the parties regarding the subject matter hereof. No amendment or modification of this Agreement will be valid or binding upon the parties unless made in writing and signed by the duly authorized representatives of both parties. In the event that any provision, including without limitation any condition, of this Agreement is held to be unenforceable, this Agreement and all licenses and rights granted hereunder will immediately terminate. Waiver by Confluent of a breach of any provision of this Agreement or the failure by Confluent to exercise any right hereunder will not be construed as a waiver of any subsequent breach of that right or as a waiver of any other right.

Apache Kafka
Copyright 2016 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This distribution has a binary dependency on jersey, which is available under the CDDL License. The source code of jersey can be found at https://github.com/jersey/jersey/.

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.
It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits
A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.
Java ClassMate library was originally written by Tatu Saloranta (tatu.saloranta@iki.fi)

Other developers who have contributed code are:

* Brian Langel
This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="Home page of The Apache Software Foundation">
<link rel="apple-touch-icon" sizes="57x57" href="/favicons/apple-touch-icon-57x57.png">
<link rel="apple-touch-icon" sizes="60x60" href="/favicons/apple-touch-icon-60x60.png">
<link rel="apple-touch-icon" sizes="72x72" href="/favicons/apple-touch-icon-72x72.png">
<link rel="apple-touch-icon" sizes="76x76" href="/favicons/apple-touch-icon-76x76.png">
<link rel="apple-touch-icon" sizes="114x114" href="/favicons/apple-touch-icon-114x114.png">
<link rel="apple-touch-icon" sizes="120x120" href="/favicons/apple-touch-icon-120x120.png">
<link rel="apple-touch-icon" sizes="144x144" href="/favicons/apple-touch-icon-144x144.png">
<link rel="apple-touch-icon" sizes="152x152" href="/favicons/apple-touch-icon-152x152.png">
<link rel="apple-touch-icon" sizes="180x180" href="/favicons/apple-touch-icon-180x180.png">
<link rel="icon" type="image/png" href="/favicons/favicon-32x32.png" sizes="32x32">
<link rel="icon" type="image/png" href="/favicons/favicon-194x194.png" sizes="194x194">
<link rel="icon" type="image/png" href="/favicons/favicon-96x96.png" sizes="96x96">
<link rel="icon" type="image/png" href="/favicons/android-chrome-192x192.png" sizes="192x192">
<link rel="manifest" href="/favicons/manifest.json">
<link rel="shortcut icon" href="/favicons/favicon.ico">
<meta name="msapplication-TileColor" content="#603cba">
<meta name="msapplication-TileImage" content="/favicons/mstile-144x144.png">
<meta name="msapplication-config" content="/favicons/browserconfig.xml">
<meta name="theme-color" content="#303284">
<title>Apache License, Version 2.0</title>
<link href='https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700%7cDroid+Serif:400,700' rel='stylesheet' type='text/css'>
Apache License
Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation
or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.</p>

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).</p>

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.</p>

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."</p>

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.</p>

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.</p>

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and
- You must cause any modified files to carry prominent notices stating that You changed the files; and
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor.
regarding such Contributions.</p>
<p>6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.</p>
<p>7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.</p>
<p>8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.</p>
<p>9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.</p>
<h1 id="apply">APPENDIX: How to apply the Apache License to your work</h1>
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "["]"
Copyright: 2018 Confluent, Inc.
License: Apache-2

Files: core/*
Copyright: 2015 Confluent, Inc.
License: Confluent Community License

License: Confluent Community License
Licensed under the Confluent Community License; you may not use this file except in compliance with the License. You may obtain a copy of the License at

copyright@confluent.io/confluent-community-license

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

License: Apache-2
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

copyright@apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

On Debian systems, the Apache 2.0 license can be found in
/usr/share/common-licenses/Apache-2.0.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This distribution has a binary dependency on jersey, which is available under the CDDL
License as described below.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL - Version 1.1)
1. Definitions.
1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.
2.1. The Initial Developer Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You
distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.
4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the
license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections
2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOOPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.
NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The following commands were used to generate license and notice files. Replace <VERSION> with the Schema Registry version, <SRC_PATH> with the path to the Schema Registry source directory, and <LICENSE_TOOL_PATH> with the path of the license tool.

cd <SRC_PATH>
mvn package -DskipTests
mkdir /tmp/jars
mkdir /tmp/overrides
cd <LICENSE_TOOL_PATH>
./bin/run_license_job.bash -i /tmp/jars -l <SRC_PATH>/licenses -n <SRC_PATH>/notices -h <SRC_PATH>/licenses-and-notices.html -o /tmp/overrides

Apache Avro
Copyright 2009-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
<?!DOCTYPE HTML PUBLIC "-/W3C/DTD HTML 4.01 Transitional/EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html lang="en-US">
<head>
<title>GlassFish Dual License Header and License Notice GPL v2 and CDDL</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<style type="text/css">
li {display: block;}
</style>
</head>

<body leftmargin="0" topmargin="0" marginheight="0" marginwidth="0" bgcolor="#ffffff">

<h3>COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0</h3>

<ul>
<li><p>1. Definitions.</p></li>
</ul>
1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. “License” means this document.

1.8. “Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. “Modifications” means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. “Original Software” means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. “Patent Claims” means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. “Source Code” means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. “You” (or “Your”) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, “You” includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.
2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

The licenses granted in Sections and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

Notwithstanding Section above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.
2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- (a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

- (b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

- (c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

- (d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by:
(i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

</p>
</li>
</ul>
</li>
</ul>
</li>

<li>
<p><b>3. Distribution Obligations.</b></p>
<ul>
<li>
<p><b>3.1. Availability of Source Code.</b></p>
<p>Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.</p>
</li>
<li>
<p><b>3.2. Modifications.</b></p>
<p>The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.</p>
</li>
<li>
<p><b>3.3. Required Notices.</b></p>
<p>You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification.</p>
</li>
</ul>
</li>
You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You:
(a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim
(excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

</li>
</ul>

<li>
<p><b>6.3.</b> In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.
</p>
</li>
</ul>

<li>
<p><b>7. LIMITATION OF LIABILITY.</b></p>
</li>

<p>UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.
8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software documentation; as such terms are used in 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or
otherwise make available any Covered Software.
</p>
</li>

<li>
<p><b>10. RESPONSIBILITY FOR CLAIMS.</b></p>
<p>As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.
</p>
</li>

<li>
<p><b>NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)</b>
</p>
<p>The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.
</p>
</li>

</ul>

<h3>The GNU General Public License (GPL) Version 2, June 1991</h3>

<p>Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
</p>
<p>Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
</p>

<p><b>Preamble</b></p>
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will
individuals obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.
<p>
</p>
The precise terms and conditions for copying, distribution and modification follow.
<p>
</p><p><b>TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION</b></p><ul style="margin-left:0; padding-left:0; border-left:0">
<li>
 This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law; that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".
</li>
<li>
 Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
</li>
<li>
 You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
</li>
<li>
 You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
</li>
</ul>
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

<li><p><b>4.</b> You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.</p></li>

<li><p><b>5.</b> You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.</p></li>

<li><p><b>6.</b> Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.
SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY
SERVING, REPAIR OR CORRECTION.
</p>
</li>

<li>
<p>
<b>12.</b> IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO
LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR
THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.
</p>
</li>

<li>
<p>END OF TERMS AND CONDITIONS</p>
</li>

<li>
<p><b>How to Apply These Terms to Your New Programs</b></p>
<p>If you develop a new program, and you want it to be of the greatest possible
use to the public, the best way to achieve this is to make it free software
which everyone can redistribute and change under these terms.
</p>
</li>

<ul>
<li>
<p>One line to give the program’s name and a brief idea of what it does.</p>
</li>
</ul>

Copyright (C) <year> <name of author>
</p>
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.
You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.
module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.
following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification;

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by granter.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor
(or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.
4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If you assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If you assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such
Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. § 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to share and change it. You must show them these terms so they know their rights.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program," below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification." Each licensee is addressed as "you." Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
<blockquote>a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

<blockquote>b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

<blockquote>c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)</blockquote>
</blockquote>

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other
system and a licensee cannot impose that choice.</p>
<p>This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.</p>
<p><strong>8</strong>. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.</p>
<p><strong>9</strong>. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.</p>
<p>Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and any later version, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.</p>
<p><strong>10</strong>. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.</p>

**NO WARRANTY**

<p><strong>11</strong>. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.</p>
<p><strong>12</strong>. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.</p>

**END OF TERMS AND CONDITIONS**

**How to Apply These Terms to Your New Programs**

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.</p>
<p>To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the copyright notice; line and a pointer to where the full notice is found.</p>
<blockquote>One line to give the program’s name and a brief idea of what it does.</blockquote>
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a copyright disclaimer; for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Oracle in the License file that accompanied this code. 
<br />
As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.
This copy of Jackson JSON processor is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/

A copy is also included with both the downloadable source code package and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/asl"; and in jar "META-INF/"

The project is licensed under the Confluent Community License, except for client libs, which is under the Apache 2.0 license.

See LICENSE file in each subfolder for detailed license agreement.

Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.
It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

</head>

<body>

<!-- Navigation -->
<header>
<nav class="navbar navbar-default navbar-fixed-top">
<div class="container">
<div class="navbar-header">
<button class="navbar-toggle" type="button" data-toggle="collapse" data-target="#mainnav-collapse">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a href="#" class="navbar-brand"><span class="glyphicon glyphicon-home"></span></a>
</div>
<div class="collapse navbar-collapse" id="mainnav-collapse">
<div style="line-height:20px; padding-top:5px; float:left"><a href="/" class="navbar-brand">Home</a>&nbsp;&raquo&nbsp;<a href="/licenses/" class="navbar-brand">Licenses</a></div>
<ul class="nav navbar-nav navbar-right">
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown">About <span class="caret"></span></a>
<ul class="dropdown-menu" role="menu">
<li><a href="/foundation/" class="navbar-brand">Overview</a></li>
<li><a href="/foundation/members.html" class="navbar-brand">Members</a></li>
<li><a href="/foundation/how-it-works.html" class="navbar-brand">Process</a></li>
<li><a href="/foundation/sponsorship.html" class="navbar-brand">Sponsorship</a></li>
<li><a href="/foundation/glossary.html" class="navbar-brand">Glossary</a></li>
<li><a href="/foundation/preFAQ.html" class="navbar-brand">FAQ</a></li>
<li><a href="/foundation/contact.html" class="navbar-brand">Contact</a></li>
</ul>
</li>
<li><a href="/index.html#projects-list" class="navbar-brand">Projects</a></li>
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown">People <span class="caret"></span></a>
<ul class="dropdown-menu" role="menu">
<li><a href="http://people.apache.org/" class="navbar-brand">Overview</a></li>
<li><a href="http://people.apache.org/committer-index.html" class="navbar-brand">Committers</a></li>
<li><a href="/foundation/how-it-works.html#meritocracy" class="navbar-brand">Meritocracy</a></li>
<li><a href="/foundation/how-it-works.html#roles" class="navbar-brand">Roles</a></li>
<li><a href="http://planet.apache.org/" class="navbar-brand">Planet Apache</a></li>
</ul>
</li>
</ul>
</div>
</nav>
</header>

...
The Apache Software Foundation uses various licenses to distribute software and documentation, to accept regular contributions from individuals and corporations, and to accept larger grants of existing software products. These licenses help us achieve our goal of providing reliable and long-lived software products through collaborative open source software development. In all cases, contributors retain full rights to use their original contributions for any other purpose outside of Apache while providing the ASF and its projects the right to distribute and build upon their work within Apache.

All software produced by The Apache Software Foundation or any of its projects or subjects is licensed according to the terms of the documents listed below.

Apache License, Version 2.0 (current)

The 2.0 version of the Apache License was approved by the ASF in 2004. The goals of this license revision have been to reduce the number of frequently asked questions, to allow the license to be reusable without modification by any project (including non-ASF projects), to allow the license to be included by reference instead of listed in every file, to clarify the license on submission of contributions, to require a patent license on contributions that necessarily infringe the contributor's own patents, and to move comments regarding Apache and other inherited attribution notices to a location outside the license terms (the NOTICE file).
The result is a license that is supposed to be compatible with other open source licenses, while remaining true to the original goals of the Apache Group and supportive of collaborative development across both nonprofit and commercial organizations. The Apache Software Foundation is still trying to determine if this version of the Apache License is compatible with the GPL.

All packages produced by the ASF are implicitly licensed under the Apache License, Version 2.0, unless otherwise explicitly stated. More developer documentation on how to apply the Apache License to your work can be found in Applying the Apache License, Version 2.0.

The 1.1 version of the Apache License was approved by the ASF in 2000. The primary change from the 1.0 license is in the 'advertising clause' (section 3 of the 1.0 license); derived products are no longer required to include attribution in their advertising materials, only in their documentation.

Individual packages licensed under the 1.1 version may have used different wording due to varying requirements for attribution or mark identification, but the binding terms were all the same.

This is the original Apache License which applies only to older versions of Apache packages (such as version 1.2 of the Web server).

The ASF desires that all contributors of ideas, code, or documentation to any Apache projects complete, sign, and submit (via fax or email) an Individual Contributor License Agreement (ICLA). The purpose of this agreement is to clearly define the terms under which intellectual property has been contributed to the ASF and thereby allow us to defend the project should there be a legal dispute regarding the software at some future time. A signed ICLA is required to be on file before an individual is given commit rights to an ASF project.

For a corporation that has assigned employees to work on an Apache project, a Corporate CLA (CCLA) is available for contributing intellectual property via the corporation, that may have been assigned as part of an employment agreement. Note that a Corporate CLA does not remove the need for every developer to sign their own ICLA as an individual, to cover any of their contributions which are not owned by the corporation signing the CCLA.

The ICLA is not tied to any employer you may have, so it is recommended to use one's personal email address in the contact details, rather than an @work address.

Your Full name will be published unless you provide an alternative Public name.
For example if your full name is Andrew Bernard Charles Dickens, but you wish to be known as Andrew Dickens, please enter the latter as your Public name.</p>

<p>The email address and other contact details are not published.</p>

<h1 id="grants">Software Grants</h1>

<p>When an individual or corporation decides to donate a body of existing software or documentation to one of the Apache projects, they need to execute a formal Software Grant Agreement (SGA) with the ASF. Typically, this is done after negotiating approval with the ASF Incubator or one of the PMCs, since the ASF will not accept software unless there is a viable community available to support a collaborative project.</p>

<h1 id="submitting">Submitting License Agreements and Grants</h1>

<p>Documents may be submitted by fax or email.</p>

<p>If submitting by fax, please print, sign, and fax all pages of the document to +1-919-573-9199. Please send documents right-side-up, first page first; and send only one document per fax.</p>

<p>If submitting by email, please fill the form with a pdf viewer, print, sign, scan all pages into a single pdf file, and email the pdf file as an attachment to secretary@apache.org. If possible, send the attachment from the email address in the document. Please send only one document per email.</p>

<p>If you prefer to sign electronically, please fill the form, save it locally (e.g. icla.pdf), and sign the file by preparing a detached PGP signature. For example,</p>

<blockquote>
<p>gpg --armor --detach-sign icla.pdf</p>
</blockquote>

<p>The above will create a file icla.pdf.asc. Send both the file and signature as attachments in the same email to secretary@apache.org. Please send only one document (file plus signature) per email. Please do not submit your public key to Apache. Instead, please upload your public key to pgpkeys.mit.edu.</p>

<p>The files typically are named icla.pdf and icla.pdf.asc for individual agreements; ccla.pdf and ccla.pdf.asc for corporate agreements; software-grant.pdf and software-grant.pdf.asc for grants.</p>

<h1 id="crypto">Export restrictions</h1>

<p>For export restriction information, please consult our ASF Export Classifications page.</p>

<h1 id="trademarks">Trademark and Logo Usage</h1>

<p>For ASF trademark and logo usage information, please consult our ASF Trademark Use Policy page.</p>

<h1 id="questions">Questions?</h1>

<p>For answers to frequently asked licensing questions, please consult our Licensing Frequently Asked Questions page.</p>

<br/>
This product currently only contains code developed by authors of specific components, as identified by the source code files; if such notes are missing files have been created by Tatu Saloranta.
For additional credits (generally to people who reported problems) see CREDITS file.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
   distribution, then any Derivative Works that You distribute must
   include a readable copy of the attribution notices contained
   within such NOTICE file, excluding those notices that do not
   pertain to any part of the Derivative Works, in at least one
   of the following places: within a NOTICE text file distributed
   as part of the Derivative Works; within the Source form or
   documentation, if provided along with the Derivative Works; or,
   within a display generated by the Derivative Works, if and
   wherever such third-party notices normally appear. The contents
   of the NOTICE file are for informational purposes only and
   do not modify the License. You may add Your own attribution
   notices within Derivative Works that You distribute, alongside
   or as an addendum to the NOTICE text from the Work, provided
   that such additional attribution notices cannot be construed
   as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement You may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1915 is-boolean-object 1.1.0
1.1915.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.1916 ms 2.1.2
1.1916.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Zeit, Inc.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1917 hbase-http 2.4.2
1.1917.1 Available under license:

Apache HBase - HTTP
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1918 animal-sniffer-annotation 1.18

1.1919 babel-plugin-transform-es2015-sticky-regex 6.24.1
Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THE SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.1920 yargs 16.2.0
1.1920.1 Available under license:
MIT License

Copyright 2010 James Halliday (mail@substack.net); Modified work Copyright 2014 Contributors (ben@npmjs.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1921 foodmart-data-json 0.4
1.1921.1 Available under license:
foodmart-data-json
Foodmart data set in JSON format
Copyright (C) 2013-2013 Pentaho
Copyright (C) 2015-2015 Julian Hyde
All Rights Reserved.

Based upon the Pentaho mondrian-data-foodmart-json artifact developed as part of the Pentaho Mondrian OLAP engine
(http://mondrian.pentaho.com).

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this software except in compliance with the License.
You may obtain a copy of the License at:
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1922 tiger-types 1.4

1.1923 jersey-guava 2.25.1
1.1923.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2010 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/ForwardingListMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/base/Ascii.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/annotations/package-info.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/ForwardingSetMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/base/Equivalence.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/SortedMapDifference.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/UnmodifiableListIterator.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/ForwardingSortedSetMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/ForwardingImmutableCollection.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/UncaughtExceptionHandler.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/primitives/package-info.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/AtomicReferenceTable.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/RowSortedTable.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/base/Strings.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/MinMaxPriorityQueue.java

No license file was found, but licenses were detected in source scan.

/
* Copyright (C) 2008 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/RegularImmutableMap.java
No license file was found, but licenses were detected in source scan.

*/

* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/ImmutableAsList.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/reflect/TypeResolver.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Platform.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/net/MediaType.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/math/package-info.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/math/IntMath.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Absent.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/WellBehavedMap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/ForwardingLoadingCache.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Absent.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Enforced.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/math/DoubleMath.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/Uninterruptibles.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/CacheBuilderSpec.java
Open Source Used In DNAC Platform Antelope

* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collection/RegularImmutableMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/base/Optional.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/util/concurrent/DateTimeFormatter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/util/concurrent/ExecutionError.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/util/concurrent/ForwardingCheckedFuture.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Present.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/math/DoubleUtils.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Ticker.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/ForwardingCache.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/UncheckedExecutionException.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2006 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/annotations/VisibleForTesting.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/escape/CharEscaper.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2013 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/MultimapBuilder.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collection/ImmutableMapEntry.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/io/CharSequenceReader.java
No license file was found, but licenses were detected in source scan.
/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collect/SortedMultiset.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2013 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/eventbus/SubscriberExceptionContext.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 */
* http://www.apache.org/licenses/LICENSE-2.0
 *
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/io/GwtWorkarounds.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/FilteredKeyMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableRangeSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Utf8.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2013 The Guava Authors
 *
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
 *
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Utf8.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Verify.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/VerifyException.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/HashingInputStream.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2011 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/Queues.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/AbstractSortedMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/HashedMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/AbstractHashMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/HashCode.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/HashFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/Funnels.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/Hashing.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/BoundType.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/AbstractMapBasedMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/EmptyImmutableSortedMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/Hasher.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/BloomFilter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/TreeRangeSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/Murmur3_128HashFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/TreeRangeSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/AbstractMapBasedMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/package-info.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/package-info.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/AbstractStreamingHashFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/BloomFilter.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/HashingOutputStream.java

Open Source Used In DNAC Platform Antelope  6888
jar/jersey/repackaged/com/google/common/base/Supplier.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/Constraint.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/MapDifference.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/package-info.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/MapConstraint.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/MapMultimap.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/MutableClassToInstanceMap.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/MutableSetMultiset.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/ObjectArrays.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/TreeMultimap.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/ReverseOrdering.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/eventbus/AllowConcurrentEvents.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/ConcurrentHashMultiset.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/AbstractSortedSetMultimap.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/ObjectArrays.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/io/LineReader.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/AbstractBiMap.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/HashMultimap.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/MutableClassToInstanceMap.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/MapConstraint.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collection/Multimap.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/io/LineBuffer.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/Predicate.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/RunnableFuture.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/ListenableFuture.java
* /opt/cola/permits/113613812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repa...
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/collect/EmptyContiguousSet.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/base/Preconditions.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/collect/Interners.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/collect/EnumMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/collect/HashMap.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2008 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/internal/Finalizer.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2010 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ComputingConcurrentHashMap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/SortedLists.java
No license file was found, but licenses were detected in source scan.
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/SerializingExecutor.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableSortedSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/TreeBasedTable.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/SerializingExecutor.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableMapEntrySet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableMapKeySet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/SerializingExecutor.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/EmptyImmutableSortedSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/EmptyImmutableListMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/Table.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/FluentIterable.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/Platform.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/ListenableFutureTask.java
No license file was found, but licenses were detected in source scan.

/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/Striped64.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/util/concurrent/AtomicDoubleArray.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/reflect/Reflection.java

No license file was found, but licenses were detected in source scan.
/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/*
 * This method was written by Doug Lea with assistance from members of JCP
 * JSR-166 Expert Group and released to the public domain, as explained at
 * http://creativecommons.org/licenses/publicdomain
 *
 * As of 2010/06/11, this method is identical to the (package private) hash
 * method in OpenJDK 7's java.util.HashMap class.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/util/concurrent/Striped.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the
 * License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/util/concurrent/AbstractRangeSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collect/Count.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/GeneralRange.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableSortedMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/RangeSet.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/primitives/UnsignedLong.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/SortedIterable.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/collect/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy
 * of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/base/FunctionalEquivalence.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */

Found in path(s):
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and limitations under the License.
*/

No license file was found, but licenses were detected in source scan.
* Copyright (C) 2010 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS-IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/collection/GenericMapMaker.java

No license file was found, but licenses were detected in source scan.

//*
* Copyright (C) 2008 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

/*
* This method was rewritten in Java from an intermediate step of the Murmur hash function in
* http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
* following header:
* *
* MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
* hereby disclaims copyright to this source code.
*/

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
  jar/jersey/repackaged/com/google/common/collection/Hashing.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-jar/jersey/repackaged/com/google/common/hash/package-info.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

/*
 * This following method is a modified version of one found in
 * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
 * which contained the following notice:
 * Written by Doug Lea with assistance from members of JCP JSR-166
 * Expert Group and released to the public domain, as explained at
 * http://creativecommons.org/publicdomain/zero/1.0/
 * Other contributors include Andrew Wright, Jeffrey Hayes,
 * Pat Fisher, Mike Judd.
 */
Open Source Used In DNAC Platform Antelope 6906

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/util/concurrent/MoreExecutors.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/cache/Weigher.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/math/MathPreconditions.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
jar/jersey/repackaged/com/google/common/collect/FilteredSetMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/0/jersey-guava-2-25-1-sources-4-
Open Source Used In DNAC Platform Antelope 6907

jar/jersey/repackaged/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jar/jersey/repackaged/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/reflect/TypeCapture.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/reflect/Parameter.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/RangeMap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/xml/package-info.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/FilteredEntryMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/TreeTraverser.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/io/ByteSource.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/reflect/MutableTypeToInstanceMap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/reflect/AbstractInvocationHandler.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/cache/LongAddable.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/ForwardingDeque.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4.jar/jersey/repackaged/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1136113812_1613679412.19/jersey-guava-2-25-1-sources-4-
Open Source Used In DNAC Platform Antelope

1.1924 async-replace-with-limit 1.1.0
1.1924.1 Available under license:
MIT

1.1925 babel-plugin-transform-es2015-literals 6.22.0
1.1925.1 Available under license:
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1926 @babel/plugin-transform-named-capturing-groups-regex 7.12.13
1.1926.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1927 @babel/plugin-transform-unicode-regex 7.12.13
1.1927.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1928 swagger-annotations 1.6.2

1.1928.1 Available under license:
   No license file was found, but licenses were detected in source scan.

   Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.html

   Found in path(s):
   * /opt/ws_local/PERMITS_SQL/1088518024_1599800722.99/0/swagger-annotations-1-6-2-jar/META-INF/MANIFEST.MF

1.1929 parseqs 0.0.6

1.1929.1 Available under license:
   The MIT License (MIT)

   Copyright (c) 2015 Gal Koren

   Permission is hereby granted, free of charge, to any person obtaining a copy
   of this software and associated documentation files (the "Software"), to deal
   in the Software without restriction, including without limitation the rights
   to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
   copies of the Software, and to permit persons to whom the Software is
   furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in
   all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
   IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
   FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
   AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
   LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
   OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
   THE SOFTWARE.

1.1930 onetime 2.0.1

1.1930.1 Available under license:
   The MIT License (MIT)

   Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

   Permission is hereby granted, free of charge, to any person obtaining a copy
   of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1931 nomnom 1.8.1 1.8.1

1.1931.1 Available under license :
Copyright (c) 2010 Heather Arthur

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1932 swagger2openapi 7.0.3 7.0.3

1.1932.1 Available under license :
BSD 3-Clause License

Copyright (c) 2016, Mermade Software
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.1933 bean-validation-api 2.0.1.Final
1.1933.1 Available under license :
Bean Validation API

License: Apache License, Version 2.0
See the license.txt file in the root directory or <http://www.apache.org/licenses/LICENSE-2.0>.

1.1934 babel-plugin-transform-async-to-generator 6.24.1 6.24.1
1.1934.1 Available under license :
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1935 readable-stream 2.2.7 2.2.7

1.1935.1 Available under license:

Node.js is licensed for use as follows:

```text
===
Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===
```

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

```text
===
Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

1.1936 negotiator 0.6.1
1.1936.1 Available under license :
(The MIT License)

Copyright (c) 2012-2014 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schlueter
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
-distribute, sublicense, and/or sell copies of the Software, and to
-permit persons to whom the Software is furnished to do so, subject to
-the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
-CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
-TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.1937 reactapp 1.0.0 1.0.0

1.1937.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1938 tweetnacl 0.14.5 0.14.5

1.1938.1 Available under license:

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <http://unlicense.org>

1.1939 nopt 3.0.6 3.0.6
1.1939.1 Available under license :
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1940 hoek 2.16.3 2.16.3
1.1940.1 Available under license :
Copyright (c) 2011-2014, Walmart and other contributors.
Copyright (c) 2011, Yahoo Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/hapi/graphs/contributors
Portions of this project were initially based on the Yahoo! Inc. Postmile project, published at https://github.com/yahoo/postmile.

1.1941 emoji-regex 8.0.0

1.1941.1 Available under license:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1942 @babel/code-frame 7.5.5
1.1942.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1943 @babel/runtime 7.0.0-beta.44

1.1943.1 Available under license:

MIT License

Copyright (c) 2014-2018 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
1.1944 xml-apis-ext 1.3.04

1.1944.1 Available under license:

xml-commons/java/external/README.dom.txt

HEAR YE, HEAR YE!

All of the .java software and associated documentation about the DOM in this repository are distributed under the license from the W3C, which is provided herein.

LICENSE.dom-software.txt covers all software from the W3C including the following items in the xml-commons project:

xml-commons/java/external/src/org/w3c
and all subdirectories

(Note: SAC (The Simple API for CSS) has been published under an older version of the W3C license. The original license file is LICENSE.sac.html.)

LICENSE.dom-documentation.txt covers all documentation from the W3C including the following items in the xml-commons project:

xml-commons/java/external/xdocs/dom
and all subdirectories

The actual DOM Java Language Binding classes in xml-commons came from:
http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/java-binding.html

The specification of DOM Level 3’s various parts is at:
http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/
http://www.w3.org/TR/2004/REC-DOM-Level-3-LS-20040407/
http://www.w3.org/TR/2004/NOTE-DOM-Level-3-XPath-20040226/

The specification of DOM Level 2’s various parts is at:
http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113/
http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113/
http://www.w3.org/TR/2000/REC-DOM-Level-2-Traversal-Range-20001113/
http://www.w3.org/TR/2000/REC-DOM-Level-2-Views-20001113/

The specification of DOM Level 1’s various parts is at:
Links to all available W3C DOM Java Bindings can be found at:
http://www.w3.org/DOM/DOMTR

The actual classes of The Simple API for CSS (SAC) came from:
http://www.w3.org/Style/CSS/SAC/
http://www.w3.org/2002/06/sacjava-1.3.zip

The actual DOM Java Language Binding classes for SMIL came from:
http://dev.w3.org/cvsweb/java/classes/org/w3c/dom/smil/
(both ElementTimeControl.java and TimeEvent.java were taken at revision 1.1)

The actual DOM Java Language Binding classes for SVG 1.1 came from:
http://www.w3.org/TR/SVG11/java.html

========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache xml-commons xml-apis ==
== distribution. ==
========================================================================

Apache XML Commons XML APIs
Copyright 2006 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This license came from: http://www.w3.org/Consortium/Legal/copyright-software-20021231
W3C SOFTWARE NOTICE AND LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common
questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle $Date: 2005-06-03 18:49:13 -0400 (Fri, 03 Jun 2005) $

This license came from: http://www.w3.org/Consortium/Legal/copyright-documents-20021231

W3C DOCUMENT LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231

Public documents on the W3C site are provided by the copyright holders under the following license. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright [$date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.
THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO
REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT
LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE,
NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE
FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT
INFRINGEMENT ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE
PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising
or publicity pertaining to this document or its contents without specific,
written prior permission. Title to copyright in this document will at all
times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002.
This version removes the copyright ownership notice such that this license can
be used with materials other than those owned by the W3C, moves information on
style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is
now a host of the W3C, includes references to this specific dated version of
the license, and removes the ambiguous grant of "use". See the older
formulation for the policy prior to this date. Please see our Copyright FAQ for
common questions about using materials from our site, such as the translating
or annotating specifications. Other questions about this notice can be directed
to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle $Date: 2005-06-03 18:49:13 -0400 (Fri, 03 Jun 2005) $

1.1945 q 0.9.7 0.9.7
1.1945.1 Available under license :

Copyright 20092012 Kristopher Michael Kowal. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1946 swagger-annotations 1.5.0

1.1946.1 Available under license:

```java
/**
 * Copyright 2015 SmartBear Software
 * <p/>
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * <p/>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p/>
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package io.swagger.annotations;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
 * License metadata available within the info section of a Swagger definition, see
 * https://github.com/swagger-api/swagger-spec/blob/master/versions/2.0.md#licenseObject
 * @since 1.5.0
 */

@Target(ElementType.ANNOTATION_TYPE)
@Retention(RetentionPolicy.RUNTIME)
public @interface License {
```
1.1947 streams 2.7.0

1.1948 json-schema-validator 0.1.16

1.1948.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2016 Network New Technologies Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * You may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *       http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/TypeFactory.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/JsonValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/ValidationMessage.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/MaxItemsValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/ReadOnlyValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/DependenciesValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/TypeValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/MaximumValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/EnumValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/AnyOfValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/MinItemsValidator.java
* /opt/cola/permits/1001160735_1606872800.56/0/json-schema-validator-0-1-16-sources-jar/com/networknt/schema/MinLengthValidator.java

No license file was found, but licenses were detected in source scan.

~ Copyright (c) 2016 Network New Technologies Inc.
~
~ Licensed under the Apache License, Version 2.0 (the "License");
~ You may not use this file except in compliance with the License.
~ You may obtain a copy of the License at
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS,
~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
~ See the License for the specific language governing permissions and
~ limitations under the License.

Found in path(s):

1.1949 jackson-dataformat-yaml 2.9.9
Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses. This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivates work.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

`String.fromCodePoint` by Mathias Bynens used according to terms of MIT License, as follows:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1951 type-fest 0.8.1

1.1951.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Creative Commons Legal Code

CC0 1.0 Universal

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS DOCUMENT DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION OR WORKS PROVIDED HEREUNDER.

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

   i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;
ii. moral rights retained by the original author(s) and/or performer(s);
iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;
iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;
v. rights protecting the extraction, dissemination, use and reuse of data in a Work;
vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and
vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such
partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.
b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.
c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

1.1952 @babel/types 7.11.5

1.1952.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1953 gensync 1.0.0-beta.2

1.1953.1 Available under license:

Copyright 2018 Logan Smyth <loganfsmyth@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1954 jsreport 2.11.0

1.1954.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.
If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy’s public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.1955 dom-serializer 0.1.0 0.1.0

1.1955.1 Available under license:
License
(The MIT License)

Copyright (c) 2014 The cheeriojs contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ‘Software’), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1956 component-inherit 0.0.3 0.0.3

1.1956.1 Available under license:
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1957 lodash.isequal 4.5.0 4.5.0

1.1957.1 Available under license:

Copyright JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.1958 passport-http-bearer 1.0.1

1.1958.1 Available under license:

(The MIT License)

Copyright (c) 2011-2013 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1959 lodash-foreach 4.5.0

1.1959.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
1.1960 callsite 1.0.0 1.0.0

1.1960.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1961 handlebars 4.7.6

1.1961.1 Available under license:

Copyright (C) 2011-2019 by Yehuda Katz

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1962 @babel/runtime 7.12.1 7.12.1

1.1962.1 Available under license:
MIT License

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1963 react-pdf 3.0.6 3.0.6

1.1963.1 Available under license :
MIT License

Copyright (c) 20172018 Wojciech Maj

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1964 metrics-json 4.1.0

1.1965 dont-sniff-mimetype 1.0.0

1.1965.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014, 2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1966 @babel/plugin-transform-for-of 7.13.0
1.1966.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1967 content-disposition 0.5.1
1.1967.1 Available under license:
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1968 p-locate 4.1.0 4.1.0

1.1968.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1969 object-defaults 1.1.0
1.1969.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015, 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1970 debug 4.1.0

1.1970.1 Available under license:
MIT

1.1971 path-type 2.0.0

1.1971.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1972 expand-tilde 2.0.2
1.1972.1 Available under license :
   The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1973 content-security-policy-builder 1.0.0
1.1973.1 Available under license :
   The MIT License (MIT)

Copyright (c) 2015 Evan Hahn

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1974 detect-indent 4.0.0
1.1974.1 Available under license :
   The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1975 split-string 3.1.0
1.1975.1 Available under license :
   The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1976 negotiator 0.5.3 0.5.3
1.1976.1 Available under license :
(The MIT License)

Copyright (c) 2012-2014 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schluter
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1977 write-file-atomic 3.0.3
1.1977.1 Available under license:

Copyright (c) 2015, Rebecca Turner

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.1978 decompress-response 3.3.0

1.1978.1 Available under license:

`The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1979 fresh 0.5.2 0.5.2

1.1979.1 Available under license:

(The MIT License)

Copyright (c) 2012 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2016-2017 Douglas Christopher Wilson <doug@somethingdoug.com>
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1980 jetty 9.4.34.v20201102

1.1980.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient
3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial
Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the
remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.  
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.1981 woodstox-core 5.3.0

1.1981.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.1982 hibernate-validator 5.0.0.Final
1.1982.1 Available under license:

Found license 'GNU Lesser General Public License' in

'* This is free software; you can redistribute it and/or modify it * under the terms of the GNU Lesser General Public License as * published by the Free Software Foundation; either version 2.1 of * the License, or (at your option) any later version. * This software is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU * Lesser General Public License for more details. * You should have received a copy of the GNU Lesser General Public * Expert Group and released to the public domain, as explained at

1.1983 babel-helpers 6.24.1 6.24.1

1.1983.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1984 socket.io 2.3.0

1.1984.1 Available under license:

(The MIT License)

Copyright (c) 2014-2018 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the

Open Source Used In DNAC Platform Antelope 6971
'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1985 guava 30.1-jre

1.1985.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2010 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/Strings.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/annotations/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/Atomics.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/Equivalence.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/annotations/Beta.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/UncaughtExceptionHandlers.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/PatternCompiler.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/CommonMatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/CommonPattern.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/JdkPattern.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/GwtTransient.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AtomicLongMap.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/LinkedHashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/DirectedGraphConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/CollectCollectors.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/AbstractGraphBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/MapIteratorCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/graph/StandardMutableGraph.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {code Doubles}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/primitives/DoublesMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {code Ints}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/primitives/IntsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ImmutableSortedAsList.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/CollectSpliterators.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ImmutableBiMapFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/AbstractMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/AbstractIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/Sets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/Multimaps.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/BiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/Multisets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingConcurrentMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ReverseNaturalOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/SortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/NullsFirstOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/SingletonImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/EnumHashBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/Multiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/Ordering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/TreeMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/LinkedListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/AbstractSortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ConcurrentHashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/HashMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/AbstractSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/NullsLastOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/AbstractBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/AbstractListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/Multimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/Iterators.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/Interner.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/Synchronized.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ComparatorOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/CompoundOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/AbstractMapBasedMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingSortedMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/MutableClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/AbstractMapBasedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/UsingToStringOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/Iterables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/TreeMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/NaturalOrdering.java

Open Source Used In DNAC Platform Antelope 6980
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/Lists.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ExplicitOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/RegularImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/LinkedHashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ByFunctionOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/LinkingHashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/AbstractMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/MapDifference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/MapDifference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/EnumBiMap.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
*/
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
/*
 * Copyright (C) 2017 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableLongArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingLock.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingCondition.java

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Functions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CountingInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/LittleEndianDataOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteStreams.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2017 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Functions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CountingInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/LittleEndianDataOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteStreams.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/math/ToDoubleRounnder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/math/BigDecimalMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/hash/Java8Compatibility.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
 * may not use this file except in compliance with the License. You may
 * obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 * implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/Streams.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Internal.java

No license file was found, but licenses were detected in source scan.

*/
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/LittleEndianByteArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/FarmHashFingerprint64.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ConsumingQueueIterator.java

Open Source Used In DNAC Platform Antelope 6886
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
/*
* Copyright 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/IgnoreJRERequirement.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
 * and released to the public domain, as explained at
 * http://creativecommons.org/licenses/publicdomain
 * 
 * As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
 * java.util.HashMap class.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Striped.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
Open Source Used In DNAC Platform Antelope 6989

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractScheduledService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/ForwardingLoadingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Optional.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Ticker.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Present.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedLong.java
Open Source Used In DNAC Platform Antelope 6991

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/RemovalCause.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/Cache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Funnel.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Funnels.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Funnel.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/ForwardingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Queues.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/LongMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/DoubleUtils.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the
 * License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * either
 * express or implied. See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/LoadingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/BloomFilter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/HashFunction.java

Open Source Used In DNAC Platform Antelope 6992
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

/**
 * Returns an array containing all of the elements in the specified collection. This method
 * returns the elements in the order they are returned by the collection's iterator. The returned
 * array is "safe" in that no references to it are maintained by the collection. The caller is
 * thus free to modify the returned array.
 *
 * <p>This method assumes that the collection size doesn't change while the method is running.
 *
 * <p>TODO(kevinb): support concurrently modified collections?
 *
 * @param c the collection for which to return an array of elements
 */

Found in path(s):
*/
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

* This method was rewritten in Java from an intermediate step of the Murmur hash function in
* http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
* following header:
*
* MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
* hereby disclaims copyright to this source code.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/base/MoreObjects.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/ListenerCallQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/math/Quantiles.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2006 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/VisibleForTesting.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ServiceManagerBridge.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Java8Usage.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/HashingInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Runnables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Verify.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CharSequenceReader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/IndexedImmutableSet.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AllEqualOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TransformedListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedMultisetBridge.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompactLinkedHashSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/CompactLinkedHashMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/FilteredSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/CompactHashMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/ForwardingDeque.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/TransformedIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/TreeTraverser.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/DescendingMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/FilteredEntryMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/RangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/TreeRangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/DescendingImmutableSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-jre-sources-1-
jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/AbstractNavigableMap.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors.
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 The Guava Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/**
 * Holder for extra methods of [@code Objects} only in web. Intended to be empty for regular
 * version.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This following method is a modified version of one found in
 * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
 * which contained the following notice:
 *
 * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
 * the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
 *
 * Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
 */

Found in path(s):
*/
*/
*/
*/
*/
*/
*/

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2013 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MultimapBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/MoreFiles.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Tables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Serialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/PeekingIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SingletonImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableBiMap.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Shorts}. Intended to be empty for regular
* version.
*/

Found in path(s):
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2012 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/ImmutableRangeSet.java
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/LinearTransformation.java
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/CharSink.java
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LongAddable.java
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/SmoothRateLimiter.java
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/FilteredKeyMultimap.java
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/Parameter.java
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/Closer.java
*/ opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/LongAddables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/WriteMode.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/ClassPath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/xml/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ImmutableRangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ServiceManager.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/IncidentEdgeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/CompactHashing.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2017 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
  * distributed under the License is distributed on an "AS IS" BASIS,
  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  * See the License for the specific language governing permissions and
  * limitations under the License.
*/

No license file was found, but licenses were detected in source scan.

*/
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
  * distributed under the License is distributed on an "AS IS" BASIS,
  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  * See the License for the specific language governing permissions and
  * limitations under the License.
*/
/

/*
  This method was rewritten in Java from an intermediate step of the Murmur hash function in
  http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
  following header:
  *
  MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
  hereby disclaims copyright to this source code.
*/

No license file was found, but licenses were detected in source scan.
/*
 * Copyright (C) 2005 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/**
 * Outer class that exists solely to let us write { @code Partially.GwtIncompatible} instead of plain
 * { @code GwtIncompatible}. This is more accurate for { @link Futures#catching}, which is available
 * under GWT but with a slightly different signature.
 *
 * <p>We can't use { @code PartiallyGwtIncompatible} because then the GWT compiler wouldn't recognize
 * it as a { @code GwtIncompatible} annotation. And for { @code Futures.catching}, we need the GWT
 * compiler to autostrip the normal server method in order to expose the special, inherited GWT
 * version.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/SuccessorsFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/ImmutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/InsecureRecursiveDeleteException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/Graph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/RecursiveDeleteOption.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/MutableGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/Network.java

No license file was found, but licenses were detected in source scan.

/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/AtomicDoubleArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/primitives/Doubles.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/FluentIterable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/primitives/Chars.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except 
* in compliance with the License. You may obtain a copy of the License at 
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License 
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either 
* express 
* or implied. See the License for the specific language governing permissions and limitations under 
* the License.
*/
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Joiner.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Floats.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenableFutureTask.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Bytes.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/MultiReader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Converter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/internal/Finalizer.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/TrieParser.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/MinMaxPriorityQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/AbstractSequentialIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/RowSortedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingSortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ForwardingImmutableCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/UnmodifiableListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/SortedMapDifference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/SortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/DenseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingFuture.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedBytes.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/HostSpecifier.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/JdkFutureAdapters.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/UnsignedBytes.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/GwtIncompatible.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/SparseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/MapMakerInternalMap.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeResolver.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/MapMaker.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1119092615_1608716259_54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/Cut.java

Open Source Used In DNAC Platform Antelope 7013
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteArrayDataInput.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/SettableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Callables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteArrayDataOutput.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/html/HtmlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/LineProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/xml/XmlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/RegularImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/ArrayBasedCharEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/InternetDomainName.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/Escapers.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
 * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
 * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
 * version.
 *
 * @throws UnsupportedOperationException always
 * @deprecated <b>Pass a key of type {@code Comparable} to use {@link
 *     ImmutableSortedMap#of(Comparable, Object)}.</b>
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/coll/Immuta/Sort/MapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
 * Holder for web specializations of methods of {@code Floats}. Intended to be empty for regular
 * version.
 */

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License);
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/primitives/FloatsMethodsForWeb.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ComputationException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/DiscreteDomain.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableAsList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ArrayTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ForwardingTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/TableCollectors.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableEnumSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/RegularImmutableSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableSortedMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/SingletonImmutableList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableSortedSet.java
*/
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License, Version 2.0 (the "License");

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at [apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0)
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.1988 json5 1.0.1
1.1988.1 Available under license:

# The MIT License (MIT)

Copyright (c) 2017 Colin O'Dell <colinodell@gmail.com>

Based on https://github.com/json5/json5; Copyright (c) 2012-2016 Aseem Kishore, and others.

>` Permission is hereby granted, free of charge, to any person obtaining a copy
>` of this software and associated documentation files (the "Software"), to deal
>` in the Software without restriction, including without limitation the rights
>` to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
>` copies of the Software, and to permit persons to whom the Software is
>` furnished to do so, subject to the following conditions:
>`
>` The above copyright notice and this permission notice shall be included in
>` all copies or substantial portions of the Software.
>
>` THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
>` IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
>` FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
>` AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
>` LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
>` OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
>` THE SOFTWARE.

1.1989 de.huxhorn.lilith.data.converter 0.9.44
1.1989.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs
whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

/*
 * Lilith - a log event viewer.
 * Copyright (C) 2007-2011 Joern Huxhorn
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Lesser General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

/*
 * Copyright 2007-2011 Joern Huxhorn
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
1.1990 glibc 1.1.0

1.1990.1 Available under license:

Found license 'General Public License 2.0' in '# This program is free software; you can redistribute it and/or # modify it under the terms of the GNU General Public License as # published by the Free Software Foundation; either version 2 of # the License, or (at your option) any later version. # This program is distributed in the hope that it will be useful, # but WITHOUT ANY WARRANTY; without even the implied warranty of # or NON INFRINGEMENT. See the GNU General Public License for more'

1.1991 drill-druid-storage 1.19.0

1.1991.1 Available under license:

Drill : Contrib : Storage : Druid
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.1992 findbugs-annotations 1.3.9-1

1.1993 async 0.9.2
1.1993.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1994 jcommander 1.78
1.1994.1 Available under license :
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (C) 2010 the original author or authors.
 * See the notice.md file distributed with this work for additional
 * information regarding copyright ownership.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2.jar/com/beust/jcommander/IUsageFormatter.java
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2.jar/com/beust/jcommander/UnixStyleUsageFormatter.java
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2.jar/com/beust/jcommander/JCommander.java
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2.jar/com/beust/jcommander/DefaultUsageFormatter.java

No license file was found, but licenses were detected in source scan.

/**
 * Copyright (C) 2010 the original author or authors.
* See the notice.md file distributed with this work for additional
* information regarding copyright ownership.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/IStringConverterFactory.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/converters/URLConverter.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/internal/Lists.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/ParameterException.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/internal/DefaultConverterFactory.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/converters/IntegerConverter.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/converters/FileConverter.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/converters/StringConverter.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/IStringConverter.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/internal/Sets.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/converters/LongConverter.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/internal/Maps.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/converters/InetAddressConverter.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/Parameters.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-jar/com/beust/jcommander/ParameterDescription.java
* /opt/col/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (C) 2011 the original author or authors.
 * See the notice.md file distributed with this work for additional
 * information regarding copyright ownership.
 *
 * Licensed under the Apache License, Version 2.0 (the "License);
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-
jar/com/beust/jcommander/validators/NoValidator.java
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-
jar/com/beust/jcommander/validators/PositiveInteger.java
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-
jar/com/beust/jcommander/validators/NoValueValidator.java
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-
jar/com/beust/jcommander/IParameterValidator.java
* /opt/cola/permits/1135840359_1613613123.36/0/jcommander-1-78-sources-2-
jar/com/beust/jcommander/IParameterValidator2.java

1.1995 has-value 0.3.1
1.1995.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.1996 pinkie 2.0.4 2.0.4
1.1996.1 Available under license :
The MIT License (MIT)

Copyright (c) Vsevolod Strukchinsky <floatdrop@gmail.com> (github.com/floatdrop)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1997 jersey-client 2.5.1

1.1997.1 Available under license:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. "Contributor" means each individual or entity that
creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the
Original Software, prior Modifications used by a
Contributor (if any), and the Modifications made by that
particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or
(b) Modifications, or (c) the combination of files
containing Original Software with files containing
Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form
other than Source Code.

1.5. "Initial Developer" means the individual or entity
that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such
entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to
use, reproduce, modify, display, perform, sublicense
and distribute the Modifications created by such
Contributor (or portions thereof), either on an
unmodified basis, with other Modifications, as
Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making,
using, or selling of Modifications made by that
Contributor either alone and/or in combination with
its Contributor Version (or portions of such
combination), to make, use, sell, offer for sale,
have made, and/or otherwise dispose of: (1)
Modifications made by that Contributor (or portions
thereof); and (2) the combination of Modifications
made by that Contributor with its Contributor Version
(or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and
2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications
available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent
license is granted: (1) for any code that Contributor
has deleted from the Contributor Version; (2) for
infringements caused by: (i) third party
modifications of Contributor Version, or (ii) the
combination of Modifications made by that Contributor
with other software (except as part of the
Contributor Version) or other devices; or (3) under
Patent Claims infringed by Covered Software in the
absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make
available in Executable form must also be made available in
Source Code form and that Source Code form must be
distributed only under the terms of this License. You must
include a copy of this License with every copy of the
Source Code form of the Covered Software You distribute or
otherwise make available. You must inform recipients of any
such Covered Software in Executable form as to how they can
obtain such Covered Software in Source Code form in a
reasonable manner on or through a medium customarily used
for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a
different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer orContributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename
the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with
Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. §252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice
contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction’s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys’ fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

1.1998 commons-compress 1.2.2

1.1998.1 Available under license:

Copyright (c) 2014 Chris Talkington, contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1999 babel-eslint 6.1.2
1.1999.1 Available under license :

Copyright (c) 2014-2016 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2000 emoji-regex 7.0.3
1.2000.1 Available under license :

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2001 method-override 2.3.10 2.3.10
1.2001.1 Available under license :
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2002 codemirror 5.39.2 5.39.2
1.2002.1 Available under license :
MIT License

Copyright (C) 2017 by Marijn Haverbeke <marijnh@gmail.com> and others

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2003 commons-text 1.9

1.2003.1 Available under license:

Apache Commons Text
Copyright 2014-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2004 ee-first 1.1.1 1.1.1

1.2004.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2005 txw2 2.3.1

Found license 'General Public License 2.0' in 'Copyright (c) 2005-2018 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in '* Copyright (c) 2005-2017 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 2005-2018 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 2005-2017 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in '* Copyright (c) 2017 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL")
or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use
this file except in compliance with the License. You can * Oracle designates this particular file as subject to the
"Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License
Found license 'General Public License 2.0' in * Copyright (c) 2005-2017 Oracle and/or its affiliates. All rights
reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2
only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License // This
class is in the Public Domain. NO WARRANTY! * Public Domain, and comes with <strong>NO
WARRANTY</strong>.</em>
Found license 'General Public License 2.0' in * Copyright (c) 2005-2017 Oracle and/or its affiliates. All rights
reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2
only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License // NO
WARRANTY! This class is in the public domain.'

1.2006 babel-plugin-transform-es2015-
computed-properties 6.24.1 6.24.1
1.2006.1 Available under license :

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2007 once 1.4.0 1.4.0
1.2007.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2008 orc-shims 1.5.1

1.2008.1 Available under license:

ORC Shims
Copyright 2013-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
you may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2009 qs 4.0.0
1.2009.1 Available under license:

Copyright (c) 2014 Nathan LaFreniere and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote
  products derived from this software without specific prior written
  permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE
FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors

1.2010 regexp-quote 0.0.0 0.0.0
1.2010.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.2011 string_decoder 1.1.1 1.1.1

1.2011.1 Available under license:

Node.js is licensed for use as follows:

```

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
```

This license applies to parts of Node.js originating from the
https://github.com/joyent/node repository:

```

Copyright Joyent, Inc. and other Node contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2012 braces 3.0.2

1.2012.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2013 buffer-more-ints 0.0.2 0.0.2

1.2013.1 Available under license:

Copyright (c) 2012 David Wragg. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2014 jsesc 1.3.0 1.3.0

1.2014.1 Available under license:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2015 moment 2.22.2 2.22.2

1.2015.1 Available under license:
Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2016 encodeurl 1.0.2 1.0.2

1.2016.1 Available under license:
(The MIT License)

Copyright (c) 2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2017 junit-platform-launcher 1.7.0
1.2017.1 Available under license:
Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

### 1. Definitions

Contribution means:
* **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
* **b)** in the case of each subsequent Contributor:
  * **i)** changes to the Program, and
  * **ii)** additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)** distributing or **b)** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.
### 2. Grant of Rights

**a)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

**b)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

**c)** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

**d)** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

**e)** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

### 3. Requirements

**3.1** If a Contributor Distributes the Program in any form, then:

* **a)** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

* **b)** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
  * **i)** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  * **ii)** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
  * **iii)** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
  * **iv)** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.
**3.2** When the Program is Distributed as Source Code:

* **a)** it must be made available under this Agreement, or if the Program *(i)* is combined with other material in a separate file or files made available under a Secondary License, and *(ii)* the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and
* **b)** a copy of this Agreement must be included with each copy of the Program.

**3.3** Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

### 4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **a)** promptly notify the Commercial Contributor in writing of such claim, and **b)** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

### 5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

### 6. Disclaimer of Liability
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

#### Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}. 

---

Open Source Used In DNAC Platform Antelope 7070
Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses
====================

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.

1.2018 generate-function 2.3.1

1.2018.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2019 drill-format-xml 1.19.0
1.2019.1 Available under license:

Drill : Contrib : Format : XML
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2020 strip-bom 4.0.0
1.2020.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2021 js-base64 2.4.8 2.4.8
1.2021.1 Available under license :
Copyright (c) 2014, Dan Kogai
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of {{{project}}} nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2022 mocha 7.2.0
1.2022.1 Available under license:
(The MIT License)

Copyright (c) 2011-2020 OpenJS Foundation and contributors, https://openjsf.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
1.2023 elfutils 0.165 3ubuntu1.2

1.2023.1 Available under license:
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
to your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:
Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an
exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.
The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.


You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.

b) The work must carry prominent notices stating that it is
released under this License and any conditions added under section
7. This requirement modifies the requirement in section 4 to
"keep intact all notices".

c) You must license the entire work, as a whole, under this
License to anyone who comes into possession of a copy. This
License will therefore apply, along with any applicable section 7
additional terms, to the whole of the work, and all its parts,
regardless of how they are packaged. This License gives no
permission to license the work in any other way, but it does not
invalidate such permission if you have separately received it.
d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no
further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in
reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.
Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.
11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the
Program specifies that a certain numbered version of the GNU General
Public License “or any later version” applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy’s
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.
If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.
The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license
document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this
License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer
system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
Version.

e) Provide Installation Information, but only if you would otherwise
be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is
necessary to install and execute a modified version of the
Combined Work produced by recombining or relinking the
Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)


You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.2024 semver 6.3.0
1.2024.1 Available under license :

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2025 aggdesigner-algorithm 6.0

1.2026 mongodb-driver 3.12.4

1.2026.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1.jar/com/mongodb/client/internal/MongoChangeStreamCursorImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1.jar/com/mongodb/ReplicaSetStatus.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1.jar/com/mongodb/BulkWriteUpsert.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1.jar/com/mongodb/client/model/DBCollectionUpdateOptions.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1.jar/com/mongodb/client/jndi/MongoClientFactory.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1.jar/com/mongodb/gridfs/GridFSFindIterable.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1.jar/com/mongodb/client/internal/MongoltableImpl.java
Open Source Used In DNAC Platform Antelope 7104
Open Source Used In DNAC Platform Antelope 7105
Open Source Used In DNAC Platform Antelope 7106
Open Source Used In DNAC Platform Antelope 7107
/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/ClientSession.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8MapReduceIterableImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8AggregateIterableImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8ListDatabasesIterableImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/ClientSessionClock.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8ChangeStreamIterableImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/package-info.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/MongoIterableFactory.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8ListIndexesIterableImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8MongoIterableFactory.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/FallbackMongoIterableFactory.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/MongoIterables.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8FindIterableImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8ListCollectionsIterableImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8DistinctIterableImpl.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/TransactionBody.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/internal/Java8ForEachHelper.java
* /opt/cola/permits/1136447742_1613767064.98/0/mongodb-driver-3-12-4-sources-1-
  jar/com/mongodb/client/ClientSessionImpl.java
1.2027 accepts 1.2.13

1.2027.1 Available under license:

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2028 jsonwebtoken 7.4.3

1.2028.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Auth0, Inc. <support@auth0.com> (http://auth0.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2029 lodash.curry 4.1.1 4.1.1

1.2029.1 Available under license:
Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.
Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

**1.2030 drill-format-esri 1.19.0**

**1.2030.1 Available under license:**

Drill : Contrib : Format : Esri  
Copyright 2021 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (http://www.apache.org/).

Apache License  
Version 2.0, January 2004  
http://www.apache.org/licenses/

**TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION**

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "["]
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2031 drill-format-excel 1.19.0
1.2031.1 Available under license :

Drill : Contrib : Format : Excel
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.2032 babel-plugin-transform-object-rest-spread 6.26.0

1.2032.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE/licensing OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>

Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2033 path-to-regexp 0.1.7 0.1.7
1.2033.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2034 asn1 0.2.4 0.2.4
1.2034.1 Available under license:
Copyright (c) 2011 Mark Cavage, All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell...
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2035 is-unc-path 1.0.0

1.2035.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2036 websocket-api 9.3.27.v20190418

1.2037 to-fast-properties 1.0.3

1.2037.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Petka Antonov
2015 Sindre Sorhus

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.2038 string-width 1.0.2

1.2038.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2039 repeating 2.0.1

1.2039.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

---

Open Source Used In DNAC Platform Antelope 7128
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2040 after 0.8.2

1.2040.1 Available under license:

Copyright (c) 2011 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2041 on-headers 1.0.2

1.2041.1 Available under license:

(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND.
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2042 atob 2.1.2

1.2042.1 Available under license:
Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE
LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN
ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS
INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES
REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR
DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE
COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY
COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS
AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE
TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY
BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS
CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND
CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and
other pre-existing works, such as a translation, adaptation,
derivative work, arrangement of music or other alterations of a
literary or artistic work, or phonogram or performance and includes
cinematographic adaptations or any other form in which the Work may be
recast, transformed, or adapted including in any form recognizably
derived from the original, except that a work that constitutes a
Collection will not be considered an Adaptation for the purpose of
this License. For the avoidance of doubt, where the Work is a musical
work, performance or phonogram, the synchronization of the Work in
timed-relation with a moving image ("synching") will be considered an
b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the
License to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in
which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for
attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS,
WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified
here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at http://creativecommons.org/.

At your option you may choose either of the following licenses:

* The MIT License (MIT)
* The Apache License 2.0 (Apache-2.0)
The MIT License (MIT)

Copyright (c) 2015 AJ ONeal

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2015 AJ ONeal

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2043 has-flag 4.0.0

1.2043.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.2044 unescape 1.0.1

#### 1.2044.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014, 2016-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.2045 array-unique 0.3.2

#### 1.2045.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2046 slf4j 1.7.2
1.2046.1 Available under license :

Copyright (c) 2004-2008 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2047 busboy 0.2.14 0.2.14
1.2047.1 Available under license :

Copyright Brian White. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2048 strip-ansi 0.1.1 0.1.1

1.2048.1 Available under license :

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2049 babel-generator 6.26.1

1.2049.1 Available under license :

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
MIT License

Copyright (c) 2014-2017 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2015 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2050 jakarta xml bind api 2.3.2

1.2050.1 Available under license :

# Notices for Eclipse Project for JAXB

This content is produced and maintained by the Eclipse Project for JAXB project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaxb

## Trademarks

Eclipse Project for JAXB is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0 which is available at http://www.eclipse.org/org/documents/edl-v10.php.
## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jaxb-api

## Third-party Content

This project leverages the following third-party content.

None

## Cryptography

Content may contain encryption software. The country in which you are currently
may have restrictions on the import, possession, and use, and/or re-export to
another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the import,
possession, or use, and re-export of encryption software, to see if this is
permitted.

Copyright (c) 2017, 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its
  contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
1.2051 @babel/plugin-proposal-json-strings

7.13.8

1.2051.1 Available under license : MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2052 lcms 2.6 3ubuntu2.1

1.2052.1 Available under license : Little CMS

Copyright (c) 1998-2011 Marti Maria Saguer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

THIS LICENSE APPLIES ONLY TO iccjpeg.c file

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-2013, Thomas G. Lane, Guido Vollbeding. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from...
it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltmain.sh). Another support script, install-sh, is copyright by X Consortium but is also freely distributable.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that
"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

1.2053 source-map-support 0.4.18 0.4.18

1.2053.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Evan Wallace

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2054 get-stream 4.1.0

1.2054.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2055 jsoup 1.9.2

1.2055.1 Available under license:
The MIT License

2009-2016, Jonathan Hedley <jonathan@hedley.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
1.2056 sketches-core 0.9.0

1.2056.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-16, Yahoo! Inc.
 * Licensed under the terms of the Apache License 2.0. See LICENSE file at the project root for terms.
 */

/*
The MurmurHash3 is a fast, non-cryptographic, 128-bit hash function that has
* excellent avalanche and 2-way bit independence properties.
* </p>
* 
* <p>
The C++ MurmurHash3_x64_128(...), revision 150, of the MurmurHash3, written by Austin Appleby,
* which is in the Public Domain, was the inspiration for this implementation in Java.
* The C++ version can be found at
* <a href= "https://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp">SMHasher &amp;
* MurmurHash</a>.
* </p>
* 
* <p>
The java implementation pays close attention to the C++ algorithms in order to
* maintain bit-wise compatibility, but the design is quite different. This implementation has also
* been extended to include processing of arrays of longs, char or ints, which was not part of the
* original C++ implementation. This implementation produces the same exact output hash bits as
* the above C++ method given the same input.</p>
* 
* <p>In addition, with this implementation, the hash of byte[], char[], int[], or long[] will
* produce the same hash result if, and only if, all the arrays have the same exact length in
* bytes, and if the contents of the values in the arrays have the same byte endianness and
* overall order. There is a unit test for this class that demonstrates this.</p>
* 
* <p>The structure of this implementation also reflects a separation of code that is dependent on the
* input structure (in this case byte[], int[] or long[]) from code that is independent of the input
* structure. This also makes the code more readable and suitable for future extensions.
* </p>
* 
* @author Lee Rhodes
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016, Yahoo! Inc.
 * Licensed under the terms of the Apache License 2.0. See LICENSE file at the project root
 * for terms.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/package-info.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-17, Yahoo! Inc.
 * Licensed under the terms of the Apache License 2.0. See LICENSE file at the project root for terms.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/sampling/PreambleUtil.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/sampling/ReservoirLongsUnion.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/sampling/SamplingUtil.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/sampling/ReservoirItemsUnion.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/sampling/ReservoirSize.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/sampling/VersionConverter.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/sampling/ReservoirItemsSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/sampling/ReservoirLongsSketch.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016, Yahoo! Inc.
 * Licensed under the terms of the Apache License 2.0. See LICENSE file at the project root
 * for terms.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/frequencies/package-info.java

No license file was found, but licenses were detected in source scan.
/*
 * Copyright 2016-17, Yahoo! Inc.
 * Licensed under the terms of the Apache License 2.0. See LICENSE file at the project root
 * for terms.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/sampling/package-info.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016, Yahoo! Inc.
 * Licensed under the terms of the Apache License 2.0. See LICENSE file at the project root for terms.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/frequencies/ErrorType.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/frequencies/Util.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/quantiles/ItemsUtil.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/quantiles/DoublesSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/quantiles/ItemsSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/quantiles/PreambleUtil.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017, Yahoo! Inc. Licensed under the terms of the
 * Apache License 2.0. See LICENSE file at the project root for terms.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/quantiles/ItemsPmfCdfImpl.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/quantiles/ItemsMergeImpl.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/quantiles/ItemsByteArrayImpl.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-
jar/com/yahoo/sketches/quantiles/ItemsUpdateImpl.java

No license file was found, but licenses were detected in source scan.

Copyright 2015, Yahoo! Inc.
Licensed under the terms of the Apache License 2.0.
See LICENSE file at the project root for terms.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/META-INF/maven/com.yahoo.datasketches/sketches-core/pom.xml

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015-16, Yahoo! Inc.
* Licensed under the terms of the Apache License 2.0. See LICENSE file at the project root for terms.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/theta/package-info.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/quantiles/DoublesAuxiliary.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/quantiles/ItemsUnion.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/ArrayOfDoublesUnion.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/HeapArrayOfDoublesCompactSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/HeapArrayOfDoublesUnion.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/Sketches.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/SketchesArgumentException.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/SketchesArgumentOutOfRangeException.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/SketchesAutoBuffering.java
Open Source Used In DNAC Platform Antelope 7162

* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/DeserializeResult.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/ArrayOfStringsSerDe.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/hll/OnHeapHash.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/hll/DoubleSummaryFactory.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/theta/DirectCompactOrderedSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/theta/ArrayOfItemsSerDe.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/hll/OnHeapHash.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/theta/HeapAnotB.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/UpdatableSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/hll/Fields.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/UpdatableSketchBuilder.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/HeapArrayOfDoublesSketchIterator.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/HeapArrayOfDoublesSetOperationBuilder.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/UpdatableSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/hll/Fields.java
Open Source Used in DNAC Platform Antelope 7163
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/theta/HeapQuickSelectSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/SummaryFactory.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/hll/HllUtils.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/theta/AnotB.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/Sketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/theta/Sketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/DirectArrayOfDoublesSketchIterator.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/tuple/HeapArrayOfDoublesAnotB.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/quantiles/DoublesUnionBuilder.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/theta/UpdateSketch.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016, Yahoo! Inc. Licensed under the terms of the Apache License 2.0. See LICENSE file
* at the project root for terms.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/frequencies/LongsSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/frequencies/ReversePurgeItemHashMap.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/frequencies/ReversePurgeLongHashMap.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/frequencies/ItemsSketch.java
* /opt/ws_local/PERMITS_SQL/1066571456_1593690082.73/0/sketches-core-0-9-0-sources-jar/com/yahoo/sketches/frequencies/PreambleUtil.java

1.2057 aws-java-sdk-sts 1.11.837

1.2057.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
Licensed under the Apache License, Version 2.0 (the "License").
You may not use this file except in compliance with the License.
A copy of the License is located at

http://aws.amazon.com/apache2.0

or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
permissions and limitations under the License.

Copyright 2012-2020 Amazon Technologies, Inc.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at:

http://aws.amazon.com/apache2.0

This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and
limitations under the License.

Copyright 2011-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
Licensed under the Apache License, Version 2.0 (the "License").
You may not use this file except in compliance with the License.
A copy of the License is located at

http://aws.amazon.com/apache2.0

or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
 * with
 * the License. A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
 * permissions
 * and limitations under the License.
 */

/**
 * An IAM policy in JSON format that you want to use as an inline session policy.
 * 
 * You must pass an inline or managed <a
 * href="https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html#policies_session">session
 * policy</a>
 * to this operation. You can pass a single JSON policy document to use as an inline session policy. You can also
 * specify up to 10 managed policies to use as managed session policies.
 * 
 * This parameter is optional. However, if you do not pass any session policies, then the resulting federated user
 * session has no permissions.
 * 
 */
When you pass session policies, the session permissions are the intersection of the IAM user policies and the session policies that you pass. This gives you a way to further restrict the permissions for a federated user. You cannot use session policies to grant more permissions than those that are defined in the permissions policy of the IAM user. For more information, see Session Policies in the IAM User Guide.

The resulting credentials can be used to access a resource that has a resource-based policy. If that policy specifically references the federated user session in the Principal element of the policy, the session has the permissions allowed by the policy. These permissions are granted in addition to the permissions that are granted by the session policies.

The plain text that you use for both inline and managed session policies can't exceed 2,048 characters. The JSON policy characters can be any ASCII character from the space character to the end of the valid character list (through \u00FF). It can also include the tab (\t), linefeed (\n), and carriage return (\r) characters.

An AWS conversion compresses the passed session policies and session tags into a packed binary format that has a separate limit. Your request can fail for this limit even if your plain text meets the other requirements. The PackedPolicySize response element indicates by percentage how close the policies and tags for your request are to the upper size limit.

Found in path(s):

No license file was found, but licenses were detected in source scan.

Copyright 2015-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. A copy of the License is located at

http://aws.amazon.com/apache2.0

or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
permissions
* and limitations under the License.
*/
/**
 * <p>
 * Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a
 * security token) for a federated user. A typical use is in a proxy application that gets temporary security
 * credentials on behalf of distributed applications inside a corporate network. You must call the
 * <code>GetFederationToken</code> operation using the long-term security credentials of an IAM user. As a result,
 * this call is appropriate in contexts where those credentials can be safely stored, usually in a server-based
 * application. For a comparison of <code>GetFederationToken</code> with the other API operations that produce
 * </p>
 * <note>
 * <p>
 * You can create a mobile-based or browser-based app that can authenticate users using a web identity provider like
 * Login with Amazon, Facebook, Google, or an OpenID Connect-compatible identity provider. In this case, we
 * recommend that you use <a href="http://aws.amazon.com/cognito/">Amazon Cognito</a> or
 * <code>AssumeRoleWithWebIdentity</code>. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html#api_assumerolewithwebidentity">Federation Through a Web-based Identity Provider</a> in the <i>IAM User Guide</i>.
 * </p>
 * </note>
 * <p>
 * You can also call <code>GetFederationToken</code> using the security credentials of an AWS account root user, but
 * we do not recommend it. Instead, we recommend that you create an IAM user for the purpose of the proxy
 * application. Then attach a policy to the IAM user that limits federated users to only the actions and resources
 * that they need to access. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html">IAM Best Practices</a> in the <i>IAM User Guide</i>.
 * </p>
* The temporary credentials are valid for the specified duration, from 900 seconds (15 minutes) up to a maximum of 129,600 seconds (36 hours). The default session duration is 43,200 seconds (12 hours). Temporary credentials that are obtained by using AWS account root user credentials have a maximum duration of 3,600 seconds (1 hour).

* Permissions

You can use the temporary credentials created by `<code>GetFederationToken</code>` in any AWS service except the following:

* You cannot call any IAM operations using the AWS CLI or the AWS API.
* You cannot call any STS operations except `<code>GetCallerIdentity</code>`.

* You must pass an inline or managed session policy to this operation. You can pass a single JSON policy document to use as an inline session policy. You can also specify up to 10 managed policies to use as managed session policies. The plain text that you use for both inline and managed session policies can't exceed 2,048 characters.

* Though the session policy parameters are optional, if you do not pass a policy, then the resulting federated user session has no permissions. When you pass session policies, the session permissions are the intersection of the IAM user policies and the session policies that you pass. This gives you a way to further restrict the permissions for a federated user. You cannot use session policies to grant more permissions than those that are defined in the permissions policy of the IAM user. For more information, see Session Policies in the `<i>`IAM User Guide</i>`. For information about using `<code>GetFederationToken</code>` to create temporary security credentials, see `<a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html#api_getfederationtoken">GetFederationToken—Federation Through a Custom Identity Broker</a>.

Open Source Used In DNAC Platform Antelope 7170
* You can use the credentials to access a resource that has a resource-based policy. If that policy specifically
* references the federated user session in the <code>Principal</code> element of the policy, the session has the
* permissions allowed by the policy. These permissions are granted in addition to the permissions granted by the
* session policies.

* <p>
* <p>
* <b>Tags</b>
* <p>
* <p>
* (Optional) You can pass tag key-value pairs to your session. These are called session tags. For more information
* about session tags, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_session-tags.html">Passing
Session Tags in STS</a> in the <i>IAM User Guide</i>.

* <p>
* <p>
* An administrator must grant you the permissions necessary to pass session tags. The administrator can also
create
* granular permissions to allow you to pass only specific session tags. For more information, see <a
Using Tags for Attribute-Based Access Control</a> in the <i>IAM User Guide</i>.

* <p>
* <p>
* Tag key-value pairs are not case sensitive, but case is preserved. This means that you cannot have separate
* <code>Department</code> and <code>department</code> tag keys. Assume that the user that you are
federating has
* the <code>Department</code>=<code>Marketing</code> tag and you pass the <code>department</code>=
* <code>engineering</code> session tag. <code>Department</code> and <code>department</code> are not
saved as
* separate tags, and the session tag passed in the request takes precedence over the user tag.

* <p>
* *

* @param getFederationTokenRequest
* @return A Java Future containing the result of the GetFederationToken operation returned by the service.
* @sample AWSSecurityTokenServiceAsync.GetFederationToken
* @see <a href="http://docs.aws.amazon.com/goto/WebAPI/sts-2011-06-15/GetFederationToken" target="_top">AWS API</a>
* Documentation</p>
*/

Found in path(s):
* /opt/cola/permits/1137760099_1614121467.90/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceAsync.java

No license file was found, but licenses were detected in source scan.
* Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a
* security token) for a federated user. A typical use is in a proxy application that gets temporary security
* credentials on behalf of distributed applications inside a corporate network. You must call the
* <code>GetFederationToken</code> operation using the long-term security credentials of an IAM user. As a
* result,
* this call is appropriate in contexts where those credentials can be safely stored, usually in a server-based
* application. For a comparison of <code>GetFederationToken</code> with the other API operations that
* produce
* temporary credentials, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html">Requesting
* Temporary
* </p>
* <note>
* <p>
* You can create a mobile-based or browser-based app that can authenticate users using a web identity provider
* like
* Login with Amazon, Facebook, Google, or an OpenID Connect-compatible identity provider. In this case, we
* recommend that you use <a href="http://aws.amazon.com/cognito/">Amazon Cognito</a> or
* <code>AssumeRoleWithWebIdentity</code>. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html#api_assumerolewithwebidentity">Federation Through a Web-based Identity Provider</a> in the <i>IAM User Guide</i>. 
* </p>
* </note>
* You can also call <code>GetFederationToken</code> using the security credentials of an AWS account root 
* user, but
* we do not recommend it. Instead, we recommend that you create an IAM user for the purpose of the proxy application. Then attach a policy to the IAM user that limits federated users to only the actions and resources that they need to access. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html">IAM Best Practices</a> in the IAM User Guide.

**Session duration**

The temporary credentials are valid for the specified duration, from 900 seconds (15 minutes) up to a maximum of 129,600 seconds (36 hours). The default session duration is 43,200 seconds (12 hours). Temporary credentials that are obtained by using AWS account root user credentials have a maximum duration of 3,600 seconds (1 hour).

**Permissions**

You can use the temporary credentials created by <code>GetFederationToken</code> in any AWS service except the following:

- You cannot call any IAM operations using the AWS CLI or the AWS API.
- You cannot call any STS operations except <code>GetCallerIdentity</code>.
- You must pass an inline or managed session policy to this operation. You can pass a single JSON policy document to use as an inline session policy. You can also specify up to 10 managed policies to use as managed session policies. The plain text that you use for both inline and managed session policies can’t exceed 2,048 characters.
- Though the session policy parameters are optional, if you do not pass a policy, then the resulting federated user session has no permissions. When you pass session policies, the session permissions are the intersection of the...
IAM user policies and the session policies that you pass. This gives you a way to further restrict the
permissions for a federated user. You cannot use session policies to grant more permissions than those that are
defined in the permissions policy of the IAM user. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html#policies_session">Session Policies</a> in the <i>IAM User Guide</i>. For information about using <code>GetFederationToken</code> to create
temporary security credentials, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_request.html#api_getfederatortoken">GetFederationToken—Federation Through a Custom Identity Broker</a>. You can use the credentials to access a resource that has a resource-based policy. If that policy specifically references the federated user session in the <code>Principal</code> element of the policy, the session has the permissions allowed by the policy. These permissions are granted in addition to the permissions granted by the session policies.

You can pass tag key-value pairs to your session. These are called session tags. For more information about session tags, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_session-tags.html">Passing Session Tags in STS</a>. An administrator must grant you the permissions necessary to pass session tags. The administrator can also create granular permissions to allow you to pass only specific session tags. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_attribute-based-access-control.html">Tutorial: Using Tags for Attribute-Based Access Control</a> in the <i>IAM User Guide</i>. Tag key–value pairs are not case sensitive, but case is preserved. This means that you cannot have separate <code>Department</code> and <code>department</code> tag keys. Assume that the user that you are federating has the <code>Department</code>=<code>Marketing</code> tag and you pass the <code>department</code>=<code>engineering</code> session tag. <code>Department</code> and <code>department</code> are not saved as separate tags, and the session tag passed in the request takes precedence over the user tag.

@throws MalformedPolicyDocumentException

* @param getFederationTokenRequest
* @return Result of the GetFederationToken operation returned by the service.
* @throws MalformedPolicyDocumentException
The request was rejected because the policy document was malformed. The error message describes the specific error.

@throws PackedPolicyTooLargeException

The request was rejected because the total packed size of the session policies and session tags combined was too large. An AWS conversion compresses the session policy document, session policy ARNs, and session tags into a packed binary format that has a separate limit. The error message indicates by percentage how close the policies and tags are to the upper size limit. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_session-tags.html">Passing Session Tags in STS</a> in the <i>IAM User Guide</i>.

You could receive this error even though you meet other defined session policy and session tag limits. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_enable-regions.html">IAM and STS Entity Character Limits</a> in the <i>IAM User Guide</i>.

@throws RegionDisabledException

STS is not activated in the requested region for the account that is being asked to generate credentials. The account administrator must use the IAM console to activate STS in that region. For more information, see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_enable-regions.html">Activating and Deactivating AWS STS in an AWS Region</a> in the <i>IAM User Guide</i>.

@sample AWSSecurityTokenService.GetFederationToken

@see <a href="http://docs.aws.amazon.com/goto/WebAPI/sts-2011-06-15/GetFederationToken" target="_top">AWS API Documentation</a>

*/

Found in path(s):

* /opt/cola/permits/1137760099_1614121467.90/aws-java-sdk-sts-1-11-837-sources.jar/com/amazonaws/services/securitytoken/AWSSecurityTokenService.java
* /opt/cola/permits/1137760099_1614121467.90/aws-java-sdk-sts-1-11-837-sources.jar/com/amazonaws/services/securitytoken/AWSSecurityTokenServiceClient.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright 2013-2020 Amazon Technologies, Inc.

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

* http://aws.amazon.com/apache2.0

* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
/*

Found in path(s):
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/auth/policy/STSActions.java

No license file was found, but licenses were detected in source scan.

*/

* Copyright 2015-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
with
* the License. A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR
* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
permissions
* and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/AssumeRoleWithWebIdentityRequest.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/ExpiredTokenException.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/AssumeRoleWithWebIdentityResult.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/AssumeRoleWithSAMLResult.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/transform/GetCallerIdentityResultStaxUnmarshaller.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/transform/GetFederationTokenResult.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/transform/GetFederationTokenResultStaxUnmarshaller.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/transform/AssumeRoleWithWebIdentityResultStaxUnmarshaller.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/GetFederationTokenResult.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/GetFederationTokenResultStaxUnmarshaller.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/transform/InvalidIdentityTokenExceptionUnmarshaller.java

Open Source Used In DNAC Platform Antelope 7176
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/IDPRejectedClaimException.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/transform/RegionDisabledExceptionUnmarshaller.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/GetAccessKeyInfoRequest.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/GetCallerIdentityResult.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/transform/GetFederationTokenRequestMarshaller.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/services/securitytoken/model/transform/TagStaxUnmarshaller.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2011-2020 Amazon Technologies, Inc.
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* http://aws.amazon.com/apache2.0
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/auth/STSSessionCredentials.java
* /opt/cola/permits/1137760099_1614121467.9/0/aws-java-sdk-sts-1-11-837-sources-jar/com/amazonaws/auth/STSSessionCredentialsProvider.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
1.2058 junit 4.13.2

1.2058.1 Available under license:

JUnit

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.
“Contributor” means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

   a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

   b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

   c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

   d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS
A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages
and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the
terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.2059 pdfjs-dist 2.0.489
1.2059.1 Available under license:

%%Copyright:  ...............................................................
%%Copyright: Copyright 1990-2009 Adobe Systems Incorporated.
%%Copyright: All rights reserved.
%%Copyright:
%%Copyright: Redistribution and use in source and binary forms, with or
%%Copyright: without modification, are permitted provided that the
%%Copyright: following conditions are met:
%%Copyright:
%%Copyright: Redistributions of source code must retain the above
%%Copyright: copyright notice, this list of conditions and the following
%%Copyright: disclaimer.
%%Copyright:
%%Copyright: Redistributions in binary form must reproduce the above
%%Copyright: copyright notice, this list of conditions and the following
%%Copyright: disclaimer in the documentation and/or other materials
%%Copyright: provided with the distribution.
%%Copyright:
%%Copyright: Neither the name of Adobe Systems Incorporated nor the names
%%Copyright: of its contributors may be used to endorse or promote
%%Copyright: products derived from this software without specific prior
%%Copyright: written permission.
%%Copyright:
%%Copyright: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
%%Copyright: CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
%%Copyright: INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
%%Copyright: MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
%%Copyright: DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
%%Copyright: CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
%%Copyright: SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
%%Copyright: NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
%%Copyright: LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
%%Copyright: HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
%%Copyright: CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
%%Copyright: OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
%%Copyright: SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
%%Copyright: ................................................................

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2060 jnr-posix 3.0.44
1.2060.1 Available under license:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.
It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.
***** BEGIN LICENSE BLOCK *****
* Version: CPL 1.0/GPL 2.0/LGPL 2.1
*
* The contents of this file are subject to the Common Public
* License Version 1.0 (the "License"); you may not use this file
* except in compliance with the License. You may obtain a copy of
* the License at http://www.eclipse.org/legal/cpl-v10.html
*
* Software distributed under the License is distributed on an "AS
* IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or
* implied. See the License for the specific language governing
* rights and limitations under the License.
*
*
* Alternatively, the contents of this file may be used under the terms of
* either of the GNU General Public License Version 2 or later (the "GPL"),
* or the GNU Lesser General Public License Version 2.1 or later (the "LGPL"),
* in which case the provisions of the GPL or the LGPL are applicable instead
* of those above. If you wish to allow use of your version of this file only
* under the terms of either the GPL or the LGPL, and not to allow others to
* use your version of this file under the terms of the CPL, indicate your
* decision by deleting the provisions above and replace them with the notice
* and other provisions required by the GPL or the LGPL. If you do not delete
* the provisions above, a recipient may use your version of this file under
* the terms of any one of the CPL, the GPL or the LGPL.
***** END LICENSE BLOCK *****/
1.2061 big-integer 1.6.48
1.2061.1 Available under license:

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <http://unlicense.org>

1.2062 isarray 0.0.1 0.0.1
1.2062.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.2063 color-name 1.1.3

1.2063.1 Available under license:
The MIT License (MIT)
Copyright (c) 2015 Dmitry Ivanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2064 text-table 0.2.0

1.2064.1 Available under license:
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2065 proxy-from-env 1.1.0

1.2065.1 Available under license:
The MIT License

Copyright (C) 2016-2018 Rob Wu <rob@robwu.nl>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2066 jersey-mvc-freemarker 2.25.1

1.2066.1 Available under license:

Found license 'General Public License 2.0' in

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License

Found license 'General Public License 2.0' in

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License

Found license 'General Public License 2.0' in

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License
only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License’

1.2067 jersey-mvc 2.25.1

1.2067.1 Available under license:

Found license 'General Public License 2.0' in /* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved. */ The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License’

Found license 'General Public License 2.0' in # Copyright (c) 2013-2014 Oracle and/or its affiliates. All rights reserved. # The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License’

Found license 'General Public License 2.0' in /* Copyright (c) 2013-2015 Oracle and/or its affiliates. All rights reserved. */ The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License’

Found license 'General Public License 2.0' in /* Copyright (c) 2010-2014 Oracle and/or its affiliates. All rights reserved. */ The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License’

Found license 'General Public License 2.0' in /* Copyright (c) 2010-2015 Oracle and/or its affiliates. All rights reserved. */ The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License’

Found license 'General Public License 2.0' in /* Copyright (c) 2012-2015 Oracle and/or its affiliates. All rights reserved. */ The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License’

Found license 'General Public License 2.0' in /* Copyright (c) 2012-2015 Oracle and/or its affiliates. All rights reserved. */ The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License ("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License’

1.2068 decamelize 1.2.0

1.2068.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2069 sqlite 3.21.0
1.2069.1 Available under license :
   The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:

   May you do good and not evil.
   May you find forgiveness for yourself and forgive others.
   May you share freely, never taking more than you give.

1.2070 antlr 3.5.2
1.2070.1 Available under license :
   BSD licence
   http://antlr.org/license.html

1.2071 safe-buffer 5.1.1
1.2071.1 Available under license :
   The MIT License (MIT)

   Copyright (c) Feross Aboukhadijeh

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2072 is-fullwidth-code-point 3.0.0 3.0.0
1.2072.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2073 denque 1.5.0 1.5.0
1.2073.1 Available under license:
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this library except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2074 async 2.6.3

1.2074.1 Available under license:

Copyright (c) 2010-2018 Caolan McMahon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2075 hadoop 1.2.1

1.2075.1 Available under license:

JZlib 0.0.* were released under the GNU LGPL license. Later, we have switched over to a BSD-style license.

-----------------------------------------------------------------------------

Copyright (c) 2000,2001,2002,2003 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT,
INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

Copyright 2000-2007 Sun Microsystems, Inc. All rights reserved.

The contents of this file are subject to the terms of either the GNU
General Public License Version 2 only ("GPL") or the Common Development
and Distribution License("CDDL") (collectively, the "License"). You
may not use this file except in compliance with the License. You can obtain
a copy of the License at https://glassfish.dev.java.net/public/CDDL+GPL.html
or glassfish/bootstrap/legal/LICENSE.txt. See the License for the specific
language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each
file and include the License file at glassfish/bootstrap/legal/LICENSE.txt.
Sun designates this particular file as subject to the "Classpath" exception
as provided by Sun in the GPL Version 2 section of the License file that
accompanied this code. If applicable, add the following below the License
Header, with the fields enclosed by brackets [ ] replaced by your own
identifying information: "Portions Copyrighted [year]
[name of copyright owner]"

Contributor(s):

If you wish your version of this file to be governed by only the CDDL or
only the GPL Version 2, indicate your decision by adding "[Contributor]
elects to include this software in this distribution under the [CDDL or GPL
Version 2] license.” If you don’t indicate a single choice of license, a
recipient has the option to distribute your version of this file under
either the CDDL, the GPL Version 2 or to extend the choice of license to
its licensees as provided above. However, if you add GPL Version 2 code
and therefore, elected the GPL Version 2 license, then the option applies
only if the new code is made subject to such option by the copyright
holder.

--> */
/* Copyright (c) 1995-2000, The Hypersonic SQL Group.

* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
* Redistributions of source code must retain the above copyright notice, this
* list of conditions and the following disclaimer.
*
* Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
*
* Neither the name of the Hypersonic SQL Group nor the names of its
* contributors may be used to endorse or promote products derived from this
* software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP,
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* This software consists of voluntary contributions made by many individuals
* on behalf of the Hypersonic SQL Group.

* For work added by the HSQL Development Group:

* Copyright (c) 2001-2004, The HSQL Development Group
* All rights reserved.

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
* list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.

* Neither the name of the HSQL Development Group nor the names of its
* contributors may be used to endorse or promote products derived from this
* software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQldb.ORG,
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works
“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and

3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and
assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation.
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

/* Copyright (c) 1995-2000, The Hypersonic SQL Group.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 * *
 * Redistributions of source code must retain the above copyright notice, this
 * list of conditions and the following disclaimer.
 * *
 * Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 * *
 * Neither the name of the Hypersonic SQL Group nor the names of its
 * contributors may be used to endorse or promote products derived from this
 * software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP,
 * OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 * This software consists of voluntary contributions made by many individuals
 * on behalf of the Hypersonic SQL Group.
 *
 * For work added by the HSQL Development Group:
 *
 * Copyright (c) 2001-2004, The HSQL Development Group
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this
* list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* Neither the name of the HSQL Development Group nor the names of its
* contributors may be used to endorse or promote products derived from this
* software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG,
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:
/**
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL
 * nor the names of its contributors may be used to endorse or
 * promote products derived from this software without specific prior
 * written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
 * ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 *
 *
* Copyright (c) 2005, European Commission project OneLab under contract

* 034819 (http://www.one-lab.org)
* All rights reserved.

* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

  * Neither the name of the University Catholique de Louvain - UCL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

  
  
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
Any way out of the use of this software, even if advised of the possibility of such damage.

Apache MINA
Copyright 2007 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Please refer to each LICENSE.<component>.txt file for the license terms of the components that Apache MINA depends on.

Message logging is provided by the SLF4J library package, which is open source software, written by Ceki Glic, and copyright by SLF4J.ORG and QOS.ch. The original software is available from

http://www.slf4j.org/

Data compression support is provided by the JZLib library package, which is open source software, written by JCraft, and copyright by JCraft. The original software is available from

http://www.jcraft.com/jzlib/

Spring framework is provided by the Spring framework library package, which is open source software, written by Rod Johnson et al, and copyright by Springframework.org. The original software is available from

http://www.springframework.org/

OGNL is provided by the OGNL library package, which is open source software, written by Drew Davidson and Luke Blanshard. The original software is available from

http://www.ognl.org/

Thrift Software License

Copyright (c) 2006- Facebook, Inc.

Permission is hereby granted, free of charge, to any person or organization

obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor
(if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files
containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software you distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE
The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.
The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not
responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM
"AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY
COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE
PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL,
SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY
TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER
PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to
achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to
most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a
pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public
License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later
version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the
Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and
you are welcome to redistribute it under certain conditions; type `show c' for details.
The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.
Thrift Software License
Copyright (c) 2006- Facebook, Inc.

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2004-2007 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

OGNL is the creation of Luke Blanshard and Drew Davidson.

Copyright 1997-2003, Drew Davidson and Luke Blanshard. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the Drew Davidson nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<!-
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
-->

1.2076 has 1.0.3

1.2076.1 Available under license:

Copyright (c) 2013 Thiago de Arruda

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2077 saslprep 1.0.3 1.0.3

1.2077.1 Available under license:

Copyright (c) 2014 Dmitry Tsvettsikh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2078 hbase-shaded-miscellaneous 3.4.1
1.2078.1 Available under license:

Apache Commons Collections
Copyright 2001-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache HBase Relocated (Shaded) Third-party Miscellaneous Libs
Copyright 2017-2020 The Apache Software Foundation

Apache Commons CLI
Copyright 2001-2017 The Apache Software Foundation

1.2079 co 4.6.0
1.2079.1 Available under license:

(The MIT License)

Copyright (c) 2014 TJ Holowaychuk &lt;tj@vision-media.ca&gt;

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2080 lodash-flatten 4.4.0
1.2080.1 Available under license:

Copyright jQuery Foundation and other contributors &lt;https://jquery.org/&gt;

Based on Underscore.js, copyright Jeremy Ashkenas,
This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
1.2081 string-prototype-trimstart 1.0.4

1.2081.1 Available under license :

MIT License

Copyright (c) 2017 Khaled Al-Ansari

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2082 dnsjava 2.1.7

1.2082.1 Available under license :

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
1.2083 path-loader 1.0.10 1.0.10

1.2083.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Jeremy Whitlock

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2084 groovy-datetime 3.0.8

1.2084.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1218311999_1634926077.36/0/groovy-datetime-3-0-8-sources-
  jar/org/apache/groovy/datetime/extensions/DateTimeStaticExtensions.java
* /opt/cola/permits/1218311999_1634926077.36/0/groovy-datetime-3-0-8-sources-
  jar/org/apache/groovy/datetime/extensions/DateTimeExtensions.java

1.2085 netty-transport 4.0.48.Final
1.2085.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
/**
 * Handles or intercepts a {link ChannelInboundInvoker} or {link ChannelOutboundInvoker} operation, and
 * forwards it
 * to the next handler in a {link ChannelPipeline}.
 *
 * <h3>Sub-types</h3>
 *
 * <ul>
 * <li>{@link ChannelInboundHandlerAdapter} handles and intercepts inbound operations</li>
 * <li>{@link ChannelOutboundHandlerAdapter} handles and intercepts outbound operations</li>
 */

You will also find more detailed explanation from the documentation of each sub-interface on how an event is interpreted when it goes upstream and downstream respectively.

<h3>The context object</h3>

A `{@link ChannelHandler}` is provided with a `{@link ChannelHandlerContext}` object. A `{@link ChannelHandler}` is supposed to interact with the `{@link ChannelPipeline}` it belongs to via a context object. Using the context object, the `{@link ChannelHandler}` can pass events upstream or downstream, modify the pipeline dynamically, or store the information (using `{@link AttributeKey}`) which is specific to the handler.

<h3>State management</h3>

A `{@link ChannelHandler}` often needs to store some stateful information. The simplest and recommended approach is to use member variables:

```java
public interface Message {
    // your methods here
}

public class DataServerHandler extends `{@link SimpleChannelInboundHandler}`<Message>
{
    private boolean loggedIn;

    @Override
    public void channelRead0({@link ChannelHandlerContext} ctx, Message message) {
        `{@link Channel}` ch = e.getChannel();
        if (message instanceof LoginMessage) {
            authenticate((LoginMessage) message);
            loggedIn = true;
        } else if (message instanceof GetDataMessage) {
            if (loggedIn) {
                ch.write(fetchSecret((GetDataMessage) message));
            } else {
                fail();
            }
        }
    }
    ...
}
```

Because the handler instance has a state variable which is dedicated to one connection, you have to create a new handler instance for each new channel to avoid a race condition where a unauthenticated client can get
* the confidential information:

* <pre>
* // Create a new handler instance per channel.
* // See { @link ChannelInitializer#initChannel(Channel)}.
* public class DataServerInitializer extends { @link ChannelInitializer} &lt;{ @link Channel} &gt; { 
*     { @code @Override}
*     public void initChannel({ @link Channel} channel) { 
*         channel.pipeline().addLast("handler", &lt;b&gt;new DataServerHandler&lt;/b&gt;); 
*     }
* }
* </pre>

* <h4>Using { @link AttributeKey}</h4>

* Although it’s recommended to use member variables to store the state of a
* handler, for some reason you might not want to create many handler instances.
* In such a case, you can use { @link AttributeKey}s which is provided by
* { @link ChannelHandlerContext}:

* <pre>
* public interface Message {
*     // your methods here
* }
* 
* { @code @Sharable}
* public class DataServerHandler extends { @link SimpleChannelInboundHandler} &lt;Message&gt; { 
*     private final { @link AttributeKey} &lt;{ @link Boolean} &gt; auth = 
*         { @link AttributeKey#valueOf(String) AttributeKey.valueOf("auth");
* 
*     { @code @Override}
*     public void channelRead({ @link ChannelHandlerContext} ctx, Message message) {
*         { @link Attribute} &lt;{ @link Boolean} &gt; attr = ctx.attr(auth); 
*         { @link Channel} ch = ctx.channel(); 
*         if (message instanceof LoginMessage) {
*             authenticate((LoginMessage) o); 
*             &lt;b&gt;attr.set(true)&lt;/b&gt;;
*         } else (message instanceof GetDataMessage) {
*             if (&lt;b&gt;Boolean.TRUE.equals(attr.get())&lt;/b&gt;) {
*                 ch.write(fetchSecret((GetDataMessage) o));
*             } else {
*                 fail();
*             }
*         }
*     }
* }
* </pre>

* Now that the state of the handler is attached to the { @link ChannelHandlerContext}, you can add the
* same handler instance to different pipelines:
* <pre>
* public class DataServerInitializer extends @link ChannelInitializer.&lt;@link Channel&gt; {
* *
*     private static final DataServerHandler &lt;b&gt;SHARED&lt;/b&gt; = new DataServerHandler();
* *
*     &lt;code&gt;@Override&lt;/code&gt;
*     public void initChannel(&lt;@link Channel&gt; channel) {
*         channel.pipeline().addLast("handler", &lt;b&gt;SHARED&lt;/b&gt;);
*     }
* }
* </pre>
* *
* <h4>The &lt;code&gt;@Sharable&lt;/code&gt; annotation</h4>
* <p>
* In the example above which used an &lt;@link AttributeKey&gt;, you might have noticed the &lt;code&gt;@Sharable&lt;/code&gt; annotation.
* <p>
* If a &lt;@link ChannelHandler&gt; is annotated with the &lt;code&gt;@Sharable&lt;/code&gt; annotation, it means you can create an instance of the handler just once and add it to one or more &lt;@link ChannelPipeline&gt;’s multiple times without a race condition.
* <p>
* If this annotation is not specified, you have to create a new handler instance every time you add it to a pipeline because it has unshared state such as member variables.
* <p>
* This annotation is provided for documentation purpose, just like a href="http://www.javaconcurrencyinpractice.com/annotations/doc/"the JCIP annotations</a>.
* <p>
* <h3>Additional resources worth reading</h3>
* <p>
* Please refer to the &lt;@link ChannelHandler&gt;, and &lt;@link ChannelPipeline&gt; to find out more about inbound and outbound operations, what fundamental differences they have, how they flow in a pipeline, and how to handle the operation in your application.
* /

Found in path(s):
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/ChannelHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
* *
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
  * with the License. You may obtain a copy of the License at:
  *
  *   http://www.apache.org/licenses/LICENSE-2.0
  *
  * Unless required by applicable law or agreed to in writing, software
  * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
  * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
  * License for the specific language governing permissions and limitations
  * under the License.
  */

Found in path(s):
*/opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
jar/io/netty/channel/PendingWriteQueue.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
*/opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
jar/io/netty/channel/DefaultAddressedEnvelope.java
*/opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
jar/io/netty/channel/ChannelProgressiveFutureListener.java
*/opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
jar/io/netty/bootstrap/ChannelFactory.java
*/opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
jar/io/netty/channel/ChannelHandlerAdapter.java
*/opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
jar/io/netty/channel/ChannelOutboundBuffer.java
*/opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
jar/io/netty/channel/ChannelProgressivePromise.java
*/opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
jar/io/netty/channel/DefaultMessageSizeEstimator.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
Found in path(s):
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/oio/OioByteStreamChannel.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/CombinedChannelDuplexHandler.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/ChannelFutureListener.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/socket/package-info.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/socket/nio/NioSocketChannel.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/socket/nio/package-info.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/ChannelPipelineException.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/local/LocalChannel.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/group/ChannelGroupFuture.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/socket/DefaultDatagramChannelConfig.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/bootstrap/AbstractBootstrap.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/socket/nio/NioDatagramChannelConfig.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/ChannelException.java
Open Source Used In DNAC Platform Antelope  7265

* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/CompleteChannelFuture.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/ChannelDuplexHandler.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/VoidChannelPromise.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/ChannelOutboundHandler.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/ChannelInboundHandlerAdapter.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/oio/OioEventLoopGroup.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/oio/package-info.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/oio/OioSocke
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 * *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
/*

Found in path(s):
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/DefaultChannelHandlerContext.java
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/META-INF/maven/io.netty/netty-transport/pom.xml
No license file was found, but licenses were detected in source scan.

*/
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-jar/io/netty/channel/nio/SelectedSelectionKeySetSelector.java
No license file was found, but licenses were detected in source scan.
/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/ChannelPoolHandler.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/FixedChannelPool.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/ChannelPool.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/AbstractChannelPoolMap.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/AbstractChannelPoolHandler.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/ChannelPoolMap.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/ChannelHealthChecker.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/SimpleChannelPool.java
* /opt/cola/permits/1202446789_1631568416.78/0/netty-transport-4-0-48-final-sources-
  jar/io/netty/channel/pool/package-info.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

1.2086 glib 2.48.2 0ubuntu4.4

1.2086.1 Available under license:

This work may be reproduced and distributed in whole or in part, in any medium, physical or electronic, so as long as this copyright notice remains intact and unchanged on all copies. Commercial redistribution is permitted and encouraged, but you may not redistribute, in whole or in part, under terms more restrictive than those under which you received it. If you redistribute a modified or translated version of this work, you must also make the source code to the modified or translated version available in electronic form without charge. However, mere aggregation as part of a larger work shall not count as a modification for this purpose.

All code examples in this work are placed into the public domain, and may be used, modified and redistributed without restriction.

BECAUSE THIS WORK IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE WORK, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE WORK "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SHOULD THE WORK PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE WORK AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE WORK, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

PCRE LICENCE

Please see the file LICENCE in the PCRE distribution for licensing details.

End

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.
For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.
If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.
14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.2087 unicode-canonical-property-names-ecmascript 1.0.4

1.2087.1 Available under license:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2088 monorepo-symlink-test 0.0.0

1.2088.1 Available under license :
MIT License

Copyright (c) 2012 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2089 rc 1.2.8

1.2089.1 Available under license :
Copyright (c) 2013, Dominic Tarr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, 
this list of conditions and the following disclaimer in the documentation 
and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND 
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED 
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE 
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR 
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES 
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; 
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND 
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT 
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS 
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those 
of the authors and should not be interpreted as representing official policies, 
either expressed or implied, of the FreeBSD Project.

The MIT License

Copyright (c) 2011 Dominic Tarr

Permission is hereby granted, free of charge, 
to any person obtaining a copy of this software and 
associated documentation files (the "Software"), to 
deal in the Software without restriction, including 
without limitation the rights to use, copy, modify, 
merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom 
the Software is furnished to do so, 
subject to the following conditions:

The above copyright notice and this permission notice 
shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES 
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. 
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR 
ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, 
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE 
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License, Version 2.0

Copyright (c) 2011 Dominic Tarr

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2090 batik-script 1.13

1.2090.1 Available under license:

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/rhino/resources/messages.properties
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
No license file was found, but licenses were detected in source scan.
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# This file registers the interpreter factories for the
# script languages that Batik understands.
#
# $Id: org.apache.batik.script.InterpreterFactory 1733416 2016-03-03 07:07:13Z gadams $
# --------------------------------------------------------------
#
# ECMAScript
org.apache.batik.bridge.RhinoInterpreterFactory
#
# TCL
#org.apache.batik.script.jacl.JaclInterpreterFactory
#
# Python
#org.apache.batik.script.jpython.JPythonInterpreterFactory

Found in path(s):
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/InterpreterFactory.java
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/InterpreterException.java
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/ImportInfo.java
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/jacl/JaclInterpreter.java
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/rhino/RhinoClassShutter.java
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/InterpreterPool.java
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/Interpreter.java
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/rhino/Messages.java
* /opt/cola/permits/1146878633_1616810515.71/0/batik-script-1-13-sources-1-jar/org/apache/batik/script/rhino/RhinoClassLoader.java

1.2091 form-data 2.3.2 2.3.2
1.2091.1 Available under license:
Copyright (c) 2012 Felix Geisendrfer (felix@debuggable.com) and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2092 amqplib 0.5.6
1.2092.1 Available under license :
MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

1.2093 batik-util 1.13
1.2093.1 Available under license :

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/CSSConstants.java
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/SVGFeatureStrings.java
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/resources/Messages.java
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/io/UTF8Decoder.java
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/io/NormalizingReader.java
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/io/UTF8Decoder.java
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/io/NormalizingReader.java
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/ParsedURLDefaultProtocolHandler.java
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/org/apache/batik/util/ApplicationSecurityEnforcer.java

No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Opal Source Used In DNAC Platform Antelope 7291
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-
jar/org/apache/batik/util/RunnableQueue.java

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1135872869_1613620994.25/0/batik-util-1-13-sources-1-jar/META-
INF/maven/org.apache.xmlgraphics/batik-util/pom.xml

1.2094 jackson-dataformat-yaml 2.4.2
1.2094.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing
Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses. This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

### 1.2095 leveldbjni-all 1.8

**1.2095.1 Available under license :**

Found license 'Eclipse Public License 1.0' in 'All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0'

### 1.2096 bcrypt-pbkdf 1.0.2

**1.2096.1 Available under license :**

The Blowfish portions are under the following license:

Blowfish block cipher for OpenBSD
Copyright 1997 Niels Provos <provos@physnet.uni-hamburg.de>
All rights reserved.

Implementation advice by David Mazieres <dm@lcs.mit.edu>.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The bcrypt_pbkdf portions are under the following license:

Copyright (c) 2013 Ted Unangst <tedu@openbsd.org>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE including all implied warranties of merchantability and fitness. In no event shall the author be liable for any special, direct, indirect, or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Performance improvements (Javascript-specific):

Copyright 2016, Joyent Inc
Author: Alex Wilson <alex.wilson@joyent.com>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE including all implied warranties of merchantability and fitness. In no event shall the author be liable for any special, direct, indirect, or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.
1.2097 thrift 0.14.0

1.2097.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square

Open Source Used In DNAC Platform Antelope 7298
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.
7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

Open Source Used In DNAC Platform Antelope 7304
This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Tue Oct 24 12:28:44 CDT 2006

Copyright (c) <2006> <Martin J. Logan, Erlware>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software (OTP Base, fslib, G.A.S) and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This package was debianized by Thrift Developer's <dev@thrift.apache.org>.

This package and the Debian packaging is licensed under the Apache License, see `/usr/share/common-licenses/Apache-2.0`.

The following information was copied from Apache Thrift LICENSE file.

SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

The following files contain some portions of code contributed under the Thrift Software License (see doc/old-thrift-license.txt), and relicensed under the Apache 2.0 License:

compiler/cpp/Makefile.am
compiler/cpp/src/generate/t_cpp_generator.cc
compiler/cpp/src/generate/t_netstd_generator.cc
compiler/cpp/src/generate/t_erl_generator.cc
compiler/cpp/src/generate/t_hs_generator.cc
compiler/cpp/src/generate/t_java_generator.cc
compiler/cpp/src/generate/t_ocaml_generator.cc
compiler/cpp/src/generate/t_perl_generator.cc
compiler/cpp/src/generate/t_php_generator.cc
compiler/cpp/src/generate/t_py_generator.cc
compiler/cpp/src/generate/t_rb_generator.cc
compiler/cpp/src/generate/t_st_generator.cc
compiler/cpp/src/generate/t_xsd_generator.cc
compiler/cpp/src/main.cc
compiler/cpp/src/parse/t_field.h
compiler/cpp/src/parse/t_program.h
compiler/cpp/src/platform.h
compiler/cpp/src/thriftl.ll
compiler/cpp/src/thrifty.yy
lib/netstd/src/Protocol/TBinaryProtocol.cs
lib/netstd/src/Protocol/TField.cs
lib/netstd/src/Protocol/TLList.cs
lib/netstd/src/Protocol/TMap.cs
lib/netstd/src/Protocol/TMessage.cs
lib/netstd/src/Protocol/TMessageType.cs
lib/netstd/src/Protocol/TProtocol.cs
lib/netstd/src/Protocol/TProtocolException.cs
lib/netstd/src/Protocol/TProtocolFactory.cs
lib/netstd/src/Protocol/TProtocolUtil.cs
lib/netstd/src/Protocol/TSet.cs
lib/netstd/src/Protocol/TStruct.cs
lib/netstd/src/Protocol/TType.cs
lib/netstd/src/Server/TServer.cs
lib/netstd/src/Server/TSimpleServer.cs
lib/netstd/src/Server/TThreadPoolServer.cs
lib/netstd/src/TApplicationException.cs
lib/netstd/src/Thrift.csproj
lib/netstd/src/Thrift.sln
lib/netstd/src/TPProcessor.cs
lib/netstd/src/Transport/TServerSocket.cs
lib/netstd/src/Transport/TServerTransport.cs
lib/netstd/src/Transport/TSocket.cs
lib/netstd/src/Transport/TStreamTransport.cs
lib/netstd/src/Transport/TTTransport.cs
lib/netstd/src/Transport/TTTransportException.cs
lib/netstd/src/Transport/TTTransportFactory.cs
lib/netstd/ThriftMSBuildTask/Properties/AssemblyInfo.cs
lib/netstd/ThriftMSBuildTask/ThriftMSBuildTask.csproj
lib/rb/lib/thrift.rb
lib/st/README
lib/st/thrift.st
test/OptionalRequiredTest.cpp
test/OptionalRequiredTest.thrift
test/ThriftTest.thrift

--------------------------------------------------
For the aclocal/ax_boost_base.m4 and contrib/fb303/aclocal/ax_boost_base.m4 components:

#   Copyright (c) 2007 Thomas Porschberg <thomas@randspringer.de>
#
#   Copying and distribution of this file, with or without
#   modification, are permitted in any medium without royalty provided
#   the copyright notice and this notice are preserved.

--------------------------------------------------
For the compiler/cpp/src/md5.[ch] components:

/*/  
Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved. 

This software is provided 'as-is', without any express or implied 
 warranty. In no event will the authors be held liable for any damages 
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, 
including commercial applications, and to alter it and redistribute it 
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not 
   claim that you wrote the original software. If you use this software 
in a product, an acknowledgment in the product documentation would be 
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be 
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch
ghost@aladdin.com

*/

--------------------------------------------------
For the lib/rb/setup.rb: Copyright (c) 2000-2005 Minero Aoki, 
lib/ocaml/OCamlMakefile and lib/ocaml/README-OCamlMakefile components: 
   Copyright (C) 1999 - 2007 Markus Mottl

Licensed under the terms of the GNU Lesser General Public License 2.1
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

For the aclocal/ax_boost_base.m4 and contrib/fb303/aclocal/ax_boost_base.m4 components:

# Copyright (c) 2007 Thomas Porschberg <thomas@randspringer.de>
#
# Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved.

For the lib/nodejs/lib/thrift/json_parse.js:

/*
   json_parse.js
   2015-05-02
   Public Domain.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications...
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
    distribution, then any Derivative Works that You distribute must
    include a readable copy of the attribution notices contained
    within such NOTICE file, excluding those notices that do not
    pertain to any part of the Derivative Works, in at least one
    of the following places: within a NOTICE text file distributed
    as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

For the aclocal/ax_boost_base.m4 and contrib/fb303/aclocal/ax_boost_base.m4 components:

# Copyright (c) 2007 Thomas Porschberg <thomas@randspringer.de>
#
# Copying and distribution of this file, with or without
# modification, are permitted in any medium without royalty provided
# the copyright notice and this notice are preserved.

For the lib/nodejs/lib/thrift/json_parse.js:

/*
 json_parse.js
 2015-05-02
 Public Domain.
 NO WARRANTY EXPRESSED OR IMPLIED. USE AT YOUR OWN RISK.
/* socketpair.c
 * Copyright 2007 by Nathan C. Myers <ncm@cantrip.org>; some rights reserved.
 * This code is Free Software. It may be copied freely, in original or
 * modified form, subject only to the restrictions that (1) the author is
 * relieved from all responsibilities for any use for any purpose, and (2)
 * this copyright notice must be retained, unchanged, in its entirety. If
 * for any reason the author might be held responsible for any consequences
 * of copying or use, license is withheld.
 */

// ISO C9x compliant stdint.h for Microsoft Visual Studio
// Based on ISO/IEC 9899:TC2 Committee draft (May 6, 2005) WG14/N1124
//
// Copyright (c) 2006-2008 Alexander Chemeris
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are met:
//
// 1. Redistributions of source code must retain the above copyright notice,
//    this list of conditions and the following disclaimer.
//
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
//
// 3. The name of the author may be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED
// WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
// MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
// EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
// PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
// OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
// WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
// OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
Codegen template in t_html_generator.h

* Bootstrap v2.0.3
*
* Copyright 2012 Twitter, Inc
* Licensed under the Apache License v2.0
* http://www.apache.org/licenses/LICENSE-2.0
*
* Designed and built with all the love in the world @twitter by @mdo and @fat.

For t_cl_generator.cc

* Copyright (c) 2008- Patrick Collison <patrick@collison.ie>
* Copyright (c) 2006- Facebook

---------------------------------------------

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2098 parquet-common 1.12.0

1.2098.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 */
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/LittleEndianDataOutputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/CapacityByteArrayOutputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/hadoop/metadata/Canonicalizer.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/HeapByteBufferAllocator.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/Log.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/ParquetRuntimeException.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/hadoop/metadata/ColumnPath.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/LittleEndianDataInputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/BytesUtils.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/Preconditions.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/DirectByteBufferAllocator.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/Ints.java

No license file was found, but licenses were detected in source scan.

*/

* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/IOExceptionUtils.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/SemanticVersion.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/BytesInput.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/ShouldNeverHappenException.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/SingleBufferInputSteam.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/glob/GlobNode.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/glob/GlobParser.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/Files.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/glob/GlobExpander.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/util/DynConstructors.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/util/DynMethods.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/Strings.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/ByteBufferInputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/glob/WildcardPath.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/ByteBufferAllocator.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/bytes/ConcatenatingByteArrayCollector.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/VersionParser.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/Closeables.java

No license file was found, but licenses were detected in source scan.
/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing.
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/io/PositionOutputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/io/InputFile.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/io/OutputFile.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/hadoop/codec/CompressionCodecNotSupportedException.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/compression/CompressionCodecFactory.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/io/SeekableInputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/io/SeekableInputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/hadoop/metadata/CompressionCodecName.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/Exceptions.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/io/DelegatingPositionOutputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/io/DelegatingSeekableInputStream.java
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-jar/org/apache/parquet/bytes/MultiBufferInputStream.java

No license file was found, but licenses were detected in source scan.

/**
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 */
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* <p/>
* http://www.apache.org/licenses/LICENSE-2.0
* <p/>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446201_1631568151.58/0/parquet-common-1-12-0-sources-
  jar/org/apache/parquet/OutputStreamCloseException.java

1.2099 neo-async 2.6.2
1.2099.1 Available under license:

MIT License

Copyright (c) 2014-2018 Suguru Motegi
Based on Async.js, Copyright Caolan McMahon

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2100 jersey-container-jetty-servlet 2.25.1
1.2100.1 Available under license:
Found license 'General Public License 2.0' in *
Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved. *
The contents of this file are subject to the terms of either the GNU *
General Public License Version 2 only ("GPL") or the Common Development *
and Distribution License("CDDL") (collectively, the "License"). You * may not use
this file except in compliance with the License. You can * Oracle designates this particular file as subject to the
"Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'

1.2101 anymatch 2.0.0
1.2101.1 Available under license:
The ISC License

Copyright (c) 2014 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2102 fastutil 6.5.7
1.2102.1 Available under license:
Found license 'GNU Lesser General Public License' in *
This library is free software; you can redistribute it and/or
modify it * under the terms of the GNU Lesser General Public License as published by the Free *
Software Foundation; either version 2.1 of the License, or (at your option) * any later version. * This library is distributed in
the hope that it will be useful, but * WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License *
for more details. * You should have received a copy of the GNU Lesser General Public License * along
with this program; if not, write to the Free Software'
Found license 'GNU Lesser General Public License' in *
This library is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public * License as published by the Free Software
Foundation; either * version 2.1 of the License, or (at your option) any later version. * This library is distributed in
the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of *
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU * Lesser General Public License for more details. * You should have received a copy of the GNU Lesser General Public
1.2103 @babel/helper-validator-identifier

7.12.11

1.2103.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2104 keyv 3.1.0

1.2104.1 Available under license:

MIT License

Copyright (c) 2017 Luke Childs

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2105 supports-color 7.2.0
1.2105.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
Copyright holders BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2106 merge-descriptors 1.0.1 1.0.1
1.2106.1 Available under license :
(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2107 hadoop-hdfs 3.2.2

1.2107.1 Available under license:

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of org.iq80.leventdb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of Gson 2.2.4,
which has the following notices:

The Netty Project
=================
Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh
Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of ‘Apache Harmony’, an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
  * HOMEPAGE:
    * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
  * HOMEPAGE:
    * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of ‘jbzip2’, a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
  * HOMEPAGE:
    * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
  * HOMEPAGE:
    * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
  * HOMEPAGE:
    * https://github.com/JCTools/JCTools
This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/
This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/
This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasternl.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===============================================================================
The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.  
Original source copyright:Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices: 
* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices: 
* ResolverUtil.java  
  Copyright 2005-2006 Tim Fennell  
  Dumbster SMTP test server  
  Copyright 2004 Jason Paul Kitchen  
  TypeUtil.java  
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of "Java Concurrency in Practice" book annotations 1.0, which has the following notices: 
* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)  
  Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of Jetty :: Http Utility 9.3.19., Jetty :: IO Utility 9.3.19., Jetty :: Security 9.3.19., Jetty :: Server Core 9.3.19., Jetty :: Servlet Handling 9.3.19., Jetty :: Utilities 9.3.19., Jetty :: Utilities :: Ajax, Jetty :: Webapp Application Support 9.3.19., Jetty :: XML utilities 9.3.19., which has the following notices: 
* =================================================================
  Jetty Web Container  
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
  =================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both
* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
  * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
  * org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
  * org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.
The UnixCrypt.java code implements the one-way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common http://hadoop.apache.org/ (Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by “GCC RUNtime Library Exception” http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html

== Contributors ==
* Tatu Saloranta
  * Providing benchmark suite
* Alec Wysoker
  * Performance and memory usage improvement

The binary distribution of this product bundles binaries of Xerces2 Java Parser 2.9.1, which has the following notices:
* == NOTICE file corresponding to section 4(d) of the Apache License. ==
* == Version 2.0, in this case for the Apache Xerces Java distribution. ==

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

The binary distribution of this product bundles binaries of
Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0
--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting
it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was
relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of
Phoenix Core 4.7.0,
which has the following notices:
   Apache Phoenix
   Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library
distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5
(https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann
The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-
ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-
commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
  which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1,
which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9
which has the following notices:
* ==============================================================
  = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
  ==============================================================
  This product includes software developed by
  Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.
This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 * Copyright (c) 2005, European Commission project OneLab under contract 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 * - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
* ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/
/*++
 *
For src/main/native/src/org/apache/hadoop/io/compress/lz4/\{lz4.h,lz4.c,lz4hc.h,lz4hc.c\}.

/*
LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at:
- LZ4 source repository : http://code.google.com/p/lz4/
- LZ4 public forum : https://groups.google.com/forum/#!forum/lz4c
*/

For hadoop-common-project/hadoop-common/src/main/native/gtest

----------------------------------
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1

---------------------------------------------------------------

(GO license)
This is a work derived from Russ Cox's RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the
distribution.

* Neither the name of Google Inc. nor the names of its contributors
may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h
---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni
(https://github.com/fusesource/leveldbjni), which is available under the
following license:
Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of FuseSource Corp. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS `AS IS` AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

============

The binary distribution of this product bundles binaries of leveldb
(http://code.google.com/p/leveldb/), which is available under the following
license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy
(http://code.google.com/p/snappy/), which is available under the following
license:
Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/
--------------------------------------------------------------------------------

Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
- hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
- hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css

- bootstrap v3.3.6
- broccoli-asset-rev v2.4.2
- broccoli-funnel v1.0.1
- datatables v1.10.8
- em-helpers v0.5.13
- em-table v0.1.6
- ember v2.2.0
- ember-array-contains-helper v1.0.2
- ember-bootstrap v0.5.1
- ember-cli v1.13.13
- ember-cli-app-version v1.0.0
- ember-cli-babel v5.1.6
- ember-cli-content-security-policy v0.4.0
- ember-cli-dependency-checker v1.2.0
- ember-cli-htmlbars v1.0.2
- ember-cli-htmlbars-inline-precompile v0.3.1
- ember-cli-ic-ajax v0.2.1
- ember-cli-inject-live-reload v1.4.0
- ember-cli-jquery-ui v0.0.20
- ember-cli-qunit v1.2.1
- ember-cli-release v0.2.8
- ember-cli-shims v0.0.6
- ember-cli-sri v1.2.1
- ember-cli-test-loader v0.2.1
- ember-cli-uglify v1.2.0
- ember-d3 v0.1.0
- ember-data v2.1.0
- ember-disable-proxy-controllers v1.0.1
- ember-export-application-global v1.0.5
- ember-load-initializers v0.1.7
- ember-qunit v0.4.16
- ember-qunit-notifications v0.1.0
- ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this
file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js
--------------------------------------------------------------------------------
The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js
--------------------------------------------------------------------------------
The MIT License
Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery
- Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
- Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- oj! Algorithms - version 43.0

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

--------------------------------------------------------------------------------
D3 is available under a 3-clause BSD license. For details, see:
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the
following license:
HSQLDB Database 2.3.4

(HSQL License)
"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.
Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the
Hypersonic SQL Group.”

The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14
--------------------------------------------------------------------------------
(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
1. Definitions.

1.1. Contributor means each individual or entity
that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the
Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity
that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means
having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable
form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the
Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent
Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b)
above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must
include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single
product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms
herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software...
9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
grizzly-http 2.2.21
1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of
such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted
in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications available to a third
party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2)
for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with
other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or
otherwise make available in Executable form must also be made available in
Source Code form and that Source Code form must be distributed only under the
terms of this License. You must include a copy of this License with every copy
of the Source Code form of the Covered Software You distribute or otherwise make
available. You must inform recipients of any such Covered Software in Executable
form as to how they can obtain such Covered Software in Source Code form in a
reasonable manner on or through a medium customarily used for software
exchange.

3.2. Modifications.
The Modifications that You create or to which
You contribute are governed by the terms of this License. You represent that You
believe Your Modifications are Your original creation(s) and/or You have
sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that
identifies You as the Contributor of the Modification. You may not remove or
alter any copyright, patent or trademark notices contained within the Covered
Software, or any notices of licensing or any descriptive text giving attribution
to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source
Code form that alters or restricts the applicable version of this License or the
recipients' rights hereunder. You may choose to offer, and to charge a fee for,
warranty, support, indemnity or liability obligations to one or more recipients
of Covered Software. However, you may do so only on Your own behalf, and not on
behalf of the Initial Developer or any Contributor. You must make it absolutely
clear that any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial Developer
and every Contributor for any liability incurred by the Initial Developer or
such Contributor as a result of warranty, support, indemnity or liability terms
You offer.

3.5. Distribution of Executable Versions.
You may distribute the
Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS
LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE
IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR
NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED
SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY
RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE
COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF
WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED
SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will
terminate automatically if You fail to comply with terms herein and fail to cure
such breach within 30 days of becoming aware of the breach. Provisions which, by
their nature, must remain in effect beyond the termination of this License shall
survive.

6.2. If You assert a patent infringement claim (excluding declaratory
judgment actions) against Initial Developer or a Contributor (the Initial
Developer or Contributor against whom You assert such claim is referred to as
Participant) alleging that the Participant Software (meaning the
Contributor Version where the Participant is a Contributor or the Original
Software where the Participant is the Initial Developer) directly or indirectly
infringes any patent, then any and all rights granted directly or indirectly to
You by such Participant, the Initial Developer (if the Initial Developer is not
the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this
License shall, upon 60 days notice from Participant terminate prospectively and
automatically at the expiration of such 60 day notice period, unless if within
such 60 day period You withdraw Your claim with respect to the Participant
Software against such Participant either unilaterally or pursuant to a written
agreement with Participant.

6.3. If You assert a patent infringement claim
against Participant alleging that the Participant Software directly or
indirectly infringes any patent where such claim is resolved (such as by license
or settlement) prior to the initiation of patent infringement litigation, then
the reasonable value of the licenses granted by such Participant under Sections
2.1 or 2.2 shall be taken into account in determining the amount or value of any
payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2
above, all end user licenses that have been validly granted by You or any
distributor hereunder prior to termination (excluding licenses granted to You by
any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT
(INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL
DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR
ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

--------------------------------------------------------------------------------

By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

--------------------------------------------------------------------------------

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution 'originates' from a
Contributor if it was added to the Program by such Contributor itself or anyone
acting on such Contributor's behalf. Contributions do not include additions to
the Program which: (i) are separate modules of software distributed in
conjunction with the Program under their own license agreement, and (ii) are not
derivative works of the Program.
"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when
combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free copyright license to
reproduce, prepare derivative works of, publicly display, publicly perform,
distribute and sublicense the Contribution of such Contributor, if any, and such
derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed
Patents to make, use, sell, offer to sell, import and otherwise transfer the
Contribution of such Contributor, if any, in source code and object code form.
This patent license shall apply to the combination of the Contribution and the
Program if, at the time the Contribution is added by the Contributor, such
addition of the Contribution causes such combination to be covered by the
Licensed Patents. The patent license shall not apply to any other combinations
which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to
its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other intellectual
property rights of any other entity. Each Contributor disclaims any liability to
Recipient for claims brought by any other entity based on infringement of
intellectual property rights or otherwise. As a condition to exercising the
rights and licenses granted hereunder, each Recipient hereby assumes sole
responsibility to secure any other intellectual property rights needed, if any.
For example, if a third party patent license is required to allow Recipient to
distribute the Program, it is Recipient's responsibility to acquire that license
before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright
rights in its Contribution, if any, to grant the copyright license set forth in
this Agreement.

3. REQUIREMENTS
A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any
related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.
All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2
--------------------------------------------------------------------------------
(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

**THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.**

The binary distribution of this product bundles these dependencies under the following license:

FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

---

(2-clause BSD)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

**THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.**

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies,
either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.
3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
to create and reproduce Derivative Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.
Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work (“cover version”) and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).
Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).
The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats.

All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or
publicly digitally perform the Work with any technological measures that control
access or use of the Work in a manner inconsistent with the terms of this
License Agreement. The above applies to the Work as incorporated in a Collective
Work, but this does not require the Collective Work apart from the Work itself
to be made subject to the terms of this License. If You create a Collective
Work, upon notice from any Licensor You must, to the extent practicable, remove
from the Collective Work any credit as required by clause 4(b), as requested. If
You create a Derivative Work, upon notice from any Licensor You must, to the
extent practicable, remove from the Derivative Work any credit as required by
clause 4(b), as requested.
If you distribute, publicly display, publicly perform, or publicly digitally
perform the Work or any Derivative Works or Collective Works, You must keep
intact all copyright notices for the Work and provide, reasonable to the medium
or means You are utilizing: (i) the name of the Original Author (or pseudonym,
if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor
designate another party or parties (e.g. a sponsor institute, publishing entity,
journal) for attribution in Licensor’s copyright notice, terms of service or by
other reasonable means, the name of such party or parties; the title of the Work
if supplied; to the extent reasonably practicable, the Uniform Resource
Identifier, if any, that Licensor specifies to be associated with the Work,
unless such URI does not refer to the copyright notice or licensing information
for the Work; and in the case of a Derivative Work, a credit identifying the use
of the Work in the Derivative Work (e.g., “French translation of the Work by
Original Author,” or “Screenplay based on original Work by Original Author”).
Such credit may be implemented in any reasonable manner; provided, however, that
in the case of a Derivative Work or Collective Work, at a minimum such credit
will appear where any other comparable authorship credit appears and in a manner
at least as prominent as such other comparable authorship credit.
5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS
THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND
CONCERNING
THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT
LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY,
OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME
JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH
EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN
NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL,
INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS
LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

7. Termination
This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

- jamon-runtime 2.4.1

(MPL 2.0)

Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this
License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can
obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any
4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This
disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the
License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:

JDOM 1.1

="/*

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

I. Redistributions of source code must retain the above copyright
notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4

-----------------------------------------------
This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and
independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public
by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise
by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g.,
"French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996.
and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.


Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2108 normalize-path 2.1.1
1.2108.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2109 winston 2.4.5
1.2109.1 Available under license:
Copyright (c) 2010 Charlie Robbins

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2110 posix-character-classes 0.1.1

1.2110.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2111 esprima 4.0.1

1.2111.1 Available under license:
Copyright JS Foundation and other contributors, https://js.foundation/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2112 binary-extensions 2.1.0
1.2112.1 Available under license :
MIT License

Copyright (c) 2019 Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com), Paul Miller
(https://paulmillr.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2113 harfbuzz 1.0.1-1ubuntu0.1
1.2113.1 Available under license :
The contents of this directory are licensed under the following terms:

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

HarfBuzz is licensed under the so-called "Old MIT" license. Details follow.
For parts of HarfBuzz that are licensed under different licenses see individual
files names COPYING in subdirectories where applicable.

Copyright 2012 Mozilla Foundation
Copyright 2011 Codethink Limited
Copyright 2008,2010 Nokia Corporation and/or its subsidiary(-ies)
Copyright 2009 Keith Stribley
Copyright 2009 Martin Hosken and SIL International
Copyright 2007 Chris Wilson
Copyright 2006 Behdad Esfahbod
Copyright 2005 David Turner
Copyright 1998-2004 David Turner and Werner Lemberg

For full copyright notices consult the individual files in the package.

Permission is hereby granted, without written agreement and without
license or royalty fees, to use, copy, modify, and distribute this
software and its documentation for any purpose, provided that the
above copyright notice and the following two paragraphs appear in
all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES
ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN
IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS
ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

CC BY-NC-SA 3.0
http://www.crlup.org/software/license/CreativeCommons.html
https://creativecommons.org/licenses/by-nc-sa/3.0/
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0
Upstream-Name: HarfBuzz
Upstream-Contact: Behdad Esfahbod
Source: http://www.freedesktop.org/wiki/Software/HarfBuzz

Files: *

2012,2013,2015 Mozilla Foundation
2011 Codethink Limited
2008,2010 Nokia Corporation and/or its subsidiary(-ies)
2009 Keith Stribley
2009 Martin Hosken and SIL International
2007 Chris Wilson
2006 Behdad Esfahbod
2005 David Turner
1998-2004 David Turner and Werner Lemberg

License: MIT

Files: debian/*

Copyright: 2012-2015 (Ahmed El-Mahmoudy) <aelmahmoudy@users.sourceforge.net>

License: MIT

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Copyright (c) 2010 Red Hat Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2114 express 4.13.4

1.2114.1 Available under license :
(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2013-2014 Roman Shtylman <shtylman+expressjs@gmail.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2115 lodash.isequal 4.5.0

1.2115.1 Available under license :
Copyright JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2116 big.js 5.2.2 5.2.2
1.2116.1 Available under license :
The MIT Licence (Expat).

Copyright (c) 2018 Michael Mclaughlin
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2117 jetty-servlets 9.4.17.v20190418

1.2117.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are
d) Each Contributor represents that to its knowledge it has sufficient
copyright rights in its Contribution, if any, to grant the copyright
license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under
its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties
      and conditions, express and implied, including warranties or
      conditions of title and non-infringement, and implied warranties or
      conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for
      damages, including direct, indirect, special, incidental and
      consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are
      offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such
      Contributor, and informs licensees how to obtain it in a reasonable
      manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained
within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore,
if a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and
other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such Commercial
Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

(a) promptly notify the Commercial Contributor in writing of such claim, and
(b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the
remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner.
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

================================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
================================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by
Unix systems for simple password protection. Copyright 1996 Aki Yoshida,
modified April 2001 by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt
for non-commercial or commercial purposes and without fee is
granted provided that the copyright notice appears in all copies.

1.2118 d 1.0.1
1.2118.1 Available under license :
ISC License

Copyright (c) 2013-2019, Mariusz Nowak, @medikoo, medikoo.com

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.
1.2119 commons-lang3 2.4

1.2119.1 Available under license:

Apache Commons Lang
Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2120 node-extend-without-arrays 1.1.6
1.2120.1 Available under license:

No license file was found, but licenses were detected in source scan.

/* *
 * node.extend
 * Copyright 2011, John Resig
 * Dual licensed under the MIT or GPL Version 2 licenses.
 * http://jquery.org/license
 *
 * @fileoverview
 * Port of jQuery.extend that actually works on node.js
 */

Found in path(s):
* /opt/cola/permits/1226014183_1636487640.98/0/node-extend-without-arrays-1-1-6-tgz/package/lib/extend.js

1.2121 jsreport-reports 2.5.1
1.2121.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified
version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:
0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may
Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

### 1.2122 buffers 0.1.1

*1.2122.1 Available under license :*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.2123 bluebird 3.4.7

*1.2123.1 Available under license :*

The MIT License (MIT)

Copyright (c) 2013-2015 Petka Antonov
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2124 fill-range 7.0.1

1.2124.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2125 hive-shims-common 3.1.2
1.2125.1 Available under license:

Hive Shims Common
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2126 regenerator-transform 0.10.1

1.2126.1 Available under license:

No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2014, Facebook, Inc.
 * All rights reserved.
 *
 * This source code is licensed under the BSD-style license found in the
 * https://raw.githubusercontent.com/facebook/regenerator/master/LICENSE file. An
 * additional grant of patent rights can be found in the PATENTS file in
 * the same directory.
 */

Found in path(s):
* /opt/cola/permits/1110631362_1606842466.21/0/regenerator-transform-0-10-1-5-tgz/package/lib/meta.js

No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2014, Facebook, Inc.
 * All rights reserved.
 *
 * This source code is licensed under the BSD-style license found in the
 * https://raw.githubusercontent.com/facebook/regenerator/master/LICENSE file. An
 * additional grant of patent rights can be found in the PATENTS file in
 * the same directory.
 */
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2014, Facebook, Inc.
 * All rights reserved.
 *
 * This source code is licensed under the BSD-style license found in the
 * https://raw.githubusercontent.com/facebook/regenerator/master/LICENSE file. An
 * additional grant of patent rights can be found in the PATENTS file in
 * the same directory.
 */

No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2014, Facebook, Inc.
 * All rights reserved.
 *
 * This source code is licensed under the BSD-style license found in the
 * https://raw.githubusercontent.com/facebook/regenerator/master/LICENSE file. An
 * additional grant of patent rights can be found in the PATENTS file in
 * the same directory.
 */

No license file was found, but licenses were detected in source scan.

**
 * Copyright (c) 2014, Facebook, Inc.
 * All rights reserved.
 *
 * This source code is licensed under the BSD-style license found in the
 * https://raw.githubusercontent.com/facebook/regenerator/master/LICENSE file. An
 * additional grant of patent rights can be found in the PATENTS file in
 * the same directory.
 */

Permission to use, copy, modify, and/or distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright notice

1.2127 ansi-align 3.0.0
1.2127.1 Available under license:
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright notice
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2128 run-async 2.4.1

1.2128.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Simon Boudrias

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2129 form-data 1.0.0-rc4

1.2129.1 Available under license:

Copyright (c) 2012 Felix Geisendrfer (felix@debuggable.com) and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2130 strip-ansi 5.2.0
1.2130.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2131 regenerate-unicode-properties 8.2.0
1.2131.1 Available under license :
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2132 tunnel-agent 0.6.0 0.6.0

1.2132.1 Available under license:
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices...
that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
1.2133.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
1.2134 brace-expansion 1.1.11 1.1.11

1.2134.1 Available under license :

MIT License

Copyright (c) 2013 Julian Gruber <julian@juliangruber.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2135 strip-bom 3.0.0

1.2135.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2136 fragment-cache 0.2.1
1.2136.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2137 commons-configuration2 2.1.1
1.2137.1 Available under license :

Apache Commons Configuration
Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

    Apache License
    Version 2.0, January 2004
    http://www.apache.org/licenses/

    TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.2138 xmlenc 0.52
1.2138.1 Available under license:
   BSD-2-Clause

1.2139 dns-prefetch-control 0.1.0
1.2139.1 Available under license:
   The MIT License (MIT)

Copyright (c) 2016 Evan Hahn

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2140 brace 0.11.1 0.11.1
1.2140.1 Available under license:
   Copyright 2013 Thorsten Lorenz.
   All rights reserved.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2141 aws-java-sdk-dynamodb 1.11.375

1.2141.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 * 
 * http://aws.amazon.com/apache2.0
 * 
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/PaginatedQueryList.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/S3Link.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBIndexRangeKey.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/S3ClientCache.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBScanExpression.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBScanExpression.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
Open Source Used In DNAC Platform Antelope 7453
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
 * /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/xspec/BetweenCondition.java
 * /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/xspec/ArrayIndexElement.java
 * /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/xspec/B.java
 * /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/xspec/AndCondition.java
 * /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/xspec/AddAction.java
 * /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/xspec/package-info.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/internal/IndexScanImpl.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/spec/GetItemSpec.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/internal/InternalUtils.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/IncompatibleTypeException.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/ItemCollection.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/internal/ScanImpl.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/TableKeysAndAttributes.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/TableCollection.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/utils/NameMap.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/spec/UpdateItemSpec.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/internal/QueryImpl.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/api/DeleteItemApi.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/document/internal/ListTablesPage.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
Open Source Used In DNAC Platform Antelope 7457
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2018 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 */
* http://aws.amazon.com/apache2.0

* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDB.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/AbstractEnumMarshaller.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* 
* http://aws.amazon.com/apache2.0
* 
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBMapper.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/AbstractDynamoDBMapper.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/IDynamoDBMapper.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights
* Reserved.
* 
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* 
* http://aws.amazon.com/apache2.0
*/
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180379099_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/datamodeling/BatchLoadContext.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180379099_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/AmazonDynamoDBClientConfigurationFactory.java
  /opt/cola/permits/1180379099_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBAutoGenerateStrategy.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */
Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBTypeConvertedTimestamp.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBAutoGeneratedDefault.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBTyped.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBMapperTableModel.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBMapperModelFactory.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/StandardTypeConverters.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBKeyed.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBTableMapper.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBTypeConverter.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBTypeConverterFactory.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBTypeConvertedEnum.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/ConvertibleType.java

Open Source Used In DNAC Platform Antelope 7461
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance
 * with
 * the License. A copy of the License is located at
 * 
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
 * permissions
 * and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/model/UpdateGlobalTableResult.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/model/AutoScalingTargetTrackingScalingPolicyConfigurationUpdate.jav
  a
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/model/TableStatus.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/model/transform/UpdateContinuousBackupsRequestMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/model/transform/BackupSummaryMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/model/transform/SourceTableDetailsMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/model/transform/UpdateGlobalTableRequestMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
  jar/com/amazonaws/services/dynamodbv2/model/transform/UpdateGlobalTableSettingsRequest.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
Marshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/ListTagsOfResourceResult.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/ReturnValue.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/StreamViewType.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/AutoScalingSettingsUpdateJsonUnmarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/PutItemRequestMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/LocalSecondaryIndexMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/SequenceNumberRangeMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/GlobalTableJsonUnmarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/PutRequestMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/QueryRequestProtocolMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/ProvisionedThroughputDescriptionJsonUnmarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/GlobalSecondaryIndexSettingsUpdateUnmarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/CreateTableRequestProtocolMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/AutoScalingPolicyUpdate.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/ListStreamsRequestMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/StreamDescriptionMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/StreamDescriptionMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/model/transform/StreamDescriptionMarshaller.java
<table>
<thead>
<tr>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/AttributeDefinitionMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/UpdateItemResultJsonUnmarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/DeleteItemResultJsonUnmarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/CreateGlobalTableRequestProtocolMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/AutoScalingPolicyDescriptionJsonUnmarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/ReplicaUpdateJsonUnmarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/CreateTableResult.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/GlobalSecondaryIndexInfoMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/ListTablesResultJsonUnmarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/BatchGetItemRequestMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/ProvisionedThroughputExceededException.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/RestoreSummaryMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/LocalSecondaryIndexJsonUnmarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/GetRecordsRequest.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/ConditionMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/GetItemRequestProtocolMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/KeySchemaElementMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/ReplicaAlreadyExistsException.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/CreateBackupRequestProtocolMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/ReplicaDescriptionMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/UpdateGlobalSecondaryIndexActionMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/PointInTimeRecoveryStatus.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
<tr>
<td>jar/com/amazonaws/services/dynamodbv2/model/transform/ReplicaSettingsUpdateMarshaller.java</td>
</tr>
<tr>
<td>* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-</td>
</tr>
</tbody>
</table>
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/ProvisionedThroughput.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/UpdateItemResult.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/ItemCollectionMetricsMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/ListGlobalTablesRequest.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/AmazonDynamoDBAsyncClient.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/AmazonDynamoDBStreamsClientBuilder.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/LocalSecondaryIndexInfoMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/BackupDescriptionMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/DeleteBackupRequest.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/DescribeTimeToLiveRequestMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/CreateGlobalTableResultJsonUnmarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/GetItemRequestMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/UpdateTableRequestProtocolMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/DescribeTimeToLiveRequestMarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/BackupDescription.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/TableDescription.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/CreateGlobalTableResultJsonUnmarshaller.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/model/transform/TableDescription.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/DynamoDBTypeConvertedEpochDate.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/AttributeTransformer.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/ScanResultPage.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/AttributeTransformerChain.java

/*
 * Copyright 2013-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/AttributeTransformer.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/ScanResultPage.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/datamodeling/AttributeTransformerChain.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with
 * the License. A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions
 * and limitations under the License.
 */

Found in path(s):

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013-2018 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/metrics/DynamoDBRequestMetric.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-jar/com/amazonaws/services/dynamodbv2/util/TableUtils.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014-2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
/*
 * Copyright 2015-2018 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 *    http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ OrCondition. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ NegationCondition. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ MinusOperation. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ datamodeling/ DynamoDBMappingsRegistry. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ NamedElement. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ PathElement. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ RemoveAction. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ IfNotExistsFunction. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ S. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ FunctionCondition. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ xspec/ ScanExpressionSpec. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ datamodeling/ unmarshallers/ NullableUnmarshaller. java
/ opt/ cola/ permits/ 1180370909_ 1626286006. 85/ 0/ aws-java-sdk-dynamodb-1- 11- 375- sources- 1- jar/ com/ amazonaws/ services/ dynamodbv2/ datamodeling/ marshalling/ CollectionToListMarshaller. java
jar/com/amazonaws/services/dynamodbv2/xspec/Condition.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/xspec/Operand.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/xspec/LiteralOperand.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/xspec/DeleteAction.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/xspec/UnitOfExpression.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/xspec/L.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/xspec/NULL.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/xspec/InCondition.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2016. Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/datamodeling/unmarshellers/package-info.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/datamodeling/marshallers/package-info.java
* /opt/cola/permits/1180370909_1626286006.85/0/aws-java-sdk-dynamodb-1-11-375-sources-1-
jar/com/amazonaws/services/dynamodbv2/datamodeling/marshallers/ObjectToStringMarshaller.java

1.2142 parquet-jackson 1.8.1

1.2142.1 Available under license:

Apache Parquet Jackson
Copyright 2015 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This product depends on Jackson and includes it in this binary artifact. Jackson is a high-performance JSON processor.

Copyright: 2007-2015 Tatu Saloranta and other contributors
Home page: http://jackson.codehaus.org/
License: http://www.apache.org/licenses/LICENSE-2.0.txt
1.2143 static-eval 2.0.2 2.0.2

1.2143.1 Available under license:
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2144 lodash-omit 4.5.0

1.2144.1 Available under license:
Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2145 @babel/plugin-transform-unicode-escapes 7.12.13

1.2145.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2146 yeast 0.1.2 0.1.2
1.2146.1 Available under license :
    The MIT License (MIT)

    Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

    Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

    The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2147 hbase-shaded-jersey 3.4.1
1.2147.1 Available under license :

    # Notice for Jersey
    This content is produced and maintained by the Eclipse Jersey project.

    * Project home: https://projects.eclipse.org/projects/ee4j.jersey

    ## Trademarks
    Eclipse Jersey is a trademark of the Eclipse Foundation.

    ## Copyright
All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

Angular JS, v1.6.6
* License MIT (http://www.opensource.org/licenses/mit-license.php)
* Project: http://angularjs.org
* Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1
* License: all the source code provided by AOP Alliance is Public Domain.
* Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2
* License: Apache License, 2.0
* Project: http://beanvalidation.org/1.1/
* Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
* by the @authors tag.

Hibernate Validator CDI, 6.1.2.Final
* License: Apache License, 2.0
* Project: https://beanvalidation.org/
* Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

Bootstrap v3.3.7
* License: MIT license (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* Project: http://getbootstrap.com
* Copyright: 2011-2016 Twitter, Inc
Google Guava Version 18.0
* License: Apache License, 2.0
* Copyright (C) 2009 The Guava Authors

javax.inject Version 1
* License: Apache License, 2.0
* Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA
* License: Apache License, 2.0
* Project: http://www.javassist.org/
* Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.10.1
* License: Apache License, 2.0
* Project: https://github.com/FasterXML/jackson-jaxrs-providers
* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4
* License: jquery.org/license
* Project: jquery.org
* Copyright: (c) jQuery Foundation

jQuery Barcode plugin 0.3
* Project: http://www.pasella.it/projects/jquery/barcode
* Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266
* License: CC0
* No copyright
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

KineticJS, v4.7.1
* License: MIT license (http://www.opensource.org/licenses/mit-license.php)
* Copyright: Eric Rowell

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0
* License: Apache License, 2.0
* Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.
org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents
* License: W3C License
* Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/
# Notice for Jersey Core Common module
This content is produced and maintained by the Eclipse Jersey project.

* https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

Google Guava Version 18.0
* License: Apache License, 2.0
* Copyright: (C) 2009 The Guava Authors
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
   Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.

Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that
results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby
assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:
a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to
pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.
Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part
thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by
the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.
## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

### libevent 1.3.6

1.2148.1 Available under license:

CMake - Cross Platform Makefile Generator
Copyright 2000-2013 Kitware, Inc.
Copyright 2000-2011 Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the names of Kitware, Inc., the Insight Software Consortium, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The above copyright and license notice applies to distributions of
CMake in source and binary form. Some source files contain additional
notices of original copyright by their contributors; see each source
for details. Third-party software packages supplied with CMake under
compatible licenses provide their own copyright notices documented in
corresponding subdirectories.

CMake was initially developed by Kitware with the following sponsorship:

* National Library of Medicine at the National Institutes of Health
  as part of the Insight Segmentation and Registration Toolkit (ITK).

* US National Labs (Los Alamos, Livermore, Sandia) ASC Parallel
  Visualization Initiative.

* National Alliance for Medical Image Computing (NAMIC) is funded by the
  National Institutes of Health through the NIH Roadmap for Medical Research,
  Grant U54 EB005149.

* Kitware, Inc.

Libevent is available for use under the following license, commonly known
as the 3-clause (or "modified") BSD license:

===============================================
Copyright (c) 2000-2007 Niels Provos <provos@citi.umich.edu>
Copyright (c) 2007-2012 Niels Provos and Nick Mathewson

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====================================

Portions of Libevent are based on works by others, also made available by
them under the three-clause BSD license above. The copyright notices are
available in the corresponding source files; the license is as above. Here's
a list:

log.c:
   Copyright (c) 2000 Dug Song <dugsong@monkey.org>
   Copyright (c) 1993 The Regents of the University of California.

strlcpy.c:
   Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

win32select.c:
   Copyright (c) 2003 Michael A. Davis <mike@datanerds.net>

evport.c:
   Copyright (c) 2007 Sun Microsystems

ht-internal.h:
   Copyright (c) 2002 Christopher Clark

minheap-internal.h:
   Copyright (c) 2006 Maxim Yegorushkin <maxim.yegorushkin@gmail.com>

=====================================

The arc4module is available under the following, sometimes called the
"OpenBSD" license:

Copyright (c) 1996, David Mazieres <dm@uun.org>
Copyright (c) 2008, Damien Miller <djm@openbsd.org>
Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

================================

The Windows timer code is based on code from libutp, which is distributed under this license, sometimes called the "MIT" license.

Copyright (c) 2010 BitTorrent, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2149 @nicolo-ribaudo/chokidar-2 2.1.8-no-fsevents

1.2149.1 Available under license:
No license file was found, but licenses were detected in source scan.

The MIT License (MIT)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal to use, copy, modify, merge, publish, distribute, sublicense, and/or sell furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1226013601_1636487720.9/0/chokidar-2-2-1-8-no-fsevents-4-tgz/package/chokidar/README.md

1.2150 esprima 4.0.1 4.0.1

1.2150.1 Available under license:
Copyright JS Foundation and other contributors, https://js.foundation/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2151 hppc 0.7.1

1.2151.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2010-2013, Carrot Search s.c., Boznicza 11/56, Poznan, Poland

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

ACKNOWLEDGEMENT
=================

HPPC borrowed code, ideas or both from:

  (Apache license)
* Fastutil, http://fastutil.di.unimi.it/
  (Apache license)
* Koloboke, https://github.com/OpenHFT/Koloboke
  (Apache license)

1.2152 yargs 15.4.1 15.4.1

1.2152.1 Available under license:

MIT License

Copyright 2010 James Halliday (mail@substack.net); Modified work Copyright 2014 Contributors
(ben@npmjs.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2153 pakko 1.0.11
1.2153.1 Available under license:
(The MIT License)

Copyright (C) 2014-2017 by Vitaly Puzrin and Andrei Tuputcyn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2154 jetty-continuation 9.3.27.v20190418
1.2154.1 Available under license:

No license file was found, but licenses were detected in source scan.

// All rights reserved. This program and the accompanying materials
// are made available under the terms of the Eclipse Public License v1.0
// and Apache License v2.0 which accompanies this distribution.
// The Eclipse Public License is available at
// The Apache License v2.0 is available at
// You may elect to redistribute this code under either of these licenses.

Found in path(s):
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/org/eclipse/jetty/continuation/ContinuationSupport.java
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/org/eclipse/jetty/continuation/Servlet3Continuation.java
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/org/eclipse/jetty/continuation/ContinuationThrowable.java
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/org/eclipse/jetty/continuation/Continuation.java
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/org/eclipse/jetty/continuation/FauxContinuation.java
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/org/eclipse/jetty/continuation/ContinuationListener.java
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/org/eclipse/jetty/continuation/ContinuationFilter.java
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/org/eclipse/jetty/continuation/package-info.java

No license file was found, but licenses were detected in source scan.

<p>The Eclipse Foundation makes available all content in this plug-in (&quot;Content&quot;). The Content is dual licensed and is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 (&quot;EPL&quot;) as well as the Apache Software License Version 2.0. A copy of the EPL is available at &lt;a href="http://www.eclipse.org/legal/epl-v10.html"&gt;http://www.eclipse.org/legal/epl-v10.html&lt;/a&gt;. A copy of the ASL is available at &lt;a href="http://www.apache.org/licenses/LICENSE-2.0.html"&gt;http://www.apache.org/licenses/LICENSE-2.0.html&lt;/a&gt;. For purposes of the EPL, &quot;Program&quot; will mean the Content.&lt;/p&gt;

Permission to use, copy, modify and distribute UnixCrypt granted provided that the copyright notice appears in all copies.&lt;/p&gt;

Found in path(s):
* /opt/cola/permits/1136447697_1613767081.95/0/jetty-continuation-9-3-27-v20190418-sources-1.jar/about.html

1.2155 get-intrinsic 1.1.1 1.1.1

1.2155.1 Available under license :

MIT License

Copyright (c) 2020 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2156 array-each 1.0.1

1.2156.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015, 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2157 base64-arraybuffer 0.1.4
1.2157.1 Available under license :

Copyright (c) 2012 Niklas von Hertzen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2158 hk2-locator 2.6.1

1.2158.1 Available under license :

# Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.
Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and
b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

   iii) does not attempt to limit or alter the recipients’ rights in the Source Code under section 3.2; and

   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

   a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

   b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits
and other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such
Commercial Contributor in connection with its distribution of the Program
in a commercial product offering. The obligations in this section do not
apply to any claims or Losses relating to any actual or alleged
intellectual property infringement. In order to qualify, an Indemnified
Contributor must: a) promptly notify the Commercial Contributor in
writing of such claim, and b) allow the Commercial Contributor to control,
and cooperate with the Commercial Contributor in, the defense and any
related settlement negotiations. The Indemnified Contributor may
participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial
product offering, Product X. That Contributor is then a Commercial
Contributor. If that Commercial Contributor then makes performance
claims, or offers warranties related to Product X, those performance
claims and warranties are such Commercial Contributor's responsibility
alone. Under this section, the Commercial Contributor would have to
defend claims against the other Contributors related to those performance
claims and warranties, and if a court requires any other Contributor to
pay any damages as a result, the Commercial Contributor must pay
those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT
PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS"
BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR
IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF
TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR
PURPOSE. Each Recipient is solely responsible for determining the
appropriateness of using and distributing the Program and assumes all
risks associated with its exercise of rights under this Agreement,
including but not limited to the risks and costs of program errors,
compliance with applicable laws, damage to or loss of data, programs
or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT
PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS
SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST
PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.
Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor  
Boston, MA 02110-1335  
USA  

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.

(Except: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will
automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR
DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM
(INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED
INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR
OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to
attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

---

## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

# Notices for Eclipse GlassFish

This content is produced and maintained by the Eclipse GlassFish project.

* Project home: https://projects.eclipse.org/projects/ee4j.glassfish

## Trademarks
Eclipse GlassFish, and GlassFish are trademarks of the Eclipse Foundation.

### Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

### Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

### Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/glassfish-ha-api
* https://github.com/eclipse-ee4j/glassfish-logging-annotation-processor
* https://github.com/eclipse-ee4j/glassfish-shoal
* https://github.com/eclipse-ee4j/glassfish-cdi-porting-tck
* https://github.com/eclipse-ee4j/glassfish-jsftemplating
* https://github.com/eclipse-ee4j/glassfish-hk2-extra
* https://github.com/eclipse-ee4j/glassfish-hk2
* https://github.com/eclipse-ee4j/glassfish-fighterfish

### Third-party Content

This project leverages the following third party content.

None

### Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>

Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2160 url-parse-lax 3.0.0

1.2160.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2161 package-hash 4.0.0

1.2161.1 Available under license:
ISC License (ISC)

Copyright (c) 2016-2017, Mark Wubben <mark@novemberborn.net> (novemberborn.net)

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF
This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.
2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are
offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such
Contributor, and informs licensees how to obtain it in a reasonable
manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained
within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore,
if a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and
other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such Commercial
Contributor in connection with its distribution of the Program in a commercial
product offering. The obligations in this section do not apply to any claims
or Losses relating to any actual or alleged intellectual property
infringement. In order to qualify, an Indemnified Contributor must:
a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the
Commercial Contributor in, the defense and any related settlement
negotiations. The Indemnified Contributor may participate in any such claim at
its own expense.

For example, a Contributor might include the Program in a commercial product
offering, Product X. That Contributor is then a Commercial Contributor. If
that Commercial Contributor then makes performance claims, or offers
warranties related to Product X, those performance claims and warranties are
such Commercial Contributor's responsibility alone. Under this section, the
Commercial Contributor would have to defend claims against the other
Contributors related to those performance claims and warranties, and if a
court requires any other Contributor to pay any damages as a result, the
Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

---

Open Source Used In DNAC Platform Antelope 7551
notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html
* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following
artifacts may be included in the distribution or downloaded when ALPN
module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications
are hosted at github and both modified and original are under GPL v2 with
classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html
-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2163 code-point-at 1.1.0

1.2163.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2164 is-regex 1.1.1

1.2164.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2165 resolve-from 5.0.0

1.2165.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2166 cookie-parser 1.4.3 1.4.3

1.2166.1 Available under license :
(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
1.2167 on-finished 2.3.0

1.2167.1 Available under license:
(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2168 jsreport-docx 2.9.0
1.2168.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2169 http-errors 1.3.1 1.3.1
1.2169.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2170 streamsearch 0.1.2 0.1.2

1.2170.1 Available under license:

Copyright Brian White. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2171 lodash 4.17.21 4.17.21

1.2171.1 Available under license:

Software License Agreement (BSD License)

Copyright (c) 2007, Parakey Inc.
All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other
materials provided with the distribution.

* Neither the name of Parakey Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of Parakey Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESSION OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own
licenses; we recommend you read them, as their terms may differ from the
terms above.
Copyright (c) 2010-2016 Jeremy Ashkenas, DocumentCloud

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESSION OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2009-2016 Jeremy Ashkenas, DocumentCloud and Investigative
Reporters & Editors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2172 fbemitter 2.1.1 2.1.1

1.2172.1 Available under license:

BSD License

For emitter software

Copyright (c) 2013-present, Facebook, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name Facebook nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
1.2173 groovy-xml 3.0.8

1.2173.1 Available under license:

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/package.html
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/slurpersupport/package.html
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/streamingmarkupsupport/package.html
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/dom/package.html
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/util/slurpersupport/package.html

No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/Namespace.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/FactorySupport.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/markupsupport/StandardXmlFilter.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/xmlslurpersupport/Attributes.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/markupsupport/DoubleQuoteFilter.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/Entity.groovy
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/slurpersupport/NamespacetAwareHashMap.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/xmlslurpersupport/FilteredNodeChildren.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/markupsupport/StandardControlToUndefined.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/xmlslurpersupport/Attributes.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/markupsupport/StreamingMarkupBuilder.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/markupsupport/SingleMarkupBuilder.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/markupsupport/SurrogateToUndefined.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/XMLParser.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/xmlslurpersupport/GPathResult.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/xmlslurpersupport/NodeChild.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/xmlslurpersupport/Node.java
* /opt/cola/permits/1218302256_1634930914.09/0/groovy-xml-3-0-8-sources-jar/groovy/xml/markupsupport/SurrogatetToUndefined.java
1.2174 config 1.3.3

1.2174.1 Available under license:

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.2175 acorn 2.7.0

1.2175.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2176 batik-svgrasterizer 1.13

1.2176.1 Available under license:

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-
jar/org/apache/batik/apps/rasterizer/resources/Messages.properties
No license file was found, but licenses were detected in source scan.

// -----------------------------------------------------------------------------
//
// Licensed to the Apache Software Foundation (ASF) under one or more
// contributor license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright ownership.
// The ASF licenses this file to You under the Apache License, Version 2.0
// (the "License"); you may not use this file except in compliance with
grant codeBase "${app.dev.base}/classes/" {
    permission java.security.AllPermission;
};

grant codeBase "${app.dev.base}/lib/xml-apis-dom3.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.dev.base}/lib/xerces_2_5_0.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.dev.base}/lib/fop-transcoder-allinone-${xmlgraphics.fop.version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.dev.base}/lib/xmlgraphics-commons-${xmlgraphics.commons.version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.dev.base}/lib/js.jar" {
    permission java.lang.RuntimePermission "createClassLoader";
    permission java.net.SocketPermission "+", "listen, connect, resolve, accept";
    permission java.lang.RuntimePermission "accessDeclaredMembers";
    permission java.util.PropertyPermission "org.mozilla/javascript.JavaAdapter", "read";
    permission java.util.PropertyPermission "org.mozilla/javascript.JavaAdapterFactory", "read";
    permission java.io.FilePermission ":<<ALL FILES>>:", "read";
};

grant {
    permission java.io.FilePermission "resources/org/apache/batik/dom/svg/resources/svg10.dtd", "read";
    permission java.io.FilePermission "resources/org/apache/batik/dom/svg/resources/UserAgentStyleSheet.css", "read";
};
Found in path(s):
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/resources/rasterizer.policy
No license file was found, but licenses were detected in source scan.

// -----------------------------------------------------------------------------
//
// Licensed to the Apache Software Foundation (ASF) under one or more
// contributor license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright ownership.
// The ASF licenses this file to You under the Apache License, Version 2.0
// (the "License"); you may not use this file except in compliance with
// the License. You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
//
// $Id: rasterizer.bin.policy 1855026 2019-03-08 09:57:56Z ssteiner $
// -----------------------------------------------------------------------------

grant codeBase "${app.jar.base}/lib/xml-apis-dom3.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/xerces_2_5_0.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-rasterizer.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-ext.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-dom.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-css.jar" {
    permission java.security.AllPermission;
};
grant codeBase "${app.jar.base}/lib/batik-svg-dom.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-gvt.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-parser.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-script.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-bridge.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-swing.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-transcoder.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/pdf-transcoder.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-gui-util.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-awt-util.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-util.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/batik-xml.jar" {
  permission java.security.AllPermission;
};
grant codeBase "${app.jar.base}/lib/xmlgraphics-commons-${xmlgraphics.commons.version}" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/lib/js.jar" {
    permission java.lang.RuntimePermission "createClassLoader";
    permission java.net.SocketPermission "*", "listen, connect, resolve, accept";
    permission java.lang.RuntimePermission "accessDeclaredMembers";
    permission java.util.PropertyPermission "org.mozilla.javascript.JavaAdapter", "read";
    permission java.util.PropertyPermission "org.mozilla.javascript.JavaAdapterClassName", "read";
    permission java.io.FilePermission "<<ALL FILES>>", "read";
};

grant {
    permission java.io.FilePermission "lib/batik-svgrasterizer", "read";
};

Found in path(s):
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/resources/rasterizer.bin.policy
No license file was found, but licenses were detected in source scan.

// -----------------------------------------------------------------------------
// // Licensed to the Apache Software Foundation (ASF) under one or more
// // contributor license agreements. See the NOTICE file distributed with
// // this work for additional information regarding copyright ownership.
// // The ASF licenses this file to You under the Apache License, Version 2.0
// // (the "License"); you may not use this file except in compliance with
// // the License. You may obtain a copy of the License at
// // http://www.apache.org/licenses/LICENSE-2.0
// // Unless required by applicable law or agreed to in writing, software
// // distributed under the License is distributed on an "AS IS" BASIS,
// // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// // See the License for the specific language governing permissions and
// // limitations under the License.
// // $Id: rasterizer.policy.ref 1733416 2016-03-03 07:07:13Z gadams $
// -----------------------------------------------------------------------------

grant codeBase "${app.jar.base}@jarsLibDir@xml-apis-ext-1.3.04.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}@jarsLibDir@xml-apis-1.3.04.jar" {

permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@xalan-2.7.0.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@xerces_2_5_0.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-rasterizer-@batikVersion@.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@batik-ext-@batikVersion@.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@batik-dom-@batikVersion@.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@batik-css-@batikVersion@.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@batik-svg-dom-@batikVersion@.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@batik-gvt-@batikVersion@.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@batik-parser-@batikVersion@.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@batik-bridge-@batikVersion@.jar" {
   permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/@jarsLibDir@batik-swing-@batikVersion@.jar" {
permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-transcoder-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-gui-util-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-awt-util-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-util-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-xml-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}fop-transcoder-allinone-${fopVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}xmlgraphics-commons-${xgcVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-anim-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-codec-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-extension-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}batik-svgen-${batikVersion}.jar" {
  permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/${jarsLibDir}js.jar" {
  permission java.security.AllPermission;
};
permission java.lang.RuntimePermission "createClassLoader";
permission java.net.SocketPermission "*", "listen, connect, resolve, accept";
permission java.lang.RuntimePermission "accessDeclaredMembers";
permission java.util.PropertyPermission "org.mozilla/javascript.JavaAdapter", "read";
permission java.util.PropertyPermission "org.mozilla/javascript.JavaAdapterClassName", "read";
permission java.io.FilePermission "<<ALL FILES>>", "read";
}

grant {
permission java.io.FilePermission "${app.jar.base}/@jarsLibDir@batik-svg-dom-@batikVersion@.jar", "read";
}

Found in path(s):
*/opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/resources/rasterizer.policy.ref
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/Main.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/SVGConverter.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/DestinationType.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/SVGConverterException.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/SVGConverterController.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/SVGConverterFileSource.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/DefaultSVGConverterController.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/SVGConverterSource.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/Messages.java
* /opt/cola/permits/1146690669_1616599310.58/0/batik-svgrasterizer-1-13-sources-1-jar/org/apache/batik/apps/rasterizer/SVGConverterURLSource.java

1.2177 has-binary 0.1.7
1.2177.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Kevin Roark

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2178 path-key 3.1.1

1.2178.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2179 groovy 3.0.8

1.2179.1 Available under license:
No license file was found, but licenses were detected in source scan.

/** Java 1.5 Recognizer
 *
 * Run `java Main [-showtree] directory-full-of-java-files`
 *
 * [The -showtree option pops up a Swing frame that shows the AST constructed from the parser.]
 *
 * Run `java Main <directory full of java files>'
 *
 * Contributing authors:
 * Jeremy Rayner groovy@ross-rayner.com
*John Mitchell johnm@non.net
*Terence Parr parrt@magelang.com
*John Lilley jilley@empathy.com
*Scott Stanchfield thetick@magelang.com
*Markus Mohnenmohnen informatik.rwth-aachen.de
*Peter Williams pete.williams@sun.com
*Allan Jacobs Allan.Jacobs@eng.sun.com
*Steve Messick messick@redhills.com
*John Pybus johnpybus.org

* Version 1.00 December 9, 1997 -- initial release
* Version 1.01 December 10, 1997
* fixed bug in octal def (0..7 not 0..8)
* Version 1.10 August 1998 (parrt)
* added tree construction
* fixed definition of WS, comments for mac, pc, unix newlines
* added unary plus
* Version 1.11 (Nov 20, 1998)
* Added "shutup" option to turn off last ambig warning.
* Fixed inner class def to allow named class defs as statements
* synchronized requires compound not simple statement
* add [] after builtInType DOT class in primaryExpression
* "const" is reserved but not valid.. removed from modifiers
* Version 1.12 (Feb 2, 1999)
* Changed LITERAL_xxx to xxx in tree grammar.
* Updated java.g to use tokens { ... } now for 2.6.0 (new feature).

* Version 1.13 (Apr 23, 1999)
* Didn't have (stat)? for else clause in tree parser.
* Didn't gen ASTs for interface extends. Updated tree parser too.
* Updated to 2.6.0.
* Version 1.14 (Jun 20, 1999)
* Allowed final/abstract on local classes.
* Removed local interfaces from methods
* Put instanceof precedence where it belongs... in relationalExpr
* It also had expr not type as arg: fixed it.
* Missing ! on SEMI in classBlock
* fixed: (expr) + "string" was parsed incorrectly (+ as unary plus).
* fixed: didn't like Object[].class in parser or tree parser
* Version 1.15 (Jun 26, 1999)
* Screwed up rule with instanceof in it. :( Fixed.
* Tree parser didn't like (expr).something; fixed.
* Allowed multiple inheritance in tree grammar. oops.
* Version 1.16 (August 22, 1999)
* Extending an interface built a wacky tree: had extra EXTENDS.
* Tree grammar didn't allow multiple superinterfaces.
* Tree grammar didn't allow empty var initializer: {}
* Version 1.17 (October 12, 1999)
*ESC lexer rule allowed 399 max not 377 max.
*java.tree.g didn't handle the expression of synchronized
*statements.
* Version 1.18 (August 12, 2001)
* Terence updated to Java 2 Version 1.3 by
*observing/combining work of Allan Jacobs and Steve
*Messick. Handles 1.3 src. Summary:
* o primary didn't include boolean.class kind of thing
* o constructor calls parsed explicitly now:
* see explicitConstructorInvocation
* o add strictfp modifier
* o missing objBlock after new expression in tree grammar
* o merged local class definition alternatives, moved after declaration
* o fixed problem with ClassName.super.field
* o reordered some alternatives to make things more efficient
* o long and double constants were not differentiated from int/float
* o whitespace rule was inefficient: matched only one char
* o add an examples directory with some nasty 1.3 cases
* o made Main.java use buffered IO and a Reader for Unicode support
* o supports UNICODE?
* Using Unicode charVocabulary makes code file big, but only
* in the bitsets at the end. I need to make ANTLR generate
* unicode bitsets more efficiently.
* Version 1.19 (April 25, 2002)
* Terence added in nice fixes by John Pybus concerning floating
* constants and problems with super() calls. John did a nice
* reorg of the primary/postfix expression stuff to read better
* and makes f.g.super() parse properly (it was METHOD_CALL not
* a SUPER_CTOR_CALL). Also:
* o "finally" clause was a root...made it a child of "try"
* o Added stuff for asserts too for Java 1.4, but *commented out*
* as it is not backward compatible.
*
* Version 1.20 (October 27, 2002)
* Terence ended up reorging John Pybus' stuff to
* remove some nondeterminisms and some syntactic predicates.
* Note that the grammar is stricter now; e.g., this(...) must
* be the first statement.
* Ternary ?: operator wasn't working as array name:
* (isBig ? bigDigits : digits)[i];
* Checked parser/tree parser on source for
* Resin-2.0.5, jive-2.1.1, jdk 1.3.1, Lucene, antlr 2.7.2a4,
* and the 110k-line jGuru server source.
* Version 1.21 (October 17, 2003)
  * Fixed lots of problems including:
  * Ray Waldin: add typeDefinition to interfaceBlock in java.tree.g
  * He found a problem/fix with floating point that start with 0
  * Ray also fixed problem that (int.class) was not recognized.
  * Thorsten van Ellen noticed that \n are allowed incorrectly in strings.
  * TJP fixed CHAR_LITERAL analogously.

  * Changes by Matt Quail to support generics (as per JDK1.5/JSR14)
  * Notes:
    * We only allow the "extends" keyword and not the "implements"
      keyword, since that's what JSR14 seems to imply.
    * Thanks to Monty Zukowski for his help on the antlr-interest
      mail list.
    * Thanks to Alan Eliasen for testing the grammar over his Fink source base

* Version 1.22 (July, 2004)
  * Changes by Michael Studman to support Java 1.5 language extensions
  * Notes:
    * Added support for annotations types
    * Finished off Matt Quail's generics enhancements to support bound type arguments
    * Added support for new for statement syntax
    * Added support for static import syntax
    * Added support for enum types
    * Tested against JDK 1.5 source base and source base of jdigraph project
    * Thanks to Matt Quail for doing the hard part by doing most of the generics work

* Version 1.22.1 (July 28, 2004)
  * Bug/omission fixes for Java 1.5 language support
    * Fixed tree structure bug with classOrInterface - thanks to Pieter Vangorpto for spotting this
    * Fixed bug where incorrect handling of SR and BSR tokens would cause type parameters to be recognised as type arguments.
    * Enabled type parameters on constructors, annotations on enum constants
    * and package definitions
    * Fixed problems when parsing if ((char.class.equals(c))) [ ] - solution by Matt Quail at Cenqua

* Version 1.22.2 (July 28, 2004)
  * Slight refactoring of Java 1.5 language support
    * Refactored for/"foreach" productions so that original literal "for" literal
      is still used but the for sub-clauses vary by token type
    * Fixed bug where type parameter was not included in generic constructor's branch of AST

* Version 1.22.3 (August 26, 2004)
  * Bug fixes as identified by Michael Stahl; clean up of tabs/spaces
    * and other refactorings
* o Fixed typeParameters omission in identPrimary and newStatement
* o Replaced GT reconciliation code with simple semantic predicate
* o Adapted enum/assert keyword checking support from Michael Stahl's java15 grammar
* o Refactored typeDefinition production and field productions to reduce duplication

* Version 1.22.4 (October 21, 2004)
* Small bux fixes
* o Added typeArguments to explicitConstructorInvocation, e.g. new <String>MyParameterised()
* o Added typeArguments to postfixExpression productions for anonymous inner class super
* constructor invocation, e.g. new Outer().<String>super()
* o Fixed bug in array declarations identified by Geoff Roy

* Version 1.22.4.j.1
* Changes by Jeremy Rayner to support java2groovy tool
* o I have taken java.g for Java1.5 from Michael Studman (1.22.4)
* and have made some changes to enable use by java2groovy tool (Jan 2007)

* Based on an original grammar released in the PUBLIC DOMAIN
*/

Found in path(s):
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/antlr/java/JavaRecognizer.java
No license file was found, but licenses were detected in source scan.

/*/  
* Copyright 2010 Google Inc. All Rights Reserved.
*  
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*  
* http://www.apache.org/licenses/LICENSE-2.0
*  
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/apache/groovy/util/concurrent/concurrentlinkedhashmap/EvictionListener.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/apache/groovy/util/concurrent/concurrentlinkedhashmap/Weighers.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/apache/groovy/util/concurrent/concurrentlinkedhashmap/Weigher.java
No license file was found, but licenses were detected in source scan.

*/
* Copyright 2012 Google Inc. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-
jar/org/apache/groovy/util/concurrent/concurrentlinkedhashmap/EntryWeigher.java
No license file was found, but licenses were detected in source scan.

*/
* This file is adapted from the Antlr4 Java grammar which has the following license
*
* Copyright (c) 2013 Terence Parr, Sam Harwell
* All rights reserved.
* [The "BSD licence"]
*
* http://www.opensource.org/licenses/bsd-license.php
*
* Subsequent modifications by the Groovy community have been done under the Apache License v2:
* *
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
lexer grammar GroovyLexer;

options {
    superClass = AbstractLexer;
}

@header {
    import java.util.Deque;
    import java.util.ArrayDeque;
    import java.util.Map;
    import java.util.HashMap;
    import java.util.Set;
    import java.util.HashSet;
    import java.util.Collections;
    import java.util.stream.IntStream;
    import java.util.logging.Logger;
    import java.util.logging.Level;
    import java.util.EmptyStackException;
    import org.apache.groovy.util.Maps;
    import static org.apache.groovy.parser.antlr4.SemanticPredicates.*;
}

@members {
    private static final Logger LOGGER = Logger.getLogger(GroovyLexer.class.getName());
    private long tokenIndex     = 0;
    private int  lastTokenType  = 0;
    private int  invalidDigitCount = 0;

    /**
     * Record the index and token type of the current token while emitting tokens.
     */
    @Override
    public void emit(Token token) {
        this.tokenIndex++;
        int tokenType = token.getType();
        if (Token.DEFAULT_CHANNEL == token.getChannel()) {
            int tokenType = token.getType();
            if (Token.DEFAULT_CHANNEL == token.getChannel()) {
                // This code block would continue here.
            }
        }
    }
}
this.lastTokenType = tokenType;
}

if (RollBackOne == tokenType) {
    this.rollbackOneChar();
}

super.emit(token);
}

private static final int[] REGEX_CHECK_ARRAY =
    IntStream.of(
        Identifier, CapitalizedIdentifier, NullLiteral, BooleanLiteral, THIS, RPAREN, RBRACK,
        RBRACE,
        IntegerLiteral, FloatingPointLiteral, StringLiteral, GStringEnd, INC, DEC
    ).sorted().toArray();

private boolean isRegexAllowed() {
    if (Arrays.binarySearch(REGEX_CHECK_ARRAY, this.lastTokenType) >= 0) {
        return false;
    }

    return true;
}

/**
 * just a hook, which will be overridden by GroovyLangLexer
 */
protected void rollbackOneChar() {}

private static class Paren {
    private String text;
    private int lastTokenType;
    private int line;
    private int column;

    public Paren(String text, int lastTokenType, int line, int column) {
        this.text = text;
        this.lastTokenType = lastTokenType;
        this.line = line;
        this.column = column;
    }

    public String getText() {
        return this.text;
    }

    public int getLastTokenType() {
public int getLine() {
    return line;
}

public int getColumn() {
    return column;
}

@Override
public int hashCode() {
    return (int) (text.hashCode() * line + column);
}

@Override
public boolean equals(Object obj) {
    if (!(obj instanceof Paren)) {
        return false;
    }
    Paren other = (Paren) obj;
    return this.text.equals(other.text) && (this.line == other.line && this.column == other.column);
}
}

protected void enterParenCallback(String text) {}  

protected void exitParenCallback(String text) {}  

private final Deque<Paren> parenStack = new ArrayDeque<>(32);

private void enterParen() {
    String text = getText();  
    enterParenCallback(text);  
    parenStack.push(new Paren(text, this.lastTokenType, getLine(), getCharPositionInLine()));
}

private void exitParen() {
    String text = getText();  
    exitParenCallback(text);  
    Paren paren = parenStack.peek();  
    if (null == paren) return;
    parenStack.pop();
private boolean isInsideParens() {
    Paren paren = parenStack.peek();

    // We just care about "(" and "[", inside which the new lines will be ignored.
    // Notice: the new lines between "{" and "}") can not be ignored.
    if (null == paren) {
        return false;
    }

    return ("(".equals(paren.getText()) && TRY != paren.getLastTokenType()) // we don't treat try-paren(i.e. try
    (....)) as parenthesis
        || "]".equals(paren.getText());
}

private void ignoreTokenInsideParens() {
    if (!this.isInsideParens()) {
        return;
    }

    this.setChannel(Token.HIDDEN_CHANNEL);
}

private void ignoreMultiLineCommentConditionally() {
    if (!this.isInsideParens() && isFollowedByWhiteSpaces(_input)) {
        return;
    }

    this.setChannel(Token.HIDDEN_CHANNEL);
}

@Override
public int getSyntaxErrorSource() {
    return GroovySyntaxError.LEXER;
}

@Override
public int getErrorLine() {
    return getLine();
}

@Override
public int getErrorColumn() {
    return getCharPositionInLine() + 1;
}

@Override
public int popMode() { try {
    return super.popMode();
}}
private static boolean isJavaIdentifierStartAndNotIdentifierIgnorable(int codePoint) {
    return Character.isJavaIdentifierStart(codePoint) && !Character.isIdentifierIgnorable(codePoint);
}

private static boolean isJavaIdentifierPartAndNotIdentifierIgnorable(int codePoint) {
    return Character.isJavaIdentifierPart(codePoint) && !Character.isIdentifierIgnorable(codePoint);
}

// 3.10.5 String Literals
StringLiteral
    : GStringQuotationMark DqStringCharacter* GStringQuotationMark
    | SqStringQuotationMark SqStringCharacter* SqStringQuotationMark
    | Slash { this.isRegexAllowed() && _input.LA(1) != '*' }? SlashyStringCharacter+ Slash
    | TdqStringQuotationMark TdqStringCharacter* TdqStringQuotationMark
    | TsqStringQuotationMark TsqStringCharacter* TsqStringQuotationMark
    | DollarSlashyGStringQuotationMarkBegin DollarSlashyStringCharacter+ DollarSlashyGStringQuotationMarkEnd
    ;

// Groovy gstring
GStringBegin
    : GStringQuotationMark DqStringCharacter* Dollar -> pushMode(DQ_GSTRING_MODE), pushMode(GSTRING_TYPE_SELECTOR_MODE)
    ;
TdqGStringBegin
    : TdqStringQuotationMark TdqStringCharacter* Dollar -> type(GStringBegin), pushMode(TDQ_GSTRING_MODE), pushMode(GSTRING_TYPE_SELECTOR_MODE)
    ;
SlashyGStringBegin
    : Slash { this.isRegexAllowed() && _input.LA(1) != '*' }? SlashyStringCharacter* Dollar { isFollowedByJavaLetterInGString(_input) }? -> type(GStringBegin), pushMode(SLASHY_GSTRING_MODE), pushMode(GSTRING_TYPE_SELECTOR_MODE)
    ;
DollarSlashyGStringBegin
: DollarSlashyGStringQuotationMarkBegin DollarSlashyStringCharacter* Dollar { isFollowedByJavaLetterInGString(_input) }? -> type(GStringBegin), pushMode(DOLLAR_SLASHY_GSTRING_MODE), pushMode(GSTRING_TYPE_SELECTOR_MODE)

mode DQ_GSTRING_MODE;
GStringEnd
: GStringQuotationMark -> popMode
;
GStringPart
: Dollar -> pushMode(GSTRING_TYPE_SELECTOR_MODE)
;
GStringCharacter
: DqStringCharacter -> more
;

mode TDQ_GSTRING_MODE;
TdqGStringEnd
: TdqStringQuotationMark -> type(GStringEnd), popMode
;
TdqGStringPart
: Dollar -> type(GStringPart), pushMode(GSTRING_TYPE_SELECTOR_MODE)
;
TdqGStringCharacter
: TdqStringCharacter -> more
;

mode SLASHY_GSTRING_MODE;
SlashyGStringEnd
: Dollar? Slash -> type(GStringEnd), popMode
;
SlashyGStringPart
: Dollar { isFollowedByJavaLetterInGString(_input) }? -> type(GStringPart), pushMode(GSTRING_TYPE_SELECTOR_MODE)
;
SlashyGStringCharacter
: SlashyStringCharacter -> more
;

mode DOLLAR_SLASHY_GSTRING_MODE;
DollarSlashyGStringEnd
: DollarSlashyGStringQuotationMarkEnd -> type(GStringEnd), popMode
;
DollarSlashyGStringPart
: Dollar { isFollowedByJavaLetterInGString(_input) }? -> type(GStringPart), pushMode(GSTRING_TYPE_SELECTOR_MODE)
;
DollarSlashyGStringCharacter

Open Source Used In DNAC Platform Antelope 7592
: DollarSlashyStringCharacter -> more
;
mode GSTRING_TYPE_SELECTOR_MODE;
GStringLBrace
: '{' { this.enterParen(); } -> type(LBRACE), popMode, pushMode(DEFAULT_MODE)
;
GStringIdentifier
: IdentifierInGString -> type(Identifier), popMode, pushMode(GSTRING_PATH_MODE)
;

mode GSTRING_PATH_MODE;
GStringPathPart
: Dot IdentifierInGString
;
RollBackOne
: .{
// a trick to handle GStrings followed by EOF properly
int readChar = _input.LA(-1);
if (EOF == _input.LA(1) && ('"' == readChar || '/' == readChar)) {
setType(GStringEnd);
} else {
setChannel(HIDDEN);
}
} -> popMode
;

mode DEFAULT_MODE;
// character in the double quotation string. e.g. "a"
fragment
DqStringCharacter
: ~["\r\n\\$]
| EscapeSequence
;
// character in the single quotation string. e.g. 'a'
fragment
SqStringCharacter
: ~['\r\n\\]
| EscapeSequence
;
// character in the triple double quotation string. e.g. """a"""
fragment TdqStringCharacter
: ~["\\$]
| GStringQuotationMark { _input.LA(1) != '"' || _input.LA(2) != '"' || _input.LA(3) == '"' && (_input.LA(4) != '"'

Open Source Used In DNAC Platform Antelope 7593


// character in the triple single quotation string. e.g. "a"
fragment TsqStringCharacter
  : ~[\n]
  | SqStringQuotationMark { _input.LA(1) != '\' || _input.LA(2) != '\' || _input.LA(3) == '\' && (_input.LA(4) != '\"' || _input.LA(5) != '\"' ) }?
  | EscapeSequence
  ;

// character in the slashy string. e.g. /a/
fragment SlashyStringCharacter
  : SlashEscape
  | Dollar { !isFollowedByJavaLetterInGString(_input) }?
  | ~[/SUu0000]
  ;

// character in the dollar slashy string. e.g. $/a/$
fragment DollarSlashyStringCharacter
  : DollarSlashEscape | DollarDollarEscape
  | Slash { _input.LA(1) != '$' }?
  | Dollar { !isFollowedByJavaLetterInGString(_input) }?
  | ~[/SUu0000]
  ;

// Groovy keywords
AS      : 'as';
DEF     : 'def';
IN      : 'in';
TRAIT   : 'trait';
THREADSAFE   : 'threadsafe'; // reserved keyword

// the reserved type name of Java10
VAR     : 'var';

// 3.9 Keywords
BuiltInPrimitiveType
  : BOOLEAN
  | CHAR
  | BYTE
  | SHORT
  | INT
  | LONG
  | FLOAT
  | DOUBLE
  ;
ABSTRACT : 'abstract';
ASSERT : 'assert';

fragment
BOOLEAN : 'boolean';

BREAK : 'break';

fragment
BYTE : 'byte';

CASE : 'case';
CATCH : 'catch';

fragment
CHAR : 'char';

CLASS : 'class';
CONST : 'const';
CONTINUE : 'continue';
DEFAULT : 'default';
DO : 'do';

fragment
DOUBLE : 'double';

ELSE : 'else';
ENUM : 'enum';
EXTENDS : 'extends';
FINAL : 'final';
FINALLY : 'finally';

fragment
FLOAT : 'float';

FOR : 'for';
IF : 'if';
GOTO : 'goto';
IMPLEMENTS : 'implements';
IMPORT : 'import';
INSTANCEOF : 'instanceof';

fragment
INT : 'int';

INTERFACE : 'interface';
fragment
LONG : 'long';

NATIVE : 'native';
NEW : 'new';
PACKAGE : 'package';
PRIVATE : 'private';
PROTECTED : 'protected';
PUBLIC : 'public';
RETURN : 'return';

fragment
SHORT : 'short';

STATIC : 'static';
STRICTFP : 'strictfp';
SUPER : 'super';
SWITCH : 'switch';
SYNCHRONIZED : 'synchronized';
THIS : 'this';
THROW : 'throw';
THROWS : 'throws';
TRANSIENT : 'transient';
TRY : 'try';
VOID : 'void';
VOLATILE : 'volatile';
WHILE : 'while';

// 3.10.1 Integer Literals

IntegerLiteral :
  ( DecimalIntegerLiteral
  | HexIntegerLiteral
  | OctalIntegerLiteral
  | BinaryIntegerLiteral
  ) (Underscore { require(false, "Number ending with underscores is invalid", -1, true); })?

  // !!! Error Alternative !!!
  | Zero ([0-9] { invalidDigitCount++; })+ { require(false, "Invalid octal number", -(invalidDigitCount + 1), true); }
} IntegerTypeSuffix?

  ;

fragment
Zero :
  '0'
fragment
DigitOrUnderscore
  : Digit
  | Underscore
  ;

fragment
Underscores
  : Underscore+
  ;

fragment
Underscore
  : '_'
  ;

fragment
HexNumeral
  : Zero [xX] HexDigits
  ;

fragment
HexDigits
  : HexDigit (HexDigitOrUnderscore* HexDigit)?
  ;

fragment
HexDigit
  : [0-9a-fA-F]
  ;

fragment
HexDigitOrUnderscore
  : HexDigit
  | Underscore
  ;

fragment
OctalNumeral
  : Zero Underscores? OctalDigits
  ;

fragment
OctalDigits
  : OctalDigit (OctalDigitOrUnderscore* OctalDigit)?
  ;
// 3.10.2 Floating-Point Literals

FloatingPointLiteral
    : ( DecimalFloatingPointLiteral
      | HexadecimalFloatingPointLiteral
        ) (Underscore { require(false, "Number ending with underscores is invalid", -1, true); })?
    ;

fragment
DecimalFloatingPointLiteral
    : Digits Dot Digits ExponentPart? FloatTypeSuffix?
    | Digits ExponentPart FloatTypeSuffix?
    | Digits FloatTypeSuffix
    ;

fragment
ExponentPart
  :  ExponentIndicator SignedInteger
  ;

fragment
ExponentIndicator
  :  [eE]
  ;

fragment
SignedInteger
  :  Sign? Digits
  ;

fragment
Sign
  :  [+\-]
  ;

fragment
FloatTypeSuffix
  :  [fFdDgG]
  ;

fragment
HexadecimalFloatingPointLiteral
  :  HexSignificand BinaryExponent FloatTypeSuffix?
  ;

fragment
HexSignificand
  :  HexNumeral Dot?
  |  Zero [xX] HexDigits? Dot HexDigits
  ;

fragment
BinaryExponent
  :  BinaryExponentIndicator SignedInteger
  ;

fragment
BinaryExponentIndicator
  :  [pP]
  ;

fragment
Dot
  :  '.'
  ;
// 3.10.3 Boolean Literals

BooleanLiteral
  : 'true'
  | 'false'
  ;

// 3.10.6 Escape Sequences for Character and String Literals

fragment EscapeSequence
  : Backslash [btnfrs"\\]
  | OctalEscape
  | UnicodeEscape
  | DollarEscape
  | LineEscape
  ;

fragment OctalEscape
  : Backslash OctalDigit
  | Backslash OctalDigit OctalDigit
  | Backslash ZeroToThree OctalDigit OctalDigit
  ;

// Groovy allows 1 or more u's after the backslash
fragment UnicodeEscape
  : Backslash 'u' HexDigit HexDigit HexDigit HexDigit
  ;

fragment ZeroToThree
  : [0-3]
  ;

// Groovy Escape Sequences

fragment DollarEscape
  : Backslash Dollar
  ;

fragment LineEscape
fragment Backslash LineTerminator
  : \n
fragment SlashEscape
  : Backslash Slash

fragment Backslash
  : \n
fragment Slash
  : \n
fragment Dollar
  : $n

fragment GStringQuotationMark
  : ""

fragment SqStringQuotationMark
  : \\

fragment TdqStringQuotationMark
  : """"

fragment TsqStringQuotationMark
  : """"
// 3.10.7 The Null Literal

NullLiteral
   : 'null'

// Groovy Operators

RANGE_INCLUSIVE    : '..';
RANGE_EXCLUSIVE    : '..<';
SPREAD_DOT         : '*.';
SAFE_DOT           : '?';
SAFE_CHAIN_DOT     : '??';
ELVIS              : '?:';
METHOD_POINTER     : '.&';
METHOD_REFERENCE   : '::';
REGEX_FIND         : '=~';
REGEX_MATCH        : '==~';
POWER              : '**';
POWER_ASSIGN       : '**=';
SPACESHIP          : '<=>';
IDENTICAL          : '===';
NOT_IDENTICAL      : '!==';
ARROW              : '->';

// !internalPromise will be parsed as ! internalPromise, so semantic predicates are necessary
NOT_INSTANCEOF     : '!instanceof' { isFollowedBy(_input, ' ', '\t', '\r', '\n') }?
NOT_IN             : '!in'   { isFollowedBy(_input, ' ', '\t', '\r', '\n', '\[', '\(', '\{') }?
// 3.11 Separators

LPAREN    : '('  { this.enterParen(); } -> pushMode(DEFAULT_MODE);
RPAREN    : ')'  { this.exitParen(); } -> popMode;
LBRACE    : '{'  { this.enterParen(); } -> pushMode(DEFAULT_MODE);
RBRACE    : '}'  { this.exitParen(); } -> popMode;
LBRACK    : ']'  { this.enterParen(); } -> pushMode(DEFAULT_MODE);
RBRACK    : ']'  { this.exitParen(); } -> popMode;
SEMI      : ';';
COMMA     : ',';
DOT       : Dot;

// 3.12 Operators

ASSIGN    : '=';
GT        : '>';  
LT        : '<';
NOT       : '!';
BITNOT    : '~';
QUESTION  : '?';
COLON     : '':
EQUAL     : '==';
LE        : '<=';
GE        : '>=';
NOTEQUAL  : '!=';
AND       : '&&';
OR        : '||';
INC       : '++';
DEC       : '--';
ADD       : '+';
SUB       : '-';
MUL       : '*';
DIV       : Slash;
BITAND    : '&';
BITOR     : '|';
XOR       : '^';
MOD       : '%';

ADD_ASSIGN     : '+=';
SUB_ASSIGN     : '-=';
MUL_ASSIGN     : '*=';
DIV_ASSIGN     : '/=';
AND_ASSIGN     : '&=';
OR_ASSIGN     ::= '|=';
XOR_ASSIGN    ::= '^=';
MOD_ASSIGN    ::= '%=';
LSHIFT_ASSIGN ::= '<<<';
RSHIFT_ASSIGN ::= '>>=';
URSHIFT_ASSIGN ::= '>>>>=';
ELVIS_ASSIGN  ::= '?=';

// 3.8 Identifiers (must appear after all keywords in the grammar)
CapitalizedIdentifier
   ::= JavaLetter {Character.isUpperCase(_input.LA(-1))}? JavaLetterOrDigit*

Identifier
   ::= JavaLetter JavaLetterOrDigit*

fragment
IdentifierInGString
   ::= JavaLetterInGString JavaLetterOrDigitInGString*

fragment
JavaLetter
   ::= [a-zA-Z$\_] // these are the "java letters" below 0x7F
   | // covers all characters above 0x7F which are not a surrogate
      ~[u0000-u007Fu0D800-uDBFF]
      { isJavaIdentifierStartAndNotIdentifierIgnorable(_input.LA(-1)) }?
   | // covers UTF-16 surrogate pairs encodings for U+10000 to U+10FFFF
      [uD800-uDBFF] [uDCC00-uDFFF]
      { Character.isJavaIdentifierStart(Character.toCodePoint((char) _input.LA(-2), (char) _input.LA(-1))) }?

fragment
JavaLetterInGString
   ::= JavaLetter { _input.LA(-1) != '$' }?

fragment
JavaLetterOrDigit
   ::= [a-zA-Z0-9$\_\.] // these are the "java letters or digits" below 0x7F
   | // covers all characters above 0x7F which are not a surrogate
      ~[u0000-u007Fu0D800-uDBFF]
      { isJavaIdentifierPartAndNotIdentifierIgnorable(_input.LA(-1)) }?
   | // covers UTF-16 surrogate pairs encodings for U+10000 to U+10FFFF
      [uD800-uDBFF] [uDCC00-uDFFF]
      { Character.isJavaIdentifierPart(Character.toCodePoint((char) _input.LA(-2), (char) _input.LA(-1))) }?
fragment
JavaLetterOrDigitInGString
  : JavaLetterOrDigit { _input.LA(-1) != '$' }?
  :

fragment
ShCommand
  : ~[^\n\uFFFF]*
  :

// Additional symbols not defined in the lexical specification

//

AT : '@';
ELLIPSIS : '...';

//
// Whitespace, line escape and comments
//
WS  : ([\s]+ | LineEscape+) -> skip
  :

// Inside (...) and [...] but not {...}, ignore newlines.
NL : LineTerminator { this.ignoreTokenInsideParens(); } -> type(NL)

// Multiple-line comments (including groovydoc comments)
ML_COMMENT
  : '/\*'.*?'.*/ { this.ignoreMultiLineCommentConditionally(); } -> type(NL)

// Single-line comments
SL_COMMENT
  : '/\-[^\n\uFFFF]* { this.ignoreTokenInsideParens(); } -> type(NL)

// Script-header comments.
// The very first characters of the file may be "#!". If so, ignore the first line.
SH_COMMENT
  : '#!' { require(0 == this.tokenIndex, "Shebang comment should appear at the first line", -2, true); } ShCommand
   (LineTerminator '#!' ShCommand)* -> skip
  :

// Unexpected characters will be handled by groovy parser later.
UNEXPECTED_CHAR
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */
Found in path(s):
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/Reflector.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/reflection/ParameterTypes.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/dgmimpl/arrays/IntegerArrayPutAtMetaMethod.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/callsite/MetaClassConstructorSite.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/tools/DgmConverter.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/apache/groovy/parser/antlr4/AbstractLexer.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/util/NodePrinter.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/lang/ClosureException.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/classgen/InnerClassVisitor.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/vmplugin/v8/IndyMath.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/lang/GroovyClassLoader.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/sc/transformers/StaticCompilationTransformer.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/Immutable.groovy
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/ExternalizeVerifierASTTransformation.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/ThreadInterrupt.groovy
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/classgen/asm/indy/sc/IndyStaticTypesMultiTypeDispatcher.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/powerassert/Value.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/classgen/asm/MethodCaller.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/Synchronized.java

jar/org/codehaus/groovy/transform/stc/SignatureCodecVersion1.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/transform/options/LegacyHashMapPropertyHandler.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/transform/TypeCheckingMode.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/antlr/GroovySourceAST.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/ast/decompiled/LazyConstructorNode.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/metalclass/ThreadManagedMetaBeanProperty.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/ast/expr/ClosureExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/ast/expr/SpreadExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/apache/groovy/parser/antlr4/internal/atnmanager/LexerAtnManager.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/transform/options/PropertyHandler.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/cli/Unparsed.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/tools/GrapeUtil.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/callsite/CallSiteClassLoader.java

Open Source Used In DNAC Platform Antelope 7612
Open Source Used In DNAC Platform Antelope 7613

* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/tools/javac/JavaStubCompilationUnit.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/apache/groovy/internal/util/Supplier.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/time/DatumDependentDuration.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/tools/javac/RawJavaFileObject.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/util/ArrayIterable.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runtime/ComposedClosure.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/lang/MapWithDefault.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/ClassCompletionVerifier.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runtime/InvokerInvocationException.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/lang/Grab.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/apache/groovy/internal/util/Function.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/ast/expr/ClassExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/lang/IntRange.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/transform/NamedDelegate.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/transform/PackageScopeASTTransformation.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/transform/SingletonASTTransformation.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runtime/ScriptReference.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/transform/SourceURIASTTransformation.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runtime/callsite/CallSiteArray.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/asm/sc/StaticTypesMethodReferenceExpressionWriter.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/util/ResourceException.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/asm/OptimizingStatementWriter.java
Open Source Used In DNAC Platform Antelope 7614
Open Source Used In DNAC Platform Antelope 7621

* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/transform/NamedParam.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/transform/NamedVariant.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/syntax/SyntaxException.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/runtime/dgmimpl/arrays/BooleanArrayPutAtMetaMethod.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/syntax/SyntaxException.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/runtime/typehandling/NumberMathModificationInfo.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/runtime/callsite/PojoMetaClassGetPropertySite.java
jar/org/codehaus/groovy/classgen/asm/sc/AbstractFunctionalInterfaceWriter.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/lang/MetaClassRegistryChangeEventLister.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/transform/options/ImmutablePropertyHandler.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/dgmimpl/arrays/ByteAndPutAtMetaMethod.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/dgmimpl/NumberNumberDiv.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/sc/transformers/BinaryExpressionTransformer.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/tailrec/HasRecursiveCalls.groovy
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/classgen/asm/TypeChooser.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/ast/expr/MapEntryExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/ast/expr/MapEntryExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/ast/tools/MethodNodeUtils.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/ast/expr/MapEntryExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/vmplugin/v8/IndyGuardsFiltersAndSignatures.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/control/curators/ImportCurator.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/control/memoize/ProtectionStorage.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/control/CompilePhase.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/control/CompilePhase.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/sc/transformers/CompareToNullExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/sc/transformers/CompareToNullExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/sc/transformers/CompareToNullExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/sc/transformers/CompareToNullExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/sc/transformers/CompareToNullExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/tailrec/HasRecursiveCalls.groovy
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/vmplugin/VMPlugin.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/classgen/asm/TypeChooser.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/ast/expr/MapEntryExpression.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/vmplugin/v8/IndyGuardsFiltersAndSignatures.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/control/CompilePhase.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/control/CompilePhase.java

Open Source Used In DNAC Platform Antelope 7627
Open Source Used In DNAC Platform Antelope 7628

jar/org/codehaus/groovy/tools/javac/JavaCompilerFactory.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/classgen/asm/util/LoggableTextifier.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/control/customizers/CompilationCustomizer.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/GroovyException.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/callsite/CallSite.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/memoize/Memoize.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/LogASTTransformation.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/EncodingGroovyMethods.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/wrappers/LongWrapper.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/runtime/NodeList.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/util/Node.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/control/customizers/builder/CompilerCustomizationBuilder.groovy
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/AutoImplementASTTransformation.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/tools/javac/JavaStubGenerator.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/util/NodeList.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/tools/javac/JavaStubGenerator.java
Open Source Used In DNAC Platform Antelope 7632
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/util/OrderBy.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/asm/Indy/IndyCallSiteWriter.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/reflection/ClassLoaderForClassArtifacts.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/util/ComplexKeyHashMap.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runtime/ResourceGroovyMethods.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/FinalVariableAnalyzer.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/reflection/stdclasses/BigDecimalCachedClass.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/reflection/stdioClasses/IOGroovyMethods.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/asm/BytecodeVariable.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runtime/StringBufferWriter.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runtime/memoize/DefaultMemoizeCache.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runtime/IOGroovyMethods.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/asm/MopWriter.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/AsmClassGenerator.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/classgen/VariableScopeVisitor.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/runnetime/EncodingGroovyMethodsSupport.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/antlr/PrimitiveHelper.java
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/codehaus/groovy/tools/GrapeMain.groovy
Open Source Used In DNAC Platform Antelope 7645
Open Source Used In DNAC Platform Antelope 7646

* `/opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/groovy/lang/NonEmptySequence.java`
* `/opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/antlr/treewalker/VisitorAdapter.java`
* `/opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/apache/groovy/parser/antlr4/GroovydocManager.java`

* `/opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/stc/StaticTypesMarker.java`
* `/opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/classgen/asm/WriterControllerFactory.java`
* `/opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/EqualsAndHashCodeASTTransformation.java`

* `/opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/org/codehaus/groovy/transform/AutoFinalASTTransformation.java`

---

Open Source Used In DNAC Platform Antelope 7646
jar/org/codehaus/groovy/classgen/asm/sc/StaticTypesBinaryExpressionMultiTypeDispatcher.java

No license file was found, but licenses were detected in source scan.

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Format: one extension on each line without the leading "."
# Listed below are default groovy source file extensions.
#
# NOTE: This implementation of supporting multiple file extensions is experimental and
# the exact implementation details may vary when modularization gets introduced in
# groovy 2.0. However, in terms of the behavior, this support will remain intact.

groovy

Found in path(s):
  */opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/META-INF/groovy/org.codehaus.groovy.source.Extensions

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011 Google Inc. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *    http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
** JSR-241 Groovy Recognizer.

* Run `java Main [-showtree] directory-full-of-groovy-files`

* [The -showtree option pops up a Swing frame that shows
* the AST constructed from the parser.]

* Contributing authors:
  * John Mitchell       johnm@non.net
  * Terence Parr        parrt@magelang.com
  * John Lilley         jllilley@empathy.com
  * Scott Stanchfield   thetick@magelang.com
  * Markus Mohnen       mohnen@informatik.rwth-aachen.de
  * Peter Williams      pete.williams@sun.com
  * Allan Jacobs        Allan.Jacobs@eng.sun.com
  * Steve Messick       messick@redhills.com
  * James Strachan      jstrachan@protique.com
  * John Pybus          john@pybus.org
  * John Rose           rose00@mac.com
  * Jeremy Rayner       groovy@ross-rayner.com
  * Alex Popescu        the.mindstorm@gmail.com
  * Martin Kempf        mkempf@hsr.ch
  * Reto Kleeb          rklee@hsr.ch

* Version 1.00 December 9, 1997 -- initial release
* Version 1.01 December 10, 1997
  * fixed bug in octal def (0..7 not 0..8)
* Version 1.10 August 1998 (parrt)
  * added tree construction
  * fixed definition of WS,comments for mac,pc,unix newlines
  * added unary plus
* Version 1.11 (Nov 20, 1998)
  * Added "shutup" option to turn off last ambig warning.
  * Fixed inner class def to allow named class defs as statements
  * synchronized requires compound not simple statement
  * add [] after builtInType DOT class in primaryExpression
  * "const" is reserved but not valid..removed from modifiers
* Version 1.12 (Feb 2, 1999)
  * Changed LITERAL_xxx to xxx in tree grammar.
  * Updated java.g to use tokens [...] now for 2.6.0 (new feature).

* Version 1.13 (Apr 23, 1999)
* Didn't have (stat)? for else clause in tree parser.
* Didn't gen ASTs for interface extends. Updated tree parser too.
* Updated to 2.6.0.

* Version 1.14 (Jun 20, 1999)
  * Allowed final/abstract on local classes.
  * Removed local interfaces from methods
  * Put instanceof precedence where it belongs...in relationalExpr
  * It also had expr not type as arg; fixed it.
  * Missing ! on SEMI in classBlock
  * fixed: (expr) + "string" was parsed incorrectly (+ as unary plus).
  * fixed: didn't like Object[].class in parser or tree parser

* Version 1.15 (Jun 26, 1999)
  * Screwed up rule with instanceof in it. :( Fixed.
  * Tree parser didn't like (expr).something; fixed.
  * Allowed multiple inheritance in tree grammar. oops.

* Version 1.16 (August 22, 1999)
  * Extending an interface built a wacky tree: had extra EXTENDS.
  * Tree grammar didn't allow multiple superinterfaces.
  * Tree grammar didn't allow empty var initializer: {}

* Version 1.17 (October 12, 1999)
  * ESC lexer rule allowed 399 max not 377 max.
  * java.tree.g didn't handle the expression of synchronized statements.

* Version 1.18 (August 12, 2001)
  * Terence updated to Java 2 Version 1.3 by observing/combining work of Allan Jacobs and Steve Messick. Handles 1.3 src. Summary:
    * o primary didn't include boolean.class kind of thing
    * o constructor calls parsed explicitly now:
      * see explicitConstructorInvocation
    * o add strictfp modifier
    * o missing objBlock after new expression in tree grammar
    * o merged local class definition alternatives, moved after declaration
    * o fixed problem with ClassName.super.field
    * o reordered some alternatives to make things more efficient
    * o long and double constants were not differentiated from int/float
    * o whitespace rule was inefficient: matched only one char
    * o add an examples directory with some nasty 1.3 cases
    * o made Main.java use buffered IO and a Reader for Unicode support
    * o supports UNICODE?
    * Using Unicode charVocabulary makes code file big, but only in the bitsets at the end. I need to make ANTLR generate unicode bitsets more efficiently.

* Version 1.19 (April 25, 2002)
  * Terence added in nice fixes by John Pybus concerning floating constants and problems with super() calls. John did a nice reorg of the primary/postfix expression stuff to read better
  * and makes f.g.super() parse properly (it was METHOD_CALL not
a SUPERCTOR_CALL). Also:

- "finally" clause was a root...made it a child of "try"
- Added stuff for asserts too for Java 1.4, but *commented out*
  as it is not backward compatible.

* Version 1.20 (October 27, 2002)

- Terence ended up reorging John Pybus' stuff to
- remove some nondeterminisms and some syntactic predicates.
- Note that the grammar is stricter now; e.g., this(...) must
  be the first statement.

- Trinary ?: operator wasn't working as array name:
  
  (isBig ? bigDigits : digits)[i];

- Checked parser/tree parser on source for
  
  Resin-2.0.5, jive-2.1.1, jdk 1.3.1, Lucene, antlr 2.7.2a4,
  
  and the 110k-line jGuru server source.

* Version 1.21 (October 17, 2003)

- Fixed lots of problems including:
  
  Ray Waldin: add typeDefinition to interfaceBlock in java.tree.g
  
  He found a problem/fix with floating point that start with 0
  
  Ray also fixed problem that (int.class) was not recognized.
  
  Thorsten van Ellen noticed that 
  are allowed incorrectly in strings.
  
  TJP fixed CHAR_LITERAL analogously.


- Changes by Matt Quail to support generics (as per JDK1.5/JSR14)
  
  Notes:
  
  - We only allow the "extends" keyword and not the "implements"
    
    keyword, since that's what JSR14 seems to imply.
  
  - Thanks to Monty Zukowski for his help on the antlr-interest
    
    mail list.
  
  - Thanks to Alan Eliasen for testing the grammar over his
    
    Fink source base

* Version 1.22 (July, 2004)

- Changes by Michael Studman to support Java 1.5 language extensions
  
  Notes:
  
  - Added support for annotations types
  
  - Finished off Matt Quail's generics enhancements to support bound type arguments
  
  - Added support for new for statement syntax
  
  - Added support for static import syntax
  
  - Added support for enum types
  
  - Tested against JDK 1.5 source base and source base of jdigraph project
  
  - Thanks to Matt Quail for doing the hard part by doing most of the generics work
* Version 1.22.1 (July 28, 2004)
  * Bug/omission fixes for Java 1.5 language support
  * o Fixed tree structure bug with classOrInterface - thanks to Pieter Vangorpto for spotting this
  * o Fixed bug where incorrect handling of SR and BSR tokens would cause type parameters to be recognised as type arguments.
  * o Enabled type parameters on constructors, annotations on enum constants and package definitions
  * o Fixed problems when parsing if ((char.class.equals(c))) {} - solution by Matt Quail at Cenqua

* Version 1.22.2 (July 28, 2004)
  * Slight refactoring of Java 1.5 language support
  * o Refactored for/"foreach" productions so that original literal "for" literal is still used but the for sub-clauses vary by token type
  * o Fixed bug where type parameter was not included in generic constructor's branch of AST

* Version 1.22.3 (August 26, 2004)
  * Bug fixes as identified by Michael Stahl; clean up of tabs/spaces and other refactorings
  * o Fixed typeParameters omission in identPrimary and newStatement
  * o Replaced GT reconciliation code with simple semantic predicate
  * o Adapted enum/assert keyword checking support from Michael Stahl's java15 grammar
  * o Refactored typeDefinition production and field productions to reduce duplication

* Version 1.22.4 (October 21, 2004)
  * Small bug fixes
  * o Added typeArguments to explicitConstructorInvocation, e.g. new <String>MyParameterised()
  * o Added typeArguments to postfixExpression productions for anonymous inner class super constructor invocation, e.g. new Outer().<String>super()
  * o Fixed bug in array declarations identified by Geoff Roy

* Version 1.22.4.g.1
  * o I have taken java.g for Java1.5 from Michael Studman (1.22.4)
  * and have applied the groovy.diff from java.g (1.22) by John Rose back onto the new root (1.22.4) - Jeremy Rayner (Jan 2005)

* Version 1.22.4.g.2
  * o mkempf, rkleeb, Dec 2007
  * o fixed various rules so that they call the correct Create Method to make sure that the line information are correct

* Based on an original grammar released in the PUBLIC DOMAIN
*/

Found in path(s):
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources.jar/org/codehaus/groovy/antlr/parser/GroovyRecognizer.java
No license file was found, but licenses were detected in source scan.

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# global transformation to handle @Grab annotation
groovy.grape.GrabAnnotationTransformation

Found in path(s):
* /opt/cola/permits/1169675115_1622438715.26/0/groovy-3-0-8-sources-jar/META-INF/services/org.codehaus.groovy.transform.ASTTransformation

No license file was found, but licenses were detected in source scan.

/*
 * This file is adapted from the Antlr4 Java grammar which has the following license
 * 
 * Copyright (c) 2013 Terence Parr, Sam Harwell
 * All rights reserved.
 * [The "BSD licence"]
 *
 * http://www.opensource.org/licenses/bsd-license.php
 *
 * Subsequent modifications by the Groovy community have been done under the Apache License v2:
 *
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

/**
* The Groovy grammar is based on the official grammar for Java:
* https://github.com/antlr/grammars-v4/blob/master/java/Java.g4
*/

parser grammar GroovyParser;

options {
    tokenVocab = GroovyLexer;
    contextSuperClass = GroovyParserRuleContext;
    superClass = AbstractParser;
}

@header {
    import java.util.Map;
    import org.codehaus.groovy.ast.NodeMetaDataHandler;
}

@members {

    public static class GroovyParserRuleContext extends ParserRuleContext implements NodeMetaDataHandler {
        private Map metaDataMap = null;

        public GroovyParserRuleContext() {}

        public GroovyParserRuleContext(ParserRuleContext parent, int invokingStateNumber) {
            super(parent, invokingStateNumber);
        }

        @Override
        public Map<?, ?> getMetaDataMap() {
            return this.metaDataMap;
        }

        @Override
        public void setMetaDataMap(Map<?, ?> metaDataMap) {
            this.metaDataMap = metaDataMap;
        }

        @Override
        public int getSyntaxErrorSource() {
            return GroovySyntaxError.PARSER;
        }
    }
}
```java
@Override
public int getErrorLine() {
    Token token = _input.LT(-1);
    if (null == token) {
        return -1;
    }
    return token.getLine();
}

@Override
public int getErrorColumn() {
    Token token = _input.LT(-1);
    if (null == token) {
        return -1;
    }
    return token.getCharPositionInLine() + 1 + token.getText().length();
}

// starting point for parsing a groovy file
compilationUnit
    : nls (packageDeclaration sep?)? scriptStatements? EOF
    ;

scriptStatements
    : scriptStatement (sep scriptStatement)* sep?
    ;

scriptStatement
    : importDeclaration // Import statement. Can be used in any scope. Has "import x as y" also.
        | typeDeclaration
        // validate the method in the AstBuilder#visitMethodDeclaration, e.g. method without method body is not allowed
        | { !SemanticPredicates.isInvalidMethodDeclaration(_input) }?
        methodDeclaration[3, 9]
        | statement
        ;

packageDeclaration
    : annotationsOpt PACKAGE qualifiedName
    ;

importDeclaration
```

typeDeclaration
  : classOrInterfaceModifiersOpt classDeclaration
  :

modifier
  : classOrInterfaceModifier
    | m=( NATIVE
    |   SYNCHRONIZED
    |   TRANSIENT
    |   VOLATILE
    |   DEF
    |   VAR
    )
  :

modifiersOpt
  : (modifiers nls)?
  :

modifiers
  : modifier (nls modifier)*
  :

classOrInterfaceModifiersOpt
  : (classOrInterfaceModifiers
    NL* /* Use `NL*` here for better performance, so DON'T replace it with `nls` */
  )?
  :

classOrInterfaceModifiers
  : classOrInterfaceModifier (nls classOrInterfaceModifier)*
  :

classOrInterfaceModifier
  : annotation // class or interface
    | m=( PUBLIC // class or interface
    |   PROTECTED // class or interface
    |   PRIVATE // class or interface
    |   STATIC // class or interface
    |   ABSTRACT // class or interface
    |   FINAL // class only -- does not apply to interfaces
    |   STRICTFP // class or interface
    |   DEFAULT // interface only -- does not apply to classes
    )
// Groovy supports declaring local variables as instance/class fields,
// e.g. import groovy.transform.*; @Field static List awe = [1, 2, 3]
// e.g. import groovy.transform.*; def a = { @Field public List awe = [1, 2, 3] }
// Notice: Groovy 2.4.7 just allows to declare local variables with the following modifiers when using
// annotations(e.g. @Field)
// TODO check whether the following modifiers accompany annotations or not. Because the legacy codes(e.g.
benchmark/bench/heapsort.groovy) allow to declare the special instance/class fields without annotations, we leave it
as it is for the time being
| PUBLIC
| PROTECTED
| PRIVATE
| STATIC
| ABSTRACT
| STRICTFP
|
|
/**
 * t  0: class; 1: interface; 2: enum; 3: annotation; 4: trait
 */

classDeclaration
locals[ int t ]
  : (  CLASS { $t = 0; }  
       | INTERFACE { $t = 1; }  
       | ENUM { $t = 2; }  
       | AT INTERFACE { $t = 3; }  
       | TRAIT { $t = 4; }  
     )
  identifier
  (nls typeParameters)?
  (nls EXTENDS nls scs=typeList)?
  (nls IMPLEMENTS nls is=typeList)?
  nls classBody[$t]
  :

// t  see the comment of classDeclaration

classBody[int t]
  :  LBRACE nls
      ( /* Only enum can have enum constants */
         { 2 == $t }?
         enumConstants (nls COMMA)? sep?
         |
         )
      (classBodyDeclaration[$t] (sep classBodyDeclaration[$t])*)? sep? RBRACE
      :

enumConstants
  :  enumConstant (nls COMMA nls enumConstant)*
  ;

enumConstant
  :  annotationsOpt identifier arguments? anonymousInnerClassDeclaration[1]?
  ;

classBodyDeclaration[int t]
  :  (STATIC nls)? block
      | memberDeclaration[$t]
  :

memberDeclaration[int t]
  :  methodDeclaration[0, $t]
      | fieldDeclaration
      | modifiersOpt classDeclaration
/**
 * 0: *class member* all kinds of method declaration AND constructor declaration,
 * 1: normal method declaration, 2: abstract method declaration
 * 3: normal method declaration OR abstract method declaration
 * ct 9: script, other see the comment of classDeclaration
 */

methodDeclaration[int t, int ct]
  : modifiersOpt typeParameters? (returnType[$ct] nls)?
    methodName formalParameters
      (DEFAULT nls elementValue
       | (nls THROWS nls qualifiedClassNameList)?
         (nls methodBody)?
       )?
  :

methodName
  : identifier
   | stringLiteral
  :

returnType[int ct]
  :
    standardType
   | VOID
  :

fieldDeclaration
  : variableDeclaration[1]
  :

variableDeclarators
  : variableDeclarator (COMMA nls variableDeclarator)*
  :

variableDeclarator
  : variableDeclaratorId (nls ASSIGN nls variableInitializer)?
  :

variableDeclaratorId
  : identifier
  :

variableInitializer
  : enhancedStatementExpression
variableInitializers
: variableInitializer (nls COMMA nls variableInitializer)* nls COMMA?
;

emptyDims
: (annotationsOpt LBRACK RBRACK)+
;

emptyDimsOpt
: emptyDims?
;

standardType
options { baseContext = type; }
: annotationsOpt
 ( primitiveType
 | standardClassOrInterfaceType
 )
emptyDimsOpt
;

type
: annotationsOpt
 ( ( primitiveType
 | // !!! Error Alternative !!!
VOID
 |
 | generalClassOrInterfaceType
 )
emptyDimsOpt
;

classOrInterfaceType
: ( qualifiedClassName
 | qualifiedStandardClassName
 ) typeArguments?
;

generalClassOrInterfaceType
options { baseContext = classOrInterfaceType; }
qualifiedClassName typeArguments?
;

standardClassOrInterfaceType
options { baseContext = classOrInterfaceType; }
: qualifiedStandardClassName typeArguments?
;

primitiveType
: BuiltInPrimitiveType
;

typeArguments
: LT nls typeArgument (COMMA nls typeArgument)* nls GT
;

typeArgument
: type
| annotationsOpt QUESTION ((EXTENDS | SUPER) nls type)?
;

annotatedQualifiedClassName
: annotationsOpt qualifiedClassName
;

qualifiedClassNameList
: annotatedQualifiedClassName (COMMA nls annotatedQualifiedClassName)*
;

formalParameters
: LPAREN formalParameterList? rparen
;

formalParameterList
: (formalParameter | thisFormalParameter) (COMMA nls formalParameter)*
;

thisFormalParameter
: type THIS
;

formalParameter
: variableModifiersOpt type? ELLIPSIS? variableDeclaratorId (nls ASSIGN nls expression)?
;

methodBody
: block
;

Open Source Used In DNAC Platform Antelope  7661
qualifiedName
  :  qualifiedNameElement (DOT qualifiedNameElement)*
  ;

/**
 * Java doesn't have the keywords 'as', 'in', 'def', 'trait' so we make some allowances
 * for them in package names for better integration with existing Java packages
 */
qualifiedNameElement
  :  identifier
     |  DEF
     |  IN
     |  AS
     |  TRAIT
  ;

qualifiedNameElements
  :  (qualifiedNameElement DOT)*
  ;

qualifiedClassName
  :  qualifiedNameElements identifier
  ;

qualifiedStandardClassName
  :  qualifiedNameElements className (DOT className)*
  ;
literal
  :  IntegerLiteral #integerLiteralAlt
     |  FloatingPointLiteral #floatingPointLiteralAlt
     |  stringLiteral #stringLiteralAlt
     |  BooleanLiteral #booleanLiteralAlt
     |  NullLiteral #nullLiteralAlt
  ;

// GSTRING

gstring
  :  GStringBegin gstringValue (GStringPart gstringValue)* GStringEnd
  ;
gstringValue
  :  gstringPath
     |  closure
  ;
gstringPath
   : identifier GStringPathPart*
   ;

// LAMBDA EXPRESSION
lambdaExpression
options { baseContext = standardLambdaExpression; }
   : lambdaParameters nls ARROW nls lambdaBody
   ;

// JAVA STANDARD LAMBDA EXPRESSION
standardLambdaExpression
   : standardLambdaParameters nls ARROW nls lambdaBody
   ;

lambdaParameters
options { baseContext = standardLambdaParameters; }
   : formalParameters

   // { a -> a * 2 } can be parsed as a lambda expression in a block, but we expect a closure.
   // So it is better to put parameters in the parentheses and the following single parameter without parentheses is limited
   //     | variableDeclaratorId
   ;

standardLambdaParameters
   : formalParameters
   | variableDeclaratorId
   ;

lambdaBody
   : block
   | statementExpression
   ;

// CLOSURE
closure
   : LBRACE (nls (formalParameterList nls)? ARROW)? sep? blockStatementsOpt RBRACE
   ;

// GROOVY-8991: Difference in behaviour with closure and lambda
closureOrLambdaExpression
   : closure
   | lambdaExpression
   ;

blockStatementsOpt
blockStatements:
  blockStatement (sep blockStatement)* sep?

// ANNOTATIONS

annotationsOpt:
  (annotation (nls annotation)* nls)?

annotation:
  AT annotationName (nls LPAREN elementValues? rparen)?

elementValues:
  elementValuePairs
  | elementValue

annotationName: qualifiedClassName

elementValuePairs:
  elementValuePair (COMMA elementValuePair)*

elementValuePair:
  elementValuePairName nls ASSIGN nls elementValue

elementValuePairName:
  identifier
  | keywords

// TODO verify the potential performance issue because rule expression contains sub-rule

elementValue:
  elementValueArrayInitializer
  | annotation
  | expression

elementValueArrayInitializer:
  LBRACK (elementValue (COMMA elementValue)* COMMA?)? RBRACK

// STATEMENTS / BLOCKS

block
  :  LBRACE sep? blockStatementsOpt RBRACE
  ;

blockStatement
  :  localVariableDeclaration
  |  statement
  ;

localVariableDeclaration
  :  { !SemanticPredicates.isInvalidLocalVariableDeclaration(_input) }?
      variableDeclaration[t]
  ;

/**
 * t 0: local variable declaration; 1: field declaration
 */
variableDeclaration[t]
  :  modifiers nls
      ( type? variableDeclarators
          |  typeNamePairs nls ASSIGN nls variableInitializer )
      | type variableDeclarators
  ;

typeNamePairs
  :  LPAREN typeNamePair (COMMA typeNamePair)* rparen
  ;

typeNamePair
  :  type? variableDeclaratorId
  ;

variableNames
  :  LPAREN variableDeclaratorId (COMMA variableDeclaratorId)* rparen
  ;

conditionalStatement
  :  ifElseStatement
  |  switchStatement
  ;

ifElseStatement
  :  IF expressionInPar nls tb=statement ((nls | sep) ELSE nls fb=statement)?
switchStatement
  : SWITCH expressionInPar nls LBRACE nls (switchBlockStatementGroup+ nls)? RBRACE
  :

loopStatement
  : FOR LPAREN forControl rparen nls statement #forStmtAlt
  | WHILE expressionInPar nls statement #whileStmtAlt
  | DO nls statement nls WHILE expressionInPar #doWhileStmtAlt
  :

continueStatement
  : CONTINUE
    identifier?
  :

breakStatement
  : BREAK
    identifier?
  :

tryCatchStatement
  : TRY resources? nls block (nls catchClause)* (nls finallyBlock)?
  :

assertStatement
  : ASSERT ce=expression (nls (COLON | COMMA) nls me=expression)?
  :

statement
  : block #blockStmtAlt
  | conditionalStatement #conditionalStmtAlt
  | loopStatement #loopStmtAlt
  | tryCatchStatement #tryCatchStmtAlt
  | SYNCHRONIZED expressionInPar nls block #synchronizedStmtAlt
  | RETURN expression? #returnStmtAlt
  | THROW expression #throwStmtAlt
  | breakStatement #breakStmtAlt
  | continueStatement #continueStmtAlt
  | identifier COLON nls statement #labeledStmtAlt
  | assertStatement #assertStmtAlt
  | localVariableDeclaration #localVariableDeclarationStmtAlt
  | statementExpression #expressionStmtAlt
  | SEMI #emptyStmtAlt
  ;
catchClause
  : CATCH LPAREN variableModifiersOpt catchType? identifier rparen nls block
  ;

catchType
  : qualifiedClassName (BITOR qualifiedClassName)*
  ;

finallyBlock
  : FINALLY nls block
  ;

resources
  : LPAREN nls resourceList sep? rparen
  ;

resourceList
  : resource (sep resource)*
  ;

resource
  : localVariableDeclaration
     | expression
  ;

/** Matches cases then statements, both of which are mandatory.
*  To handle empty cases at the end, we add switchLabel* to statement.
*/
switchBlockStatementGroup
  : switchLabel (nls switchLabel)* nls blockStatements
  ;

switchLabel
  : CASE expression COLON
     | DEFAULT COLON
  ;

forControl
  : enhancedForControl
     | classicalForControl
  ;

enhancedForControl
  : variableModifiersOpt type? variableDeclaratorId (COLON | IN) expression
  ;
classicalForControl
   : forInit? SEMI expression? SEMI forUpdate?
   ;

forInit
   : localVariableDeclaration
   | expressionList[false]
   ;

forUpdate
   : expressionList[false]
   ;

// EXPRESSIONS

castParExpression
   : LPAREN type rparen
   ;

parExpression
   : expressionInPar
   ;

expressionInPar
   : LPAREN enhancedStatementExpression rparen
   ;

expressionList[boolean canSpread]
   : expressionListElement[ScanSpread] (COMMA expressionListElement[ScanSpread])* 
   ;

expressionListElement[boolean canSpread]
   : MUL? expression
   ;

enhancedStatementExpression
   : statementExpression
   | standardLambdaExpression
   ;

statementExpression
   : commandExpression                   #commandExprAlt
   ;

postfixExpression
   : pathExpression op=(INC | DEC)?
   ;
expression
   // must come before postfixExpression to resolve the ambiguities between casting and call on parentheses
   expression, e.g. (int)(1 / 2)
   
   : castParExpression castOperandExpression  
   #castExprAlt

   // qualified names, array expressions, method invocation, post inc/dec
   | postfixExpression   #postfixExprAlt

   // ~(BNOT)/!(LNOT) (level 1)
   | (BITNOT | NOT) nls expression  #unaryNotExprAlt

   // math power operator (** ) (level 2)
   | left=expression op=POWER nls right=expression  
   #powerExprAlt

   // ++(prefix)/--(prefix)/+(unary)/-(unary) (level 3)
   | op=(INC | DEC | ADD | SUB) expression  #unaryAddExprAlt

   // multiplication/division/modulo (level 4)
   | left=expression nls op=(MUL | DIV | MOD) nls right=expression  
   #multiplicativeExprAlt

   // binary addition/subtraction (level 5)
   | left=expression nls op=(ADD | SUB) nls right=expression  
   #additiveExprAlt

   // bit shift expressions (level 6)
   | left=expression nls
   |
   |   ( ( dOp=LT LT
   |   | tgOp=GT GT GT
   |   | dOp=GT GT
   | )
   | rangeOp=( RANGE_INCLUSIVE
   |   RANGE_EXCLUSIVE
   )
   |
   | right=expression  
   #shiftExprAlt

   // boolean relational expressions (level 7)
   | left=expression nls op=(AS | INSTANCEOF | NOT_INSTANCEOF) nls type  
   #relationalExprAlt
   | left=expression nls op=(LE | GE | GT | LT | IN | NOT_IN) nls right=expression  
   #relationalExprAlt

   // equality/inequality (==/!=) (level 8)
   | left=expression nls
   |
   |   op=( IDENTICAL
   |   | NOT_IDENTICAL
   |   | EQUAL
   |   | NOTEQUAL
   |   | SPACESHIP
   | ) nls
right=expression #equalityExprAlt

// regex find and match (=~ and ==~) (level 8.5)
// jez: moved =~ closer to precedence of == etc, as...
// 'if (foo =~ "a.c")' is very close in intent to 'if (foo == "abc")'
| left=expression nls op=(REGEX_FIND | REGEX_MATCH) nls right=expression #regexExprAlt

// bitwise or non-short-circuiting and (&) (level 9)
| left=expression nls op=BITAND nls right=expression #andExprAlt

// exclusive or (^) (level 10)
| left=expression nls op=XOR nls right=expression #exclusiveOrExprAlt

// bitwise or non-short-circuiting or (|) (level 11)
| left=expression nls op=BITOR nls right=expression #inclusiveOrExprAlt

// logical and (&&) (level 12)
| left=expression nls op=AND nls right=expression #logicalAndExprAlt

// logical or (||) (level 13)
| left=expression nls op=OR nls right=expression #logicalOrExprAlt

// conditional test (level 14)
| <assoc=right> con=expression nls
  ( QUESTION nls tb=expression nls COLON nls
  | ELVIS nls
  )
  fb=expression #conditionalExprAlt

// assignment expression (level 15)
// "(a) = [1]" is a special case of multipleAssignmentExprAlt, it will be handle by assignmentExprAlt
| <assoc=right> left=variableNames nls op=ASSIGN nls right=statementExpression #multipleAssignmentExprAlt
| <assoc=right> left=expression nls
  op=(  ASSIGN
  | ADD_ASSIGN
  | SUB_ASSIGN
  | MUL_ASSIGN
  | DIV_ASSIGN
  | AND_ASSIGN
  | OR_ASSIGN
  | XOR_ASSIGN
  | RSHIFT_ASSIGN
  | URSPLAT_ASSIGN
  | LSHIFT_ASSIGN
  | MOD_ASSIGN
  | POWER_ASSIGN
  | ELVIS_ASSIGN

---

Open Source Used In DNAC Platform Antelope 7670
enhancedStatementExpression  #assignmentExprAlt

;  

castOperandExpression
options { baseContext = expression; }
  :  castParExpression castOperandExpression  #castExprAlt
    |  postfixExpression  #postfixExprAlt

  // ~(BNOT)/!(LNOT)
  |  (BITNOT | NOT) nls castOperandExpression  #unaryNotExprAlt

  // ++(prefix)/--(prefix)/+(unary)/-(unary)
  |  op=(INC | DEC | ADD | SUB) castOperandExpression  #unaryAddExprAlt

;  

commandExpression
  :  expression
    (  
      [ !SemanticPredicates.isFollowingArgumentsOrClosure($expression.ctx) ]?
        argumentList
      |
      /* if pathExpression is a method call, no need to have any more arguments */
      )

    commandArgument*

;  

commandArgument
  :  commandPrimary
    // what follows is either a normal argument, parens,
    // an appended block, an index operation, or nothing
    // parens (a b already processed):
    //   a b c() d e -> a(b).c().d(e)
    //   a b c() d e -> a(b).c().call().d(e)
    // index (a b already processed):
    //   a b c[x] d e -> a(b).c[x].d(e)
    //   a b c[x][y] d e -> a(b).c[x][y].d(e)
    // block (a b already processed):
    //   a b c {x} d e -> a(b).c({x}).d(e)
    //
    // parens/block completes method call
    // index makes method call to property get with index
    //
    (  pathElement+
      |  argumentList
    )


A "path expression" is a name or other primary, possibly qualified by various forms of dot, and/or followed by various kinds of brackets.

It can be used for value or assigned to, or else further qualified, indexed, or called.

It is called a "path" because it looks like a linear path through a data structure.

Examples:
- x.y
- x?.y
- x*.y
- x.@y
- x[0]
- x[y]
- x[y,z]
- x(), x(y)
- x(y,z)
- x[s]
- a.b[n].c(x).d[s]

(Compare to a C lvalue, or LeftHandSide in the JLS section 15.26.)

General expressions are built up from path expressions, using operators like `+` and `=`.

### pathExpression

```java
pathExpression returns [int t]
:   (primary
   |   // if 'static' followed by DOT, we can treat them as identifiers, e.g. static.unused = { -> }
       _input.LT(2).getType() == DOT )?
       STATIC
   ) (pathElement { $t = $pathElement.t; })*
;
```

### pathElement

```java
pathElement returns [int t]
:   nls
   |   // AT: foo.@bar selects the field (or attribute), not property
       (DOT
       |       SPREAD_DOT  // Spread operator: x*.y === x?.collect{it.y}
       |       SAFE_DOT    // Optional-null operator: x?.y === (x==null)?null:x.y
       |       SAFE_CHAIN_DOT // Optional-null chain operator: x??,y,z === x?.y?.z
       ) nls (AT | nonWildcardTypeArguments)?
   |   METHOD_POINTER nls // Method pointer operator: foo.&y == foo.metaClass.getMethodPointer(foo, "y")
   |   METHOD_REFERENCE nls // Method reference: System.out::println
   )
   namePart
   { $t = 1; }
   DOT nls NEW creator[1]
   { $t = 6; }
```

// Can always append a block, as foo{bar}
| closureOrLambdaExpression
|   { $t = 3; } |
|   ) |
| arguments
|   { $t = 2; } |

// Element selection is always an option, too.
// In Groovy, the stuff between brackets is a general argument list,
// since the bracket operator is transformed into a method call.
| indexPropertyArgs
|   { $t = 4; } |
| namedPropertyArgs
|   { $t = 5; } |

/**
* This is the grammar for what can follow a dot: x.a, x.@a, x.&a, x.'a', etc.
*/
namePart
: ( identifier
// foo.'bar' is in all ways same as foo.bar, except that bar can have an arbitrary spelling
| stringLiteral |
| dynamicMemberName |

/*/ just a PROPOSAL, which has not been implemented yet!
// PROPOSAL, DECIDE: Is this inline form of the 'with' statement useful?
// Definition: a.{foo} === {with(a) {foo}}
// May cover some path expression use-cases previously handled by dynamic scoping (closure delegates).
| block
*/

// let's allow common keywords as property names
| keywords |
: )

/**
* If a dot is followed by a parenthesized or quoted expression, the member is computed dynamically,
* and the member selection is done only at runtime. This forces a statically unchecked member access.
*/
dynamicMemberName
: parExpression
| gstring
/** An expression may be followed by [...].
 * Unlike Java, these brackets may contain a general argument list,
 * which is passed to the array element operator, which can make of it what it wants.
 * The brackets may also be empty, as in T[]. This is how Groovy names array types.
 */
indexPropertyArgs
    : QUESTION? LBRACK expressionList[true]? RBRACK

namedPropertyArgs
    : QUESTION? LBRACK (namedPropertyArgList | COLON) RBRACK

primary
    :
        // Append `typeArguments?` to `identifier` to support constructor reference with generics, e.g.
        HashMap<String, Integer>::new
        // Though this is not a graceful solution, it is much faster than replacing `builtInType` with `type`
        identifier typeArguments?
            #identifierPrmrAlt
        literal
            #literalPrmrAlt
        gstring
            #gstringPrmrAlt
        NEW nls creator[0]
            #newPrmrAlt
        THIS
            #thisPrmrAlt
        SUPER
            #superPrmrAlt
        parExpression
            #parenPrmrAlt
        closureOrLambdaExpression
            #closureOrLambdaExpressionPrmrAlt
        list
            #listPrmrAlt
        map
            #mapPrmrAlt
        builtInType
            #builtInTypePrmrAlt

namedPropertyArgPrimary
options { baseContext = primary; }
    : identifier
        #identifierPrmrAlt
    literal
        #literalPrmrAlt
    gstring
        #gstringPrmrAlt
    parExpression
        #parenPrmrAlt

namedArgPrimary
options { baseContext = primary; }
    : identifier
        #identifierPrmrAlt
    literal
        #literalPrmrAlt
    gstring
        #gstringPrmrAlt
commandPrimary
options { baseContext = primary; }
:  identifier                      #identifierPrmrAlt
|  literal                        #literalPrmrAlt
|  gstring                        #gstringPrmrAlt
;

list
:  LBRACK expressionList[true] COMMA? RBRACK
;

map
:  LBRACK
   (  mapEntryList COMMA?
    |   COLON
   )
  RBRACK
;

mapEntryList
:  mapEntry (COMMA mapEntry)*
;

namedPropertyArgList
options { baseContext = mapEntryList; }
:  namedPropertyArg (COMMA namedPropertyArg)*
;

mapEntry
:  mapEntryLabel COLON nls expression
|  MUL COLON nls expression
;

namedPropertyArg
options { baseContext = mapEntry; }
:  namedPropertyArgLabel COLON nls expression
|  MUL COLON nls expression
;

namedArg
options { baseContext = mapEntry; }
:  namedArgLabel COLON nls expression
|  MUL COLON nls expression
;

mapEntryLabel
:  keywords
|  primary
typeArgumentsOrDiamond
 : LT GT
 | typeArguments
 ;

arguments
 : LPAREN enhancedArgumentListInPar? COMMA? rparen
 ;

argumentList
 options { baseContext = enhancedArgumentListInPar; }
 : firstArgumentListElement
   ( COMMA nls
   argumentListElement
   )*
 ;

enhancedArgumentList
 options { baseContext = enhancedArgumentListInPar; }
 : firstEnhancedArgumentListElement
   ( COMMA nls
   enhancedArgumentListElement
   )*
 ;

enhancedArgumentListInPar
 : enhancedArgumentListElement
   ( COMMA nls
   enhancedArgumentListElement
   )*
 ;

firstArgumentListElement
 options { baseContext = enhancedArgumentListElement; }
 : expressionListElement[true]
 | namedArg
 ;

argumentListElement
 options { baseContext = enhancedArgumentListElement; }
 : expressionListElement[true]
 | namedPropertyArg
 ;

firstEnhancedArgumentListElement
 options { baseContext = enhancedArgumentListElement; }

expressionListElement[true]
  | standardLambdaExpression
  | namedArg

enhancedArgumentListElement
  : expressionListElement[true]
  | standardLambdaExpression
  | namedPropertyArg

stringLiteral
  : StringLiteral

className
  : CapitalizedIdentifier

identifier
  : Identifier
  | CapitalizedIdentifier
  | VAR
  | IN

// | DEF
  | TRAIT
  | AS

builtinType
  : BuiltInPrimitiveType
  | VOID

keywords
  : ABSTRACT
  | AS
  | ASSERT
  | BREAK
  | CASE
  | CATCH
  | CLASS
  | CONST
  | CONTINUE
  | DEF
  | DEFAULT
  | DO
  | ELSE
<table>
<thead>
<tr>
<th>ENUM</th>
<th>EXTENDS</th>
<th>FINAL</th>
<th>FINALLY</th>
<th>FOR</th>
<th>GOTO</th>
<th>IF</th>
<th>IMPLEMENTS</th>
<th>IMPORT</th>
<th>IN</th>
<th>INSTANCEOF</th>
<th>INTERFACE</th>
<th>NATIVE</th>
<th>NEW</th>
<th>PACKAGE</th>
<th>RETURN</th>
<th>STATIC</th>
<th>STRICTFP</th>
<th>SUPER</th>
<th>SWITCH</th>
<th>SYNCHRONIZED</th>
<th>THIS</th>
<th>THROW</th>
<th>THROWS</th>
<th>TRANSIENT</th>
<th>TRAIT</th>
<th>THREADSAFE</th>
<th>TRY</th>
<th>VAR</th>
<th>VOLATILE</th>
<th>WHILE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| NullLiteral | BooleanLiteral |
|             |                |
|             |                |

| BuiltInPrimitiveType | VOID |
|                      |      |
|                      |      |

| PUBLIC | PROTECTED | PRIVATE | ; |
|        |           |         |  |

rparen : RPAREN ; 
nls
No license file was found, but licenses were detected in source scan.

* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* software distributed under the License is distributed on an
* Based on an original grammar released in the PUBLIC DOMAIN

**1.2180 jsreport-assets 1.7.0**

1.2180.1 Available under license:

MIT License

Copyright (c) 2016 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
1.2181 metrics 4.0.2

1.2182 process-nextick-args 1.0.7
1.2182.1 Available under license:

# Copyright (c) 2015 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

1.2183 moment 2.22.1
1.2183.1 Available under license:

MIT

1.2184 swagger-integration 2.0.7
1.2184.1 Available under license:

Copyright 2018 SmartBear Software

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at [apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0)

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
1.2185 xml-beans 4.0.0

1.2185.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

===========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache XmlBeans distribution. ==
===========================================================================

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:

Aside from contributions to the Apache XMLBeans project, this software also includes:

- one or more source files from the Apache Xerces-J and Apache Axis products, Copyright (c) 1999-2003 Apache Software Foundation

- W3C XML Schema documents Copyright 2001-2003 (c) World Wide Web Consortium (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University)

- resolver.jar from Apache Xml Commons project,
  Copyright (c) 2001-2003 Apache Software Foundation

1.2186 cups 2.1.3 4ubuntu0.10
1.2186.1 Available under license :

<HTML>
<!--- SECTION: Getting Started -->
<HEAD>
<TITLE>Apache License Version 2.0</TITLE>
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its
representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."</p>

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.</p>

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and
- You must cause any modified files to carry prominent notices stating that You changed the files; and
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding...
those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
<br/>
<br/>
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
</li>
</ol>

<p><strong><a name="contributions">5. Submission of Contributions</a></strong>. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.</p>

<p><strong><a name="trademarks">6. Trademarks</a></strong>. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.</p>

<p><strong><a name="no-warranty">7. Disclaimer of Warranty</a></strong>. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.</p>

<p><strong><a name="no-liability">8. Limitation of Liability</a></strong>. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.</p>

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.</p>

END OF TERMS AND CONDITIONS</p>

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.</p>

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

CUPS Exceptions to the Apache 2.0 License

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.
In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

</BODY>
</HTML>

Copyright 1992, 1993, 1994, 1997 Henry Spencer. All rights reserved. This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.

2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.

3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

CUPS

Copyright 2007-2019 by Apple Inc.
Copyright 1997-2007 by Easy Software Products.

CUPS and the CUPS logo are trademarks of Apple Inc.

The MD5 Digest code is Copyright 1999 Aladdin Enterprises.

The Kerberos support code ("KSC") is copyright 2006 by Jelmer Vernooij and is provided 'as-is', without any express or implied warranty. In no event will the author or Apple Inc. be held liable for any damages arising from the use of the KSC.

Sources files containing KSC have the following text at the top of each source file:
This file contains Kerberos support code, copyright 2006 by Jelmer Vernooij.

The KSC copyright and license apply only to Kerberos-related feature code in CUPS. Such code is typically conditionally compiled based on the present of the HAVE_GSSAPI preprocessor definition.

Permission is granted to anyone to use the KSC for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of the KSC must not be misrepresented; you must not claim that you wrote the original software. If you use the KSC in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

-- CUPS Exceptions to the Apache 2.0 License --

As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into an Object form of such source code, you may redistribute such embedded portions in such Object form without complying with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled forms of this Software with software that is licensed under the GPLv2 ("Combined Software") and if a court of competent jurisdiction determines that the patent provision (Section 3), the indemnity provision (Section 9) or other Section of the License conflicts with the conditions of the GPLv2, you may retroactively and prospectively choose to deem waived or otherwise exclude such Section(s) of the License, but only in their entirety and only with respect to the Combined Software.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
 incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2187.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2188 uuid 3.4.0

1.2189 decode-uri-component 0.2.0 0.2.0

1.2189.1 Available under license:

The MIT License (MIT)

Copyright (c) Sam Verschueren <sam.verschueren@gmail.com> (github.com/SamVerschueren)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2190 hbase-metrics 2.4.2
1.2190.1 Available under license:

Apache HBase - Metrics Implementation
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licenser" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, 
reproduction, and distribution of the Work otherwise complies with 
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, 
any Contribution intentionally submitted for inclusion in the Work 
by You to the Licensor shall be under the terms and conditions of 
this License, without any additional terms or conditions. 
Notwithstanding the above, nothing herein shall supersede or modify 
the terms of any separate license agreement you may have executed 
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade 
names, trademarks, service marks, or product names of the Licensor, 
extcept as required for reasonable and customary use in describing the 
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or 
agreed to in writing, Licensor provides the Work (and each 
Contributor provides its Contributions) on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or 
IMPLIED, including, without limitation, any warranties or conditions 
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A 
PARTICULAR PURPOSE. You are solely responsible for determining the 
appropriateness of using or redistributing the Work and assume any 
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, 
whether in tort (including negligence), contract, or otherwise, 
unless required by applicable law (such as deliberate and grossly 
negligent acts) or agreed to in writing, shall any Contributor be 
liable to You for damages, including any direct, indirect, special, 
incidental, or consequential damages of any character arising as a 
result of this License or out of the use or inability to use the 
Work (including but not limited to damages for loss of goodwill, 
work stoppage, computer failure or malfunction, or any and all 
other commercial damages or losses), even if such Contributor 
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing 
the Work or Derivative Works thereof, You may choose to offer, 
and charge a fee for, acceptance of support, warranty, indemnity, 
or other liability obligations and/or rights consistent with this 
License. However, in accepting such obligations, You may act only 
on Your own behalf and on Your sole responsibility, not on behalf 
of any other Contributor, and only if You agree to indemnify, 
defend, and hold each Contributor harmless for any liability 
incurred by, or claims asserted against, such Contributor by reason
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2191 annotations 3.0.1

1.2192 toposort 2.0.2 2.0.2
1.2192.1 Available under license :

Toposort - Topological sorting for node.js
Copyright (c) 2012 by Marcel Klehr <mklehr@gmx.net>
MIT LICENSE
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2193 ignite-indexing 2.11.0
1.2193.1 Available under license :

ignite-indexing
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
// ------------------------------------------------------------------
// List of ignite-indexing module's dependencies provided as a part of this distribution.
// ------------------------------------------------------------------

==============================================================================
For H2 Database Engine (http://www.h2database.com) com.h2database:h2:jar:1.4.197
==============================================================================
This product bundles H2 Database Engine which is available under the following:
MPL 2.0 or EPL 1.0. For details, see http://h2database.com/html/license.html.

==============================================================================
For ignite-core (http://ignite.apache.org) org.apache.ignite:ignite-core:jar:2.11.0
==============================================================================
This product bundles ignite-core which is available under the following:
The Apache Software License, Version 2.0. For details, see http://www.apache.org/licenses/LICENSE-2.0.txt.

==============================================================================
org.apache.lucene:lucene-analyzers-common:jar:7.4.0
==============================================================================
This product bundles Lucene Common Analyzers which is available under the following:
Apache 2. For details, see http://www.apache.org/licenses/LICENSE-2.0.txt.

==============================================================================
==============================================================================
This product bundles Lucene Core which is available under the following:
Apache 2. For details, see http://www.apache.org/licenses/LICENSE-2.0.txt.


This product bundles Lucene QueryParsers which is available under the following:
Apache 2. For details, see http://www.apache.org/licenses/LICENSE-2.0.txt.


This product bundles Apache CommonsCodec which is available under the following:
Apache License, Version 2.0. For details, see https://www.apache.org/licenses/LICENSE-2.0.txt.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2194 react 16.4.2 16.4.2
1.2194.1 Available under license :
MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2195 rimraf 2.4.5
1.2195.1 Available under license :
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2196 remark 1.0.0
1.2196.1 Available under license :

# The MIT License (MIT)

Copyright (c) 2018

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2197 xml2js-preserve-spaces 0.0.1

1.2197.1 Available under license:
Copyright 2010, 2011, 2012, 2013. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2198 asynckit 0.4.0 0.4.0

1.2198.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Alex Indigo

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
SOFTWARE.

1.2199 libxtst 1.2.2-1
1.2199.1 Available under license:

Copyright 1990, 1991 by UniSoft Group Limited

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation.

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall
not be used in advertising or otherwise to promote the sale, use or
other dealings in this Software without prior written authorization
from The Open Group.

***************************************************************************

Copyright 1995 Network Computing Devices

Permission to use, copy, modify, distribute, and sell this software and
its documentation for any purpose is hereby granted without fee, provided
that the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the name of Network Computing Devices
not be used in advertising or publicity pertaining to distribution
of the software without specific, written prior permission.

NETWORK COMPUTING DEVICES DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS, IN NO EVENT SHALL NETWORK COMPUTING DEVICES BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN
AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

***************************************************************************

Copyright 2005 Red Hat, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Red Hat not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Red Hat makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

RED HAT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL RED HAT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

***************************************************************************

Copyright 1992 by UniSoft Group Ltd.

Permission to use, copy, modify, and distribute this documentation for any purpose and without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. UniSoft makes no representations about the suitability for any purpose of the information in this document. This documentation is provided "as is" without express or implied warranty.

***************************************************************************

Copyright 1992, 1994, 1995 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

***************************************************************************

Copyright 1994 Network Computing Devices, Inc.

Permission to use, copy, modify, distribute, and sell this documentation for any purpose is hereby granted without fee, provided that the above copyright notice and this permission notice appear in all copies. Network Computing Devices, Inc. makes no representations about the suitability for any purpose of the information in this document. This documentation is provided "as is" without express or implied warranty.

1.2200 jersey-bean-validation 2.5.1

1.2201 aws-java-sdk-s3 1.11.704

1.2201.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3EncryptionClient.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/MultipartUploadCryptoContext.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/ContentCryptoMaterial.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/S3KeyWrapScheme.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/S3CryptoModuleAE.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/CryptoRuntime.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/S3ProgressPublisher.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/PersistableTransfer.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/AesGcm.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/exception/PauseException.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3Encryption.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/S3ProgressListener.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/S3ProgressListenerChain.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/PauseStatus.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/PersistableUpload.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/CipherLiteInputStream.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/AWSS3V4Signer.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/S3CryptoModuleBase.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/ContentCryptoScheme.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/GCMCipherLite.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/AesCbc.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-2020 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 *
 *    http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketAccelerateConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/DownloadTaskImpl.java

Open Source Used In DNAC Platform Antelope  7715
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019-2020 Amazon Technologies, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 * *
 * http://aws.amazon.com/apache2.0
 *
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketReplicationConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketLoggingConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketCrossOriginConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketTaggingConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketVersioningConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ListNextBatchOfVersionsRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketLifecycleConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ListNextBatchOfObjectsRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/FileLocks.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/TransferMeta.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 */
Open Source Used In DNAC Platform Antelope

* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/BucketLifecycleConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/ObjectRestoreResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/ResponseHeaderOverrides.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/SetBucketNotificationConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/BucketCrossOriginConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-

/*
* Copyright 2011-2020 Amazon Technologies, Inc.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at:
* 
* http://aws.amazon.com/apache2.0
* 
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
* OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/BucketLifecycleConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/ObjectRestoreResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/ResponseHeaderOverrides.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/SetBucketNotificationConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
**jar/com/amazonaws/services/s3/model/BucketCrossOriginConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 * 
 *   http://aws.amazon.com/apache2.0
 * 
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CopyPartRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteObjectsResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GenericBucketRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CORSRule.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/MultiObjectDeleteException.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ObjectExpirationResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CopyPartResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteObjectsResponse.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/ServerSideEncryptionResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/ResponseHandler.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/TransferMonitor.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteObjectsResponse.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/ObjectExpirationResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CopyPartResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/ServerSideEncryptionResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CopyPartResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ObjectExpirationResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CopyPartResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ObjectExpirationResult.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
 * for applicable license terms and NOTICE.txt for applicable notices.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/Grantee.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transform/XMLResponsesSaxParser.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/S3StringResponseHandler.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/BucketLoggingConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/AccessControlList.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/BucketPolicy.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/Grant.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/EmailAddressGrantee.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/BucketTaggingConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/RestUtils.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/RepeatableInputStream.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/BucketNotificationConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RequestPaymentConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CanonicalGrantee.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/TagSet.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/ServiceUtils.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/WebsiteConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/Bucket.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/Mimetypes.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/RepeatableFileInputStream.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/Owner.java

No license file was found, but licenses were detected in source scan.

/

* Copyright (c) 2016. Amazon.com, Inc. or its affiliates. All Rights Reserved.

* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0

* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/WebsiteConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/SkipMd5CheckStrategy.java

No license file was found, but licenses were detected in source scan.

/

* Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/KMSEncryptionMaterials.java
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2011-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not
 * use this file except in compliance with the License. A copy of the License is
 * located at
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on
 * an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/waiters/AmazonS3Waiters.java
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2012-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with
 * the License. A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR
* CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
* and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/inventory/InventoryEncryption.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteMarkerReplicationStatus.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ExistingObjectReplication.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/inventory/ServerSideEncryptionS3.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SourceSelectionCriteria.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetBucketEncryptionResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/inventory/ServerSideEncryptionKMS.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteBucketEncryptionRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SseKmsEncryptedObjectsStatus.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ServerSideEncryptionConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteBucketEncryptionResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/EncryptionConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ServerSideEncryptionRule.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketEncryptionRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketEncryptionResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetBucketEncryptionRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteMarkerReplication.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ExistingObjectReplicationStatus.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ServerSideEncryptionByDefault.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2011-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* *
* Licensed under the Apache License, Version 2.0 (the "License"). You may not
* use this file except in compliance with the License. A copy of the License is
* located at
* *
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed on
* an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/waiters/HeadBucketFunction.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/waiters/HeadObjectFunction.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2013-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* *
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
* for applicable license terms and NOTICE.txt for applicable notices.
* *
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* *
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/internal/AbstractRepeatableCipherInputStream.java
No license file was found, but licenses were detected in source scan.

/*
This appendix summarizes the current crypto related meta information associated with an S3 object encrypted using SDK client-side encryption. In SDK's terminology, these meta information are classified as "user metadata". In general, the meta information is just a map of key/value pairs. Physically they are persisted either as metadata of the encrypted S3 object, or as a separate S3 object called an "instruction file" (which is just a JSON file in plaintext.) For all references to the Cipher Algorithm Names, Modes, and Padding, such as <code>"AES/CBC/PKCS5Padding"</code>, please refer to Oracle's <a href="http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html">Java&trade; Cryptography Architecture Standard Algorithm Name Documentation</a> for more details.

Note when a key/value pair is stored as "user metadata" as part of the S3 metadata, the SDK always adds an additional prefix of <code>x-amz-meta-</code> to the key name. In contrast, the SDK doesn't add any prefix when the key names are stored in an instruction file. (The reason has to do with how the http headers are handled.)

For example, <code>x-amz-key</code> would get stored as <code>x-amz-meta-x-amz-key</code> in S3 metadata. In contrast, <code>x-amz-key</code> would simply be stored as is (ie without the prefix) in an instruction file.

All the keys described below are considered "user metadata" by the SDK.

For more details, see:

- <h2>S3 metadata vs instruction file</h2>
- <p>
  Note when a key/value pair is stored as "user metadata" as part of the S3 metadata, the SDK always adds an additional prefix of <code>x-amz-meta-</code> to the key name. In contrast, the SDK doesn't add any prefix when the key names are stored in an instruction file. (The reason has to do with how the http headers are handled.)
- For example, <code>x-amz-key</code> would get stored as <code>x-amz-meta-x-amz-key</code> in S3 metadata. In contrast, <code>x-amz-key</code> would simply be stored as is (ie without the prefix) in an instruction file.
- All the keys described below are considered "user metadata" by the SDK.
- <h2>S3 v1 crypto meta information (supported by all SDK's)</h2>
- <p>
For non-KMS client-side encryption, this v1 crypto meta information is used to support the "Encryption Only" crypto mode (which involves the use of \texttt{AES/CBC/PKCS5Padding} for content encryption, and \texttt{AES/ECB} for encrypting the one-time randomly generated data key). For KMS client-side encryption, the v2 crypto meta information is used.

This v2 crypto meta information is used to support both authenticated encryption (which involves the use of \texttt{AES/GCM/NoPadding} for content encryption, and key wrapping for the one-time randomly generated data key), and KMS client-side encryption (which can either be encryption-only or authenticated encryption.)

<table>
<thead>
<tr>
<th>key</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-key</td>
<td>Content encrypting key (cek) in encrypted form, base64 encoded. The cek is randomly generated per S3 object, and is always an AES 256-bit key. The corresponding cipher is always &quot;AES/CBC/PKCS5Padding&quot;.</td>
</tr>
<tr>
<td>x-amz-iv</td>
<td>Randomly generated IV (per S3 object), base64 encoded</td>
</tr>
<tr>
<td>x-amz-matdesc</td>
<td>Customer provided material description in JSON (UTF8) format. Used to identify the client-side master key (i.e., used to encrypt/wrap the generated content encrypting key).</td>
</tr>
<tr>
<td>x-amz-unencrypted-content-length</td>
<td>Unencrypted content length (optional but should be specified whenever possible).</td>
</tr>
</tbody>
</table>
| **x-amz-key-v2** | CEK in key wrapped form. This is necessary so that the S3 encryption
| **x-amz-iv** | Randomly generated IV (per S3 object), base64 encoded. (Same as v1.)
| **x-amz-matdesc** | Customer provided material description in JSON format. (Same as v1). Used to identify the client-side master key.
| **x-amz-unencrypted-content-length** | Unencrypted content length. (optional but should be specified whenever possible. Same as v1.)
| **x-amz-wrap-alg** | Key wrapping algorithm used.
| **x-amz-cek-alg** | Content encryption algorithm used.

*Supported values:
- "AESWrap",
- "RSA/ECB/OAEWithSHA-256AndMGF1Padding",
- "kms"

*No standard key wrapping is used if this meta information is absent
*Always set to "kms" if KMS is used for client-side encryption

*Supported values:
- "AES/GCM/NoPadding",
- "AES/CBC/PKCS5Padding"

*Default to "AES/CBC/PKCS5Padding" if this key is absent.
* <td>Tag length (in bits) when AEAD is in use.
* </td>
* <ul>
* <li>Only applicable if AEAD is in use. This meta information is absent
* otherwise, or if KMS is in use.</li>
* </ul>
* Supported value: `<code>"128"</code>`
* </table>
* </tr>
* </table>
* </td>
* </tr>
* </table>
* </td>
* </tr>
* </table>
* <p>
* <h3>KMS Integration for client-side encryption</h3>
* <ol>
* <li>All client-side KMS protected S3 objects are stored in v2 crypto meta information
* format. However, the key wrapping algorithm, `<b>x-amz-wrap-alg</b>` is always
* set to `<b>"kms"</b>`.</li>
* <li>The KMS Customer Master Key ID is currently stored as part of the
* material description, `<b>x-amz-matdesc</b>`, under the key-name
* `<b>"kms_cmk_id"</b>`.</li>
* </ol>
* </p>
* </h3>
* <p>
* <h4>Sample S3 crypto meta information for KMS client-side encryption</h4>
* <pre>
* x-amz-meta-x-amz-key-v2:
* CiA1J2GqwpJRnt4izuwoaic2YdNRNSMEslG5rw3pElJ8hKnuAQEBAwB4CdhsqsKSUZ7eIs7o8KGpQtmHTUTUjBLJRua8N6RCpiIAAAB+MHwGCSqGSib3DQEHBqBvMG0CAQAwaAYJKoZlhcNAQcBMB4GCWCGSAlFAwQBLjARBAyD0rzN3KHIIiVixDVCsCARCAOwhZ6xA8rob3ZH0mN3uKFiieHKVyiIC/PKlchALvGl0Vlnf86m
* cZJNpTXtwMkJkDocVj3Z1zGjZTuFtOsOH
* x-amz-iv: xqOHH+m6y0UogwG5SHJ3MQ==
* x-amz-uncrypted-content-length: 32768
* x-amz-cek-alg: AES/CBC/PKCS5Padding
* x-amz-wrap-alg: kms
* x-amz-matdesc: {
* "kms_cmk_id": "d3fd2273-4ca0-4da5-b1c2-c89742ecc6a26"
* }
* </pre>
* </h5>
* </h5>
* </p>
* </h5>
* <pre>
* x-amz-key-v2:
* CiA1J2GqwpJRnt4izuwoaic2YdNRNSMEslG5rw3pElJ8hKnuAQEBAwB4CdhsqsKSUZ7eIs7o8KGpQtmHTUTUjBLJRua8N6RCpiIAAAB+MHwGCSqGSib3DQEHBqBvMG0CAQAwaAYJKoZlhcNAQcBMB4GCWCGSAlFAwQBLjARBAytfVvYyW4i6QmwcPK1ECARCAO1VzoM6Kx9DHylNnM/BEbbje2vLA7IY790kUoSqdxgKxpNID9QaL/7928m1Djld0JpHefjm89C
* x-amz-iv: VZBksyij6DhrUMye
* x-amz-uncrypted-content-length: 32768

Open Source Used In DNAC Platform Antelope 7730
* x-amz-cek-alg: AES/GCM/NoPadding
* x-amz-wrap-alg: kms
* x-amz-matdesc: {"kms_cmk_id":"d3fd2273-4ca0-4da5-b1c2-c89742cc6a26"}
* x-amz-tag-len: 128
* </pre>

* <h2>S3 metadata of an instruction file</h2>
* <p>
* The following key is always included as part of the S3 metadata of the
* instruction file (which is just an S3 object). Since the SDK treats this key
* as "user metadata", the key name is always prefixed with <b>x-amz-meta-</b>
* when physically stored.
* </p>
* <table border=1>
* <tr aligned="left">
* <th>key</th>
* <th>description</th>
* </tr>
* <tr>
* <td width="200"><b>x-amz-crypto-instr-file</b></td>
* <td>the presence of this key is used to indicate this S3 object is an
* instruction file (of an associated client-side encrypted S3 object).</td>
* </tr>
* </table>
* */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* 
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
* for applicable license terms and NOTICE.txt for applicable notices.
* 
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* 
* http://aws.amazon.com/apache2.0
* 
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/
Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/SecuredCEK.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/KMSSecuredCEK.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/auth/DecodedStreamBuffer.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/S3ObjectIds.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/S3Event.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SSEAwsKeyManagementParams.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/PutInstructionFileRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SimpleMaterialProvider.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/InstructionFileId.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/NotificationConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SSECustomerKeyProvider.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/FileDeletionEvent.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/S3DataSource.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/PutObjectRequest.java
/*
 * Copyright 2013-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

No license file was found, but licenses were detected in source scan.

Found in path(s):
*/
* Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights
* Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

No license file was found, but licenses were detected in source scan.

Found in path(s):
*/
jar/com/amazonaws/services/s3/internal/auth/S3SignerProvider.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
/**
 * Gets the optional <code>versionIdMarker</code> parameter indicating where in the
 * sorted list of all versions in the specified bucket to begin returning
 * results. Results are always ordered first lexicographically (i.e.
 * alphabetically) and then from most recent version to least recent
 * version.
 * <p>
 * A key marker must be specified when specifying a <code>versionIdMarker</code>.
 * Results begin immediately after the version with the specified key and
 * version ID.
 * </p>
 *
 * @return The optional <code>versionIdMarker</code> parameter indicating where in the
 * sorted list of all versions in the specified bucket to begin
 * returning results.
 *
 * @see ListVersionsRequest#setVersionIdMarker(String)
 * @see ListVersionsRequest#withVersionIdMarker(String)
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/ListVersionsRequest.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/

Found in path(s):
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketWebsiteConfigurationRequest.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3ClientParamsWrapper.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/TransferManagerParams.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/Tier.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/BucketWebsiteConfiguration.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteBucketLifecycleConfigurationRequest.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RoutingRuleCondition.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ListObjectsV2Request.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ListObjectsV2Response.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketLocationRequest.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetBucketLocationResponse.java
*/opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RedirectRule.java

Open Source Used In DNAC Platform Antelope 7736
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3ClientConfigurationFactory.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GlacierJobParameters.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteBucketPolicyRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/TransferManagerBuilder.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RoutingRule.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3Builder.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3ClientParams.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3ClientConfigurationFactory.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3ClientParams.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/TransferManagerBuilder.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RoutingRule.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/TransferManagerBuilder.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RoutingRule.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/TransferManagerBuilder.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RoutingRule.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2020 Amazon Technologies, Inc.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 * 
 *   http://aws.amazon.com/apache2.0
 * 
 * This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES
 * OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/AbortIncompleteMultipartUpload.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/MD5DigestCalculatingInputStream.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/S3VersionSummary.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetBucketLifecycleConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ObjectListing.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/MultiFactorAuthentication.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/BucketVersioningConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DeleteVersionRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetBucketTaggingConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GroupGrantee.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/Permission.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/package-info.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetBucketCrossOriginConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/S3ObjectSummary.java
No license file was found, but licenses were detected in source scan.

/
* Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
permissions and limitations under the License.
*/

// The Amazon S3 DevPay token header is a special exception and can be safely moved

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/AmazonS3Client.java
No license file was found, but licenses were detected in source scan.

/
* Copyright 2012-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
* or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/StaticEncryptionMaterialsProvider.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/iterable/S3Versions.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/Download.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/DownloadImpl.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/Download.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/DownloadImpl.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/DownloadImpl.java
jar/com/amazonaws/services/s3/iterable/S3Objects.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

/**
 ** Interface for providing encryption materials.
 ** Implementations are free to use any strategy for providing encryption
 ** materials, such as simply providing static material that doesn't change,
 ** or more complicated implementations, such as integrating with existing
 ** key management systems.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/EncryptionMaterialsProvider.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Portions copyright 2006-2009 James Murty. Please see LICENSE.txt
 * for applicable license terms and NOTICE.txt for applicable notices.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
/*
 * Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * 
 * http://aws.amazon.com/apache2.0
 * 
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/SetObjectTaggingResponseHeaderHandler.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/GetObjectTaggingResponseHeaderHandler.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/S3VersionHeaderHandler.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transform/RequestPaymentConfigurationXmlFactory.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RestoreObjectRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/auth/policy/resources/S3ObjectResource.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transform/AnalyticsPredicateVisitorImpl.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CompleteMultipartUploadRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/PartSummary.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/S3MetadataResponseHandler.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/crypto/JceEncryptionConstants.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/MetadataDirective.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/MetadataEntry.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/UploadCallable.java
No license file was found, but licenses were detected in source scan.
Open Source Used In DNAC Platform Antelope 7743
Open Source Used In DNAC Platform Antelope 7744
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012-2020 Amazon Technologies, Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at:
 * http://aws.amazon.com/apache2.0
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES 
* OR CONDITIONS OF ANY KIND, either express or implied. See the 
* License for the specific language governing permissions and 
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/DownloadMonitor.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/MultipleFileUpload.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/MultipleFileDownload.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/MultipleFileUploadImpl.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/MultipleFileTransfer.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/MultipleFileTransferMonitor.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/MultilpleFileTransfer.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/MultipleFileDownloadImpl.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/S3ObjectInputStream.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * 
 * http://aws.amazon.com/apache2.0
 * 
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/eventstreaming/Utils.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/eventstreaming/Utils.java
jar/com/amazonaws/services/s3/internal/eventstreaming/HeaderType.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/internal/eventstreaming/Checksums.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/internal/eventstreaming/Prelude.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/internal/eventstreaming/HeaderValue.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/internal/eventstreaming/Header.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/SelectObjectContentEventException.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/transform/ObjectLockRetentionXmlFactory.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/SelectRecordsInputStream.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/Stats.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/Progress.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/PresignedUrlDownloadConfig.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/GetObjectLegalHoldResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/ObjectLockRetentionMode.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/PresignedUrlDownloadResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
jar/com/amazonaws/services/s3/model/SelectObjectContentResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetObjectLegalHoldResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SelectObjectContentEventVisitor.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetObjectLockConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetObjectLockLegalHoldRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/CompressionType.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/AbortableTransfer.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/GetObjectLegalHoldRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/PresignedUrlUploadRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/DefaultRetention.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ObjectLockConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetObjectRetentionRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ObjectLockLegalHold.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/JSONType.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/transfer/internal/PresignedUrlDownloadImpl.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/UploadObjectStrategy.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/ObjectLockLegalHoldStatus.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SelectObjectContentEvent.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetObjectRetentionResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/RequestProgress.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014-2020 Amazon Technologies, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
*
* You may obtain a copy of the License at:
*    http://aws.amazon.com/apache2.0
*
* This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/event/S3EventNotification.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*
* Portions copyright 2006-2009 James Murty. Please see LICENSE.txt for applicable license terms and NOTICE.txt for applicable notices.
*/

Found in path(s):
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/S3Object.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*

Found in path(s):
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/BucketAccelerateConfiguration.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/transform/ProgressStaxUnmarshaller.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/GetBucketNotificationConfigurationRequest.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/GetBucketReplicationConfigurationRequest.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/ReplicationRuleStatus.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/transform/LambdaConfigurationStaxUnmarshaller.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/transform/QueueConfigurationStaxUnmarshaller.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/transfer/TransferCompletionFilter.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/ReplicationRuleStatus.java
* /opt/cola/permissions/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-
  jar/com/amazonaws/services/s3/model/DeleteBucketReplicationConfigurationRequest.java

---

Open Source Used In DNAC Platform Antelope  7754
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/FilterRule.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/HeadBucketResult.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transform/HeadBucketResultHandler.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transform/NotificationConfigurationStaxUnmarshaller.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transform/FilterRuleStaxUnmarshaller.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transfer/exception/FileLockException.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/internal/CompleteMultipartUploadRetryCondition.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transform/StatsStaxUnmarshaller.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/Filter.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/S3KeyFilter.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/transform/FilterStaxUnmarshaller.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/LambdaConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/SetBucketAccelerateConfigurationRequest.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/BucketReplicationConfiguration.java
* /opt/cola/permits/1137154767_1614023256.44/0/aws-java-sdk-s3-1-11-704-sources-jar/com/amazonaws/services/s3/model/BucketReplicationConfiguration.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2017. Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
1.2202 is-bigint 1.0.1
1.2202.1 Available under license:
MIT License

Copyright (c) 2018 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2203 jaxb-api 2.3.1
1.2203.1 Available under license:
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.
1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.
2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.
4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall,
upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or
regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

---------
NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.
For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all
the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show
them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you
legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no
warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to
know that what they have is not the original, so that any problems introduced by others will not reflect on the
original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that
redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To
prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying
it may be distributed under the terms of this General Public License. The "Program", below, refers to any such
program or work, and a "work based on the Program" means either the Program or any derivative work under
copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications
and/or translated into another language. (Hereinafter, translation is included without limitation in the term
"modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its
scope. The act of running the Program is not restricted, and the output from the Program is covered only if its
contents constitute a work based on the Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium,
provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and
disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and
give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty
protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the
Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you
also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of
any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new
problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.
As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.2204 toidentifier 1.0.0

1.2204.1 Available under license:
MIT License

Copyright (c) 2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2205 fast-deep-equal 2.0.1

1.2205.1 Available under license:
MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2206 proxy-addr 1.0.10 1.0.10
1.2206.1 Available under license:
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2207 math-random 1.0.4
1.2207.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2208 is-callable 1.2.3
1.2208.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2209 xml2js 0.4.23 0.4.23
1.2209.1 Available under license:
Copyright 2010, 2011, 2012, 2013. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2210 jcip-annotation 1.0
1.2210.1 Available under license:
No license file was found, but licenses were detected in source scan.

<html>
<body>
<p>
Class, field, and method level annotations for describing thread-safety policies.
</p>

<p>
Three class-level annotations describe the <em>intended</em> thread-safety promises of a class:<br/>
<code>@Immutable</code>, <code>@ThreadSafe</code>, and <code>@NotThreadSafe</code>.
<code>@Immutable</code> means that the class is immutable,<br/>and implies <code>@ThreadSafe</code>.<br/><code>@NotThreadSafe</code> is optional;<br/>if a class is not annotated as thread-safe, it should be presumed not to be<br/>thread-safe, but if you want to make it extra clear, use<br/><code>@NotThreadSafe</code>.<br/>
</p>

<p>
These annotations are relatively unintrusive and are beneficial to<br/>both users and maintainers. Users can see immediately whether a class<br/>is thread-safe, and maintainers can see immediately whether<br/>thread-safety guarantees must be preserved. Annotations are also<br/>useful to a third constituency: tools. Static code-analysis tools may<br/>be able to verify that the code complies with the contract indicated<br/>by the annotation, such as verifying that a class annotated with<br/><code>@Immutable</code> actually is immutable.<br/>
</p>
Field and method annotations

The class-level annotations above are part of the public documentation for the class. Other aspects of a class's thread-safety strategy are entirely for maintainers and are not part of its public documentation.

Classes that use locking should document which state variables are guarded with which locks, and which locks are used to guard those variables. A common source of inadvertent non-thread-safety is when a thread-safe class consistently uses locking to guard its state, but is later modified to add either new state variables that are not adequately guarded by locking, or new methods that do not use locking properly to guard the existing state variables. Documenting which variables are guarded by which locks can help prevent both types of omissions.

The `@GuardedBy(lock)` annotation documents that a field or method should be accessed only with a specific lock held.

Copyright and license

This software is copyright (c) 2005 Brian Goetz and Tim Peierls and is released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5). The official home for this software is http://www.jcip.net. Any republication or derived work distributed in source code form must include the copyright and license notice.

Found in path(s):
* /opt/cola/permits/1135913176_1613629761.33/0/jcip-annotations-1-0-sources-2-jar/net/jcip/annotations/package.html

No license file was found, but licenses were detected in source scan.

* Copyright (c) 2005 Brian Goetz and Tim Peierls
* Released under the Creative Commons Attribution License
Open Source Used In DNAC Platform - Antelope

* (http://creativecommons.org/licenses/by/2.5)
* Official home: http://www.jcip.net

* Any republication or derived work distributed in source code form
* must include this copyright and license notice.
*/

Found in path(s):
* /opt/cola/permits/1135913176_1613629761.33/0/jcip-annotations-1-0-sources-2-jar/net/jcip/annotations/ThreadSafe.java
* /opt/cola/permits/1135913176_1613629761.33/0/jcip-annotations-1-0-sources-2-jar/net/jcip/annotations/GuardedBy.java
* /opt/cola/permits/1135913176_1613629761.33/0/jcip-annotations-1-0-sources-2-jar/net/jcip/annotations/Immutable.java
* /opt/cola/permits/1135913176_1613629761.33/0/jcip-annotations-1-0-sources-2-jar/net/jcip/annotations/NotThreadSafe.java

1.2211 json-path 1.1.1 1.1.1

1.2211.1 Available under license:
Copyright (c) 2014-2016 David Chester <david@fmail.co.uk>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2212 jackson-jaxrs-xml-provider 2.9.10

1.2212.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to
You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and
assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivates.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

1.2213 nocache 1.0.1 1.0.1

1.2213.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2214 zjsonpatch 0.3.6
1.2214.1 Available under license:

Apache License

Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2215 lucene-analyzers-common 7.4.0

1.2215.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensors" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
turn to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

/*
 * Copyright 2001-2004 Unicode, Inc.
 * 
 * Disclaimer
 * 
 * This source code is provided as is by Unicode, Inc. No claims are
 * made as to fitness for any particular purpose. No warranties of any
 * kind are expressed or implied. The recipient agrees to determine
 * applicability of information provided. If this file has been
 * purchased on magnetic or optical media from Unicode, Inc., the
 * sole remedy for any claim will be exchange of defective media
 * within 90 days of receipt.
 * 
 * Limitations on Rights to Redistribute This Code
 * 
 * Unicode, Inc. hereby grants the right to freely use the information
 * supplied in this file in the creation of products supporting the
 * Unicode Standard, and to make copies of this file in any form
 * for internal or external distribution as long as this notice
 * remains attached.
 */

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/  

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/  

Some code in core/src/java/org/apache/lucene/util/automaton was derived from Brics automaton sources available at www.brics.dk/automaton/. Here is the copyright from those sources:

/*
 * Copyright (c) 2001-2009 Anders Moeller
 * All rights reserved.
 */
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.

Here is the copyright for those sources:

# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from ICU (http://www.icu-project.org) 
The full license is available here: 
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.

**THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"**
**AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE**
**IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE**
**DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE**
**FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL**
**DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR**
**SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER**
**CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,**
**OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE**
**OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.**

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and
   "University of Massachusetts" must not be used to endorse or promote products
   derived from this software without prior written permission. To obtain
   permission, contact info@ciir.cs.umass.edu.

**THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS**
**"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,**
**THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE**
**ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE**
**LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR**
**CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE**
**GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)**
**HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT**
**LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY**
**OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF**
SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---

The dictionary comes from Morfologik project. Morfologik uses data from
Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and
is licenced on the terms of (inter alia) LGPL and Creative Commons
ShareAlike. The part-of-speech tags were added in Morfologik project and
are not found in the data from sjp.pl. The tagset is similar to IPI PAN
tagset.

---

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski, Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects, including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such as the Unicode Character Database. See http://unicode.org/copyright.html for more
Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were automatically generated with the moman/finenight FSA library, created by Jean-Philippe Barrette-LaPierre. This library is available under an MIT license, see http://sites.google.com/site/rrettesite/moman and http://bitbucket.org/jpharrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from http://snowball.tartarus.org/

The KStem stemmer in analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian, Romanian, Bulgarian, Hindi and Bengali analyzers (common) come with a default stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
The German, Spanish, Finnish, French, Hungarian, Italian, Portuguese, Russian and Swedish light stemmers (common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:

- analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil, and Edmond Nolan.

The Polish analyzer (stempel) comes with a default stopword list that is BSD-licensed created by the Carrot2 project. The file resides in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt. See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_.java (under modules/analysis/common/src/test/) is derived from Unicode data such as the Unicode Character Database. See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary (http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP) (http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===========================================================================
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===========================================================================

Open Source Used In DNAC Platform Antelope 7794
This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===========================================================================

mecab-ipadic-2.7.0-20070801 Notice

===========================================================================

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
IMPLIED, STATUTORY OR OTHERWISE. The term "wARRANTY" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

===========================================================================
Nori Korean Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from mecab-ko-dic-2.0.3-20170922

which can be obtained from

https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.0.3-20170922.tar.gz

1.2216 synchronous-promise 2.0.15 2.0.15
1.2216.1 Available under license:

Copyright (c) 2016, Davyd McColl
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of synchronous-promise nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2217 batik-svggen 1.13

1.2217.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/CachedImageHandlerJPEGEncoder.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/GlyfCompositeComp.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/CvtTable.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/CmapIndexEntry.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/HeadTable.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/SVGGraphicContextConverter.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/Feature.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/LocaTable.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/ImageHandlerPNGEncoder.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/FeatureRecord.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/AbstractSVGConverter.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/SVGClipDescriptor.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/Coverage.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/SVGGraphics2D.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/LangSys.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/SVGLine.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/RangeRecord.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/DefaultCachedImageHandler.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/CachedImageHandlerBase64Encoder.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/Ligature.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
Open Source Used In DNAC Platform Antelope 7799
Open Source Used in DNAC Platform Antelope

7800
Open Source Used In DNAC Platform Antelope 7801
Open Source Used In DNAC Platform Antelope 7803

jar/org/apache/batik/svggen/font/table/DirectoryEntry.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/CmapTable.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/FeatureList.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/SVGGraphicObjectConverter.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/StyleHandler.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/DefaultErrorHandler.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/HheaTable.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/ScriptRecord.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/DefaultErrorHandler.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/CmapFormat2.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/FeatureTags.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/NameTable.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/DefaultErrorHandler.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/CmapFormat0.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/SVGEllipse.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/SVGFilterConverter.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/SVGGraphics2DRuntimeException.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/CmapFormat6.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/Script.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/font/table/KernSubtableFormat0.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/SVGArc.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
jar/org/apache/batik/svggen/SVGStylingAttributes.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

No license file was found, but licenses were detected in source scan.

Found in path(s):


No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):

* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/CachedImageHandlerPNGEncoder.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/ImageHandler.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/SVGRenderingHints.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/SVGAlphaComposite.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/SVGSyntax.java
* /opt/cola/permits/1146690661_1616597247.74/0/batik-svggen-1-13-sources-1-jar/org/apache/batik/svggen/font/table/ClassDef.java
1.2218 ecc-jsbn 0.1.2 0.1.2

1.2218.1 Available under license:

Licensing
--------

This software is covered under the following copyright:

/*
 * Copyright (c) 2003-2005 Tom Wu
 * All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY
 * WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
 *
 * IN NO EVENT SHALL TOM WU BE LIABLE FOR ANY SPECIAL, INCIDENTAL,
 * INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER
 * RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF
 * THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT
 * OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * In addition, the following condition applies:
 *
 * All redistributions must retain an intact copy of this copyright notice
 * and disclaimer.
 */

Address all questions regarding this license to:

Tom Wu
tjw@cs.Stanford.EDU
The MIT License (MIT)
Copyright (c) 2014 Jeremie Miller

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2219 onetime 1.1.0

1.2219.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2220.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2221 fresh 0.3.0
1.2221.1 Available under license:
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2222 has-values 0.1.4
1.2222.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2223 jacoco-agent 0.7.9
1.2223.1 Available under license:

Found license 'Eclipse Public License 1.0' in 'All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0'

Found license 'Eclipse Public License 1.0' in 'under the terms and conditions of the Eclipse Public License Version 1.0 which is subject to the terms and conditions of the following license: All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holders nor the names of its this software without specific prior written permission.'
1.2224 parseurl 1.3.2 1.3.2
1.2224.1 Available under license :

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2225 netty-resolver 4.1.42.Final
1.2225.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 */
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/DefaultNameResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/InetSocketAddressResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/DefaultAddressResolverGroup.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/InetNameResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/HostsFileEntriesResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/DefaultHostsFileEntriesResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/AddressResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/CompositeNameResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/HostsFileParser.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/AbstractAddressResolver.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/ResolvedAddressTypes.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/HostsFileEntries.java

Open Source Used In DNAC Platform Antelope 7810
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-
jar/io/netty/resolver/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-
jar/io/netty/resolver/NoopAddressResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-
jar/io/netty/resolver/AddressResolverGroup.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-
jar/io/netty/resolver/SimpleNameResolver.java
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-
jar/io/netty/resolver/NameResolver.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE
2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/RoundRobinInetAddressResolver.java

1.2226 netty-buffer 4.1.42.Final
1.2226.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871174_1595214724.46/0/netty-resolver-4-1-42-final-sources-1-jar/io/netty/resolver/RoundRobinInetAddressResolver.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/PoolArenaMetric.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/PoolSubpageMetric.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/PoolChunkListMetric.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/PoolChunkMetric.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/UnpooledSlicedByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/UnpooledUnsafeHeapByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/PooledUnsafeHeapByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/UnsafeByteBufUtil.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/AbstractUnsafeSwappedByteBuf.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~    http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/FixedCompositeByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/ByteBufProcessor.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/EmptyByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/SimpleLeakAwareByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/ReadOnlyUnsafeDirectByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/AbstractReferenceCountedByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/ByteBufHolder.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/UnreleasableByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/AdvancedLeakAwareByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/ReadOnlyByteBuffer.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/WrappedByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/DefaultByteBufHolder.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/PooledUnsafeDirectByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/AbstractDerivedByteBuf.java

No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/AbstractByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/ReadOnlyByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/ByteBufUtil.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/PooledDirectByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/PoolChunk.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/UnpooledUnsafeDirectByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/ByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/PoolChunkList.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/ByteBufInputStream.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/UnpooledUnsafeByteBuf.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1.jar/io/netty/buffer/PoolThreadCache.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-jar/io/netty/buffer/UnsafeHeapSwappedByteBuf.java
/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 */

No license file was found, but licenses were detected in source scan.
* [http://www.apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0)

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/PooledByteBufAllocatorMetric.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/ByteBufAllocatorMetricProvider.java
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/ByteBufAllocatorMetric.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to the License at:
 *
 * [http://www.apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0)
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871254_1595214807.59/0/netty-buffer-4-1-42-final-sources-1-
  jar/io/netty/buffer/PooledHeapByteBuf.java

1.2227 kotlin 1.3.72

1.2227.1 Available under license:
No license file was found, but licenses were detected in source scan.

["version":3,"file":"kotlin.js","sources":[]"wrapper.js","arrayUtils.js","callableReferenceUtils.js","conversions.js","core.js","long.js","markerFunctions.js","misc.js","polyfills.js","rtti.js","arrayUtils.kt","runtime/Enum.kt","primitiveCompanionObjects.kt","generated/_Arrays.kt","generated/_Ranges.kt","kotlin/UByte.kt","kotlin/UInt.kt","kotlin/UShort.kt","Ranges.kt","collections/Collections.kt","collections/Maps.kt","collections/Sets.kt","text/StringNumberConversions.kt","kotlin/UnsignedUtils.kt","collections/Iterables.kt","collections/Sequences.kt","util/Preconditions.kt","generated/_ArraysJs.kt","comparisons/Comparisons.kt","kotlin/Comparator.kt","util/Standard.kt","generated/_Compar..."
Kotlin.toByte(Kotlin.numberToInt(a));

Kotlin.toChar(Kotlin.numberToInt(a));

Kotlin.toDouble(Kotlin.numberToDouble(a));

Kotlin.toChar(Kotlin.numberToInt(a));

Kotlin.doubleToInt(Kotlin.number);
\{\n  kind: \"class\",\n  simpleName: \"Long\",\n  interfaces:[\n  ]\n}\n
/*
 * A cache of the Long representations of small integer values.
 * @type {Object}\n * @private
 *\n * Returns a Long representing the given (32-bit) integer value.
 *\n * Converts this number value to \`Long\`.\n *\n * Returns zero if this \`Double\` value is \`NaN\`, \`Long.MIN_VALUE\` if it's less than \`Long.MIN_VALUE\`, \n * \`Long.MAX_VALUE\` if it's bigger than \`Long.MAX_VALUE\`.\n *\n * The number in question.\n *\n * The corresponding Long value.\n *\n *\n */
Kotlin.Long.IntCache_ = {};

/**
 * Returns a Long representing the 64-bit integer that comes by concatenating
 * the given high and
 * low bits.  Each is assumed to use 32 bits.
 *\n *\n */
Kotlin.Long.fromBits = function(lowBits, highBits) {
  return new Kotlin.Long(lowBits, highBits);
};

/**
 * Returns a Long representation of the given string, written using the given
 * radix.
 *\n *\n */
Kotlin.Long.fromString = function(str, opt_radix) {
  if (str.length == 0) {
    throw Error('number format error: empty string');
  }

  var radix = opt_radix || 10;
  if (radix < 2 || 36 < radix) {
    throw Error('radix out of range: ' + radix);
  }

  if (str.charAt(0) == '-') {
    return Kotlin.Long.fromString(str.substring(1), radix).negate();
  } else if (str.indexOf('-') >= 0) {
    throw Error('number format error: interior "-" character: ' + str);
  }

  // Do several (8) digits each time through the loop, so as to minimize
  // the calls to the very expensive emulated div.\n  var radixToPower = Kotlin.Long.fromNumber(Math.pow(radix, 8));

  var result = Kotlin.Long.ZERO;
  for (var i = 0; i < str.length; i += 8) {
    var size = Math.min(8, str.length - i);
    var value = parseInt(str.substring(i, i + size), radix);
    if (size < 8) {
      var power = Kotlin.Long.fromNumber(Math.pow(radix, size));
      result = result.multiply(power).add(Kotlin.Long.fromNumber(value));
    } else {
      result = result.multiply(radixToPower);
      result = result.add(Kotlin.Long.fromNumber(value));
    }
  }
  return result;
};

// NOTE: the compiler should inline these constant values below and then remove\n// these

/**
 * Number used repeated below in calculations.
 *\n *\n */
Kotlin.Long.TWO_PWR_16_DBL_ = 1 << 16;

/**
 *\n *\n */
Kotlin.Long.TWO_PWR_24_DBL_ = 1 << 24;

/**
 *\n *\n */

/**
 *\n *\n */
Kotlin.Long.TWO_PWR_31_DBL_ = Kotlin.Long.TWO_PWR_32_DBL_ / 2;

/**
 *\n *\n */

/**
 *\n *\n */
Kotlin.Long.TWO_PWR_63_DBL_ = Kotlin.Long.TWO_PWR_64_DBL_ / 2;

/**
 *\n */
Kotlin.Long.ZERO = Kotlin.Long.fromInt(0);

/**
 *\n */
Kotlin.Long.MIN_VALUE = Kotlin.Long.fromInt(-2147483648);

/**
 *\n */
Kotlin.Long.MAX_VALUE = Kotlin.Long.fromInt(2147483647);

/** @type {Object} */
Kotlin.Long.ZERO

/**
 *\n */
Kotlin.Long.TWO_PWR_64_DBL_ = Kotlin.Long.ZERO.

Open Source Used In DNAC Platform Antelope 7822
Kotlin.Long.ONE = Kotlin.Long.fromInt(1);

Kotlin.Long.NEG_ONE = Kotlin.Long.fromInt(-1);

Kotlin.Long.MAX_VALUE = Kotlin.Long.fromBits(0, 0x7FFFFFFF | 0);

Kotlin.Long.MIN_VALUE = Kotlin.Long.fromBits(0, 0x80000000 | 0);

Kotlin.Long.TWO_PWR_24_ = Kotlin.Long.fromInt(1 << 24);

Kotlin.Long.TWO_PWR_32_DBL_ = Kotlin.Long.TWO_PWR_24_.multiply(2);

Kotlin.Long.prototype.toInt = function() {
  return this.low_;
};

Kotlin.Long.prototype.toNumber = function() {
  return this.high_ * Kotlin.Long.TWO_PWR_32_DBL_ + this.getLowBitsUnsigned();
};

Kotlin.Long.prototype.hashCode = function() {
  return this.high_ ^ this.low_; // The bottom-most digit in this base and then
  // minimize the calls to the very expensive emulated div.
  var radixToPower = Kotlin.Long.fromNumber(Math.pow(radix, 6));
  var rem = this;
  var result = '';
  while (true) {
    var remDiv = rem.div(radixToPower);
    var intval = rem.subtract(remDiv.multiply(radixToPower)).toInt();
    var digits = intval.toString(radix);
    rem = remDiv;
    if (rem.isZero()) {
      return digits + result;
    } else {
      while (digits.length < 6) {
        digits = '0' + digits;
      }
      result = '' + digits + result;
    }
  }
};

Kotlin.Long.prototype.getHighBits = function() {
  return this.high_;
};

Kotlin.Long.prototype.getLowBits = function() {
  return this.low_;
};

Kotlin.Long.prototype.getLowBitsUnsigned = function() {
  return (this.low_ >= 0) ?
      this.low_ : Kotlin.Long.TWO_PWR_32_DBL_ + this.low_;
};

Kotlin.Long.prototype.getNumBitsAbs = function() {
  if (this.isNegative()) {
    if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
      return 64;
    } else {
      var val = this.high_ != 0 ? this.high_ : this.low_;
      for (var bit = 31; bit > 0; bit--) {
        if ((val & (1 << bit)) != 0) {
          break;
        }
      }
      return this.high_ != 0 ? bit + 33
        : bit + 1;
    }
  } else {
    return 64;
  }
};

Kotlin.Long.prototype.notEqualsLong = function(other) {
  return (this.high_ != other.high_) || (this.low_ != other.low_);
};

Kotlin.Long.prototype.lessThan = function(other) {
  return this.compare(other) < 0;
};

Kotlin.Long.prototype.lessThanOrEqual = function(other) {
  return this.compare(other) <= 0;
};

Kotlin.Long.prototype.equalsLong = function(other) {
  return (this.high_ == other.high_) && (this.low_ == other.low_);
};

Kotlin.Long.prototype.isOdd = function() {
  return (this.low_ & 1) == 1;
};

Kotlin.Long.prototype.isNegative = function() {
  return this.high_ < 0;
};

Kotlin.Long.prototype.isOdd = function() {
  return this.high_ & 1 == 1;
};

Kotlin.Long.prototype.isZero = function() {
  return this.high_ == 0 && this.low_ == 0;
};

Kotlin.Long.prototype.isOdd = function() {
  return this.high_ & 1 == 1;
};
Kotlin.Long.prototype.greaterThan = function(other) {
  return this.compare(other) > 0;
};

Kotlin.Long.prototype.greaterThanOrEqual = function(other) {
  return this.compare(other) >= 0;
};

Kotlin.Long.prototype.compare = function(other) {
  return this.subtract(other).isNegative() ? 1 : -1;
};

Kotlin.Long.prototype.negate = function() {
  if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return other.isOdd() ? Kotlin.Long.MIN_VALUE : Kotlin.Long.ZERO;
  } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return this.isOdd() ? Kotlin.Long.MIN_VALUE : Kotlin.Long.ZERO;
  }

  if (this.isNegative()) {
    if (other.isNegative()) {
      return this.negate().multiply(other negate());
    } else {
      return this.negate().multiply(other).negate();
    }
  } else if (other.isNegative()) {
    return this.multiply(other negate()).negate();
  }

  // If both longs are small, use float multiplication
  if (this.lessThan(Kotlin.Long.TWO_PWR_24_)) {
    return this.multiply(other negate());
  }

  // We can skip products that would overflow.
  var a48 = this.high_ >>> 16;
  var a32 = this.high_ & 0xFFFF;
  var a16 = this.low_ >>> 16;
  var a00 = this.low_ & 0xFFFF;
  var b48 = other.high_ >>> 16;
  var b32 = other.high_ & 0xFFFF;
  var b16 = other.low_ >>> 16;
  var b00 = other.low_ & 0xFFFF;
  var c48 = 0;
  var c32 = 0;
  var c16 = 0;
  var c00 = 0;
  var d48 = 0;
  var d32 = 0;
  var d16 = 0;
  var d00 = 0;

  if (this.equalsLong(Kotlin.Long.ZERO)) {
    return this.isOdd() ? Kotlin.Long.MIN_VALUE : Kotlin.Long.ZERO;
  } else if (other.equalsLong(Kotlin.Long.ZERO)) {
    return this.multiply(other negate());
  } else if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return this.isOdd() ? Kotlin.Long.NEG_VALUE : Kotlin.Long.ZERO;
  } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return this.multiply(other negate()).negate();
  }

  // Divide each number into 4 chunks of 16 bits, and then sum the chunks.
  var a48 = this.high_ >>> 16;
  var a32 = this.high_ & 0xFFFF;
  var a16 = this.low_ >>> 16;
  var a00 = this.low_ & 0xFFFF;
  var b48 = other.high_ >>> 16;
  var b32 = other.high_ & 0xFFFF;
  var b16 = other.low_ >>> 16;
  var b00 = other.low_ & 0xFFFF;
  var c48 = 0;
  var c32 = 0;
  var c16 = 0;
  var c00 = 0;
  var d48 = 0;
  var d32 = 0;
  var d16 = 0;
  var d00 = 0;

  if (this.equalsLong(Kotlin.Long.ZERO)) {
    return this.isOdd() ? Kotlin.Long.MIN_VALUE : Kotlin.Long.ZERO;
  } else if (other.equalsLong(Kotlin.Long.ZERO)) {
    return this.multiply(other negate());
  } else if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return this.isOdd() ? Kotlin.Long.NEG_VALUE : Kotlin.Long.ZERO;
  } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return this.multiply(other negate()).negate();
  }

  // Divide each number into 4 chunks of 16 bits, and then sum the chunks.
  var a48 = this.high_ >>> 16;
  var a32 = this.high_ & 0xFFFF;
  var a16 = this.low_ >>> 16;
  var a00 = this.low_ & 0xFFFF;
  var b48 = other.high_ >>> 16;
  var b32 = other.high_ & 0xFFFF;
  var b16 = other.low_ >>> 16;
  var b00 = other.low_ & 0xFFFF;
  var c48 = 0;
  var c32 = 0;
  var c16 = 0;
  var c00 = 0;
  var d48 = 0;
  var d32 = 0;
  var d16 = 0;
  var d00 = 0;

  if (this.equalsLong(Kotlin.Long.ZERO)) {
    return this.isOdd() ? Kotlin.Long.MIN_VALUE : Kotlin.Long.ZERO;
  } else if (other.equalsLong(Kotlin.Long.ZERO)) {
    return this.multiply(other negate());
  } else if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return this.isOdd() ? Kotlin.Long.NEG_VALUE : Kotlin.Long.ZERO;
  } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return this.multiply(other negate()).negate();
  }
function (other) {
  var thisRem = this.subtract(other.multiply(approx));
  var result = approx.add(thisRem.divide(other));
  return result;
}

} else if (other.equals(Kotlin.Long.ZERO)) {
  return Kotlin.Long.ZERO;
}

if (this.isNegative()) {
  if (other.isNegative()) {
    return this.negate().divide(other);  // At this point, we have |other| >= 2, so |this/other| < |MIN_VALUE|.
  } else {
    var approx = this.shiftRight(1); var rem = approx.subtract(other.multiply(approx));
    var result = approx.add(rem.divide(other));
    return result;
  }
} else if (other.isNegative()) {
  return Kotlin.Long.ZERO;
} else if (this.isNegative()) {
  return Kotlin.Long.ONE;
} else {
  var halfThis = this.shiftRight(1);
  var approx = halfThis.divide(other).shiftLeft(1); if (approx.equals(Kotlin.Long.ZERO)) {
    return other.isNegative() ? Kotlin.Long.ONE : Kotlin.Long.NEG_ONE;
  } else {
    var otherRem = approx.multiply(other);
    while (otherRem.isNegative() || otherRem.greaterThan(this)) {
      approx -= 1; otherRem = approx.multiply(other);
    }
    // We know the answer can’t be zero... and actually, zero would cause
    // infinite recursion since we would make no progress.
    if (approx.isZero()) {
      approx = Kotlin.Long.ONE;
    }
    return approx;
  }
}

/**
 * Returns this Long modulo the given one.
 * @param {Kotlin.Long} other Long by which to mod.
 * @return {!Kotlin.Long} This Long modulo the given one.
 */
Kotlin.Long.prototype.modulo = function(other) {
  return this.subtract(this.divide(other).multiply(other));
};

/** @return {!Kotlin.Long} The bitwise-NOT of this value. */
Kotlin.Long.prototype.not = function() {
  return Kotlin.Long.fromBits(~this.low_, ~this.high_);
};

/**
 * Returns the bitwise-AND of this Long and the given one.
 * @param {Kotlin.Long} other The Long with which to AND.
 * @return {!Kotlin.Long} The bitwise-AND of this and the other.
 */
Kotlin.Long.prototype.and = function(other) {
  return Kotlin.Long.fromBits(this.low_ & other.low_,
                               this.high_ & other.high_);
};

/**
 * Returns the bitwise-XOR of this Long and the given one.
 * @param {Kotlin.Long} other The Long with which to XOR.
 * @return {!Kotlin.Long} The bitwise-XOR of this and the other.
 */
Kotlin.Long.prototype.xor = function(other) {
  return Kotlin.Long.fromBits(this.low_ ^ other.low_,
                               this.high_ ^ other.high_);
};

/**
 * Returns this Long with bits shifted to the left by the given amount.
 * @param {number} numBits The number of bits by which to shift.
 * @return {!Kotlin.Long} This shifted to the left by the given amount.
 */
Kotlin.Long.prototype.shiftLeft = function(numBits) {
  numBits &= 63;
  if (numBits == 0) {
    return this;
  } else {
    var high = this.high_;
    if (numBits < 32) {
      var low = this.low_;
      return Kotlin.Long.fromBits(high, low << (32 - numBits));
    } else {
      return Kotlin.Long.fromBits(0, low << (numBits - 32));
    }
  }
};

/**
 * Returns this Long with bits shifted to the right by the given amount.
 * @param {number} numBits The number of bits by which to shift.
 * @return {!Kotlin.Long} This shifted to the right by the given amount.
 */
Kotlin.Long.prototype.shiftRight = function(numBits) {
  numBits &= 63;
  if (numBits == 0) {
    return this;
  } else {
    var high = this.high_;
    if (numBits < 32) {
      var low = this.low_;
      return Kotlin.Long.fromBits(low, high >> numBits);
    } else {
      return Kotlin.Long.fromBits(high >> numBits);  // We will tweak the approximate result by changing it in the 48-th digit or
      // the smallest non-fractional digit, whichever is larger.
  var log2 = Math.ceil(Math.log(approx) / Math.LN2);
  var delta = (log2 <= 48) ? 1 : Math.pow(2, 48 - log2);
  // Decrease the approximation until it is smaller than the remainder. Note
  // that if it is too large, the product overflows and is negative.
  var approxRes = Kotlin.Long.fromNumber(approx);
  var approxRem = approxRem.multiply(other);
  while (approxRem.isNegative() || approxRem.greaterThan(res)) {
    approx -= delta;
    approxRes = Kotlin.Long.fromNumber(approx);
    approxRem = approxRes.multiply(other);
  }
  // We know the answer can’t be zero... and actually, zero would cause
  // infinite recursion since we would make no progress.
  if (approxRes.isZero()) {
    approxRes = Kotlin.Long.ONE;
  }
  return approxRes.add(approxRem);
}

*/

Open Source Used In DNAC Platform Antelope 7825
/**
 * Returns this Long with bits shifted to the right by the given amount, with
 * zeros placed into the new leading bits.
 * @param {number} numBits The number of bits by which to shift.
 * @return {Kotlin.Long} This shifted to the right by the given amount, with
 *           zeros placed into the new leading bits.
 */
Kotlin.Long.prototype.shiftRightUnsigned = function(numBits) {
  numBits &= 63;
  if (numBits == 0) {
    return this;
  } else if (numBits == 32) {
    return Kotlin.Long.fromBits(high, 0);
  } else if (numBits >= 0) {
    var high = this.high_;
    var low = this.low_;
    return Kotlin.Long.fromBits((low >>> numBits) | (high << (32 - numBits)),
                                high >>> numBits);
  } else {
    var high = this.high_;
    var low = this.low_;
    return Kotlin.Long.fromBits(low, 0);
  }
};

// Support for Kotlin
Kotlin.Long.prototype.equals = function (other) {
    return other instanceof Kotlin.Long &&
    this.equalsLong(other);
};

Kotlin.Long.prototype.compareTo = function (other) {
    return Kotlin.compareTo(this, other);
};

Kotlin.Long.prototype.inc = function() {
    return this.add(Kotlin.Long.ONE);
};

Kotlin.Long.prototype.dec = function() {
    return this.add(Kotlin.Long.NEG_ONE);
};

Kotlin.Long.prototype.valueOf = function() {
    return this.toNumber();
};

Kotlin.Long.prototype.unaryPlus = function() {
    return this;
};

Kotlin.Long.prototype.unaryMinus = Kotlin.Long.prototype.negate;
Kotlin.Long.prototype.inv = Kotlin.Long.prototype.not;

Kotlin.Long.prototype.rangeTo = function (other) {
    return new Kotlin.kotlin.ranges.LongRange(this, other);
};

/**
 * @param {number} numBits
 * @param {number} shift
 */
Kotlin.Long.prototype.shiftLeftSigned = function(numBits, shift) {
  numBits &= 63;
  if (numBits == 0) {
    return this;
  } else if (numBits >= 32) {
    var high = this.high_;
    var low = this.low_;
    return Kotlin.Long.fromBits((low << numBits) | (high >>> (32 - numBits)),
                                low << numBits);
  } else if (numBits == 32) {
    return Kotlin.Long.fromBits(high, 0);
  } else {
    var high = this.high_;
    var low = this.low_;
    return Kotlin.Long.fromBits(high >>> numBits, low << shift);
  }
};

Kotlin.Long.prototype.equalsLong = function (other) {
    return other instanceof Kotlin.Long &&
    this.equalsLong(other);
};

Kotlin.Long.prototype.shrink = function (amount) {
  if (amount == 0) {
    return this;
  }
  var high = this.high_;
  var low = this.low_;
  return Kotlin.Long.fromBits((low >>> amount) | (high << (32 - amount)),
                               low >>> amount);
};

Kotlin.Long.prototype.shrinkCopy = function (amount) {
  if (amount == 0) {
    return this;
  }
  var high = this.high_;
  var low = this.low_;
  return Kotlin.Long.fromBits((low >>> amount) | (high << (32 - amount)),
                               low >>> amount);
};

Kotlin.Long.prototype.shrinkInPlace = function (amount) {
  if (amount == 0) {
    return this;
  }
  var high = this.high_;
  var low = this.low_;
  this.high_ = high >>> amount;
  this.low_ = low >>> amount;
};

Kotlin.Long.prototype.toNumber = function() {
    return this;
};

Kotlin.Long.prototype.valueOf = function() {
    return this;
};

Kotlin.Long.prototype.unaryPlus = function() {
    return this;
};

Kotlin.Long.prototype.unaryMinus = Kotlin.Long.prototype.negate;
Kotlin.Long.prototype.inv = Kotlin.Long.prototype.not;

Kotlin.Long.prototype.rangeTo = function (other) {
    return new Kotlin.kotlin.ranges.LongRange(this, other);
};
Kotlin.isNumber = function (a) {
    return typeof a == "number" || a instanceof Kotlin.Long;
}

Kotlin.isChar = function (value) {
    return value instanceof Kotlin.BoxedChar
};

Kotlin.isComparable = function (value) {
    var type = typeof value;

    return type === "string" ||
           type === "boolean" ||
           Kotlin.isNumber(value) ||
           Kotlin.isType(value, Kotlin.kotlin.Comparable);
};

Kotlin.isCharSequence = function (value) {
    return typeof value === "string" ||
           Kotlin.isType(value, Kotlin.kotlin.CharSequence);
};
```kotlin
[IndexOfOutOfBoundsException] except in Kotlin/JS

@kotlin.internal.InlineOnly\public inline operator fun <T> Array<out T>.component2(): T {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun ByteArray.component2(): Byte {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun ShortArray.component2(): Short {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun IntArray.component2(): Int {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun LongArray.component2(): Long {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun FloatArray.component2(): Float {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun DoubleArray.component2(): Double {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun BooleanArray.component2(): Boolean {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun CharArray.component2(): Char {\n    return get(1)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun <T> Array<out T>.component3(): T {\n    return get(2)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun ByteArray.component3(): Byte {\n    return get(2)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun ShortArray.component3(): Short {\n    return get(2)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun IntArray.component3(): Int {\n    return get(2)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun LongArray.component3(): Long {\n    return get(2)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun FloatArray.component3(): Float {\n    return get(2)\n}\n
@kotlin.internal.InlineOnly\public inline operator fun DoubleArray.component3(): Double {\n    return get(2)\n}\n```

Open Source Used in DNAC Platform Antelope 7831
*@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component3(): Boolean {
    return get(2)
}

/**
 * Returns 3rd *element* from the array.
 * 
 * If the size of this array is less than 3, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component3(): Char {
    return get(2)
}

* @kotlin.internal.InlineOnly
public inline operator fun <T> Array<out T>.component4(): T {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun ByteArray.component4(): Byte {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun ShortArray.component4(): Short {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun IntArray.component4(): Int {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun LongArray.component4(): Long {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun FloatArray.component4(): Float {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component4(): Double {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component4(): Boolean {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component4(): Char {
    return get(3)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> Array<out T>.component5(): T {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun ByteArray.component5(): Byte {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun ShortArray.component5(): Short {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun IntArray.component5(): Int {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun LongArray.component5(): Long {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun FloatArray.component5(): Float {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component5(): Double {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component5(): Boolean {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component5(): Char {
    return get(4)
}
[IndexOutOfBoundsException] except in Kotlin/JS

*\@kotlin.internal.InlineOnly\public inline operator fun FloatArray.component5(): Float { return get(4)\n\}

*\@kotlin.internal.InlineOnly\public inline operator fun DoubleArray.component5(): Double { return get(4)\n\}

*\@kotlin.internal.InlineOnly\public inline operator fun BooleanArray.component5(): Boolean { return get(4)\n\}

*\@kotlin.internal.InlineOnly\public inline operator fun CharArray.component5(): Char { return get(4)\n\}

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun FloatArray.component5(): Float {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component5(): Double {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component5(): Boolean {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component5(): Char {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun FloatArray.component5(): Float {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component5(): Double {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component5(): Boolean {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an
 [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component5(): Char {
    return get(4)
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.contains(element: T): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun ByteArray.contains(element: Byte): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun ShortArray.contains(element: Short): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun IntArray.contains(element: Int): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun LongArray.contains(element: Long): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun FloatArray.contains(element: Float): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun DoubleArray.contains(element: Double): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun BooleanArray.contains(element: Boolean): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns true if [element] is found in the array.
 */
public operator fun CharArray.contains(element: Char): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun <T> Array<out T>.elementAt(index: Int): T

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun ByteArray.elementAt(index: Int): Byte

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun ShortArray.elementAt(index: Int): Short

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun IntArray.elementAt(index: Int): Int

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun LongArray.elementAt(index: Int): Long

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun FloatArray.elementAt(index: Int): Float

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun DoubleArray.elementAt(index: Int): Double

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun BooleanArray.elementAt(index: Int): Boolean

/**
 * Returns an element at the given [index] or throws an
 [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
public expect fun CharArray.elementAt(index: Int): Char

Open Source Used In DNAC Platform Antelope  7833
Char\n\n/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
 * [index] is out of bounds of this array.\n * @sample
 * samples.collections.Collections.Elements.elementAtOrElse
 */
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
 * [index] is out of bounds of this array.\n * @sample
 * samples.collections.Collections.Elements.elementAtOrElse
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
 * [index] is out of bounds of this array.\n * @sample
 * samples.collections.Collections.Elements.elementAtOrElse
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Short): Short {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
 * [index] is out of bounds of this array.\n * @sample
 * samples.collections.Collections.Elements.elementAtOrElse
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Int): Int {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * @sample
 * samples.collections.Collections.Elements.elementAtOrNull
 */
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.elementAtOrNull(index: Int): T? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * @sample
 * samples.collections.Collections.Elements.elementAtOrNull
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.elementAtOrNull(index: Int): Byte? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * @sample
 * samples.collections.Collections.Elements.elementAtOrNull
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.elementAtOrNull(index: Int): Short? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * @sample
 * samples.collections.Collections.Elements.elementAtOrNull
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.elementAtOrNull(index: Int): Int? {
    return this.getOrNull(index)
}
IntArray.elementAtOrNull(index: Int): Int? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * @sample
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.elementAtOrNull(index: Int): Long? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * @sample
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.elementAtOrNull(index: Int): Float? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * @sample
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.elementAtOrNull(index: Int): Double? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * @sample
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.elementAtOrNull(index: Int): Boolean? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * @sample
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.elementAtOrNull(index: Int): Char? {
    return this.getOrNull(index)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.find(predicate: (Byte) -> Boolean): Byte? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.find(predicate: (Short) -> Boolean): Short? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.find(predicate: (Int) -> Boolean): Int? {
    return firstOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.findLast(predicate: (Short) -> Boolean): Short? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.findLast(predicate: (Int) -> Boolean): Int? {
    return lastOrNull(predicate)
}
Boolean): Long? {\n    return lastOrNull(predicate)\n}\n/**
 * Returns the last element matching the given [predicate], or ‘null’ if no such element was found.\n */
@kotlin.internal.InlineOnly\npublic inline fun FloatArray.findLast(predicate: (Float) -> Boolean): Float? {
    return lastOrNull(predicate)\n}\n/**
 * Returns the last element matching the given [predicate], or ‘null’ if no such element was found.\n */
@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.findLast(predicate: (Double) -> Boolean): Double? {
    return lastOrNull(predicate)\n}\n/**
 * Returns the last element matching the given [predicate], or ‘null’ if no such element was found.\n */
@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {
    return lastOrNull(predicate)\n}\n/**
 * Returns the last element matching the given [predicate], or ‘null’ if no such element was found.\n */
@kotlin.internal.InlineOnly\npublic inline fun CharArray.findLast(predicate: (Char) -> Boolean): Char? {
    return lastOrNull(predicate)\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun <T> Array<out T>.first(predicate: (T) -> Boolean): T {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun ByteArray.first(predicate: (Byte) -> Boolean): Byte {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun ShortArray.first(predicate: (Short) -> Boolean): Short {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun IntArray.first(predicate: (Int) -> Boolean): Int {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun LongArray.first(predicate: (Long) -> Boolean): Long {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun FloatArray.first(): Float {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")\n    return this[0]\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun DoubleArray.first(): Double {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")\n    return this[0]\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun BooleanArray.first(): Boolean {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")\n    return this[0]\n}\n/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun CharArray.first(): Char {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")\n    return this[0]\n}
public inline fun FloatArray.first(predicate: (Float) -> Boolean): Float {
    for (element in this) if 
        (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.\")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun DoubleArray.first(predicate: (Double) -> Boolean): Double {
    for (element in this) if 
        (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.\")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun BooleanArray.first(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if 
        (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.\")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun CharArray.first(predicate: (Char) -> Boolean): Char {
    for (element in this) if 
        (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.\")
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun <T> Array<out T>.firstOrNull(predicate: (T) -> Boolean): T? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun ByteArray.firstOrNull(predicate: (Byte) -> Boolean): Byte? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun ShortArray.firstOrNull(predicate: (Short) -> Boolean): Short? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun IntArray.firstOrNull(predicate: (Int) -> Boolean): Int? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun LongArray.firstOrNull(predicate: (Long) -> Boolean): Long? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun FloatArray.firstOrNull(predicate: (Float) -> Boolean): Float? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun DoubleArray.firstOrNull(predicate: (Double) -> Boolean): Double? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun BooleanArray.firstOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or ‘null’ if element was not found.
 * @throws
 * [NoSuchElementException] if no such element is found.
 */
public inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {
    for (element in this) if (predicate(element)) return element
    return null
}
Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

```kotlin
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.getOrElse(index: Int, defaultValue: (Int) -> T): T {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun ByteArray.getOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun ShortArray.getOrElse(index: Int, defaultValue: (Int) -> Short): Short {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun IntArray.getOrElse(index: Int, defaultValue: (Int) -> Int): Int {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun LongArray.getOrElse(index: Int, defaultValue: (Int) -> Long): Long {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun FloatArray.getOrElse(index: Int, defaultValue: (Int) -> Float): Float {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun DoubleArray.getOrElse(index: Int, defaultValue: (Int) -> Double): Double {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun BooleanArray.getOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun CharArray.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun <T> Array<out T>.getOrNull(index: Int): T? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun ByteArray.getOrNull(index: Int): Byte? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun ShortArray.getOrNull(index: Int): Short? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun IntArray.getOrNull(index: Int): Int? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun LongArray.getOrNull(index: Int): Long? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun FloatArray.getOrNull(index: Int): Float? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun DoubleArray.getOrNull(index: Int): Double? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun BooleanArray.getOrNull(index: Int): Boolean? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
@kotlin.internal.InlineOnly
public fun CharArray.getOrNull(index: Int): Char? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```
<lastIndex) get(index) else null


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.}


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.


* Returns first index of [element], or -1 if the array does not contain element.
CharArray.indexOfFirst(predicate: (Char) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.

ArrayindexOfLast(predicate: (T) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.

Array.indexOfLast(predicate: (Byte) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.

IntArray.indexOfLast(predicate: (Int) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.

ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.

CharArray.indexOfLast(predicate: (Char) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.

IntArray.indexOfFirst(predicate: (Int) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.

ByteArray.indexOfFirst(predicate: (Byte) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.

CharArray.indexOfFirst(predicate: (Char) -> Boolean): Int

if (predicate(this[index])) {
    return index
} else {
    return -1
}

* Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.

Open Source Used In DNAC Platform Antelope 7840
CharArray.last(): Char {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[lastIndex]}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun <T> Array<T>.last(predicate: (T) -> Boolean): T {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns last index of [element], or -1 if the array does not contain element.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Array<T>.lastIndexOf(element: T): Int {
    if (element == null) {
        for (index in indices.reversed()) {
            if (this[index] == null) {
                return index
            }
        }
    } else {
        for (index in indices.reversed()) {
            if (element == this[index]) {
                return index
            }
        }
    }
    return -1
}

Open Source Used In DNAC Platform Antelope 7841
Open Source Used In DNAC Platform Antelope 7842
last element matching the given [predicate], or `null` if no such element was found.\n
```kotlin
public inline fun BooleanArray.lastOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

public inline fun CharArray.lastOrNull(predicate: (Char) -> Boolean): Char? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns a random element from this array.
 * 
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(random: Random): T {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun ByteArray.random(random: Random): Byte {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun ShortArray.random(random: Random): Short {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun IntArray.random(random: Random): Int {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun LongArray.random(random: Random): Long {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun FloatArray.random(random: Random): Float {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun DoubleArray.random(random: Random): Double {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun BooleanArray.random(random: Random): Boolean {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

public fun CharArray.random(random: Random): Char {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

```
(isEmpty()) throw NoSuchElementException("Array is empty.")
return get(random.nextInt(size))

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
public fun DoubleArray.random(random: Random): Double {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
public fun BooleanArray.random(random: Random): Boolean {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
public fun CharArray.random(random: Random): Char {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.randomOrNull(random: Random): T? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun ByteArray.randomOrNull(): Byte? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun ShortArray.randomOrNull(): Short? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun IntArray.randomOrNull(): Int? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun LongArray.randomOrNull(): Long? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun FloatArray.randomOrNull(): Float? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun DoubleArray.randomOrNull(): Double? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun BooleanArray.randomOrNull(): Boolean? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun CharArray.randomOrNull(): Char? {
    return randomOrNull(Random)
}
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun IntArray.randomOrNull(random: Random): Int?
    { if (isEmpty()) return null
        return get(random.nextInt(size))
    }

/**
 * Returns a random element from this array using the specified source of randomness, or 'null' if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun LongArray.randomOrNull(random: Random): Long?
    { if (isEmpty()) return null
        return get(random.nextInt(size))
    }

/**
 * Returns a random element from this array using the specified source of randomness, or 'null' if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun FloatArray.randomOrNull(random: Random): Float?
    { if (isEmpty()) return null
        return get(random.nextInt(size))
    }

/**
 * Returns a random element from this array using the specified source of randomness, or 'null' if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun DoubleArray.randomOrNull(random: Random): Double?
    { if (isEmpty()) return null
        return get(random.nextInt(size))
    }

/**
 * Returns a random element from this array using the specified source of randomness, or 'null' if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun BooleanArray.randomOrNull(random: Random): Boolean?
    { if (isEmpty()) return null
        return get(random.nextInt(size))
    }

/**
 * Returns a random element from this array using the specified source of randomness, or 'null' if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun CharArray.randomOrNull(random: Random): Char?
    { if (isEmpty()) return null
        return get(random.nextInt(size))
    }

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun <T> Array<out T>.single(): T
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun ByteArray.single(): Byte
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun ShortArray.single(): Short
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun IntArray.single(): Int
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun LongArray.single(): Long
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun FloatArray.single(): Float
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun DoubleArray.single(): Double
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun BooleanArray.single(): Boolean
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun CharArray.single(): Char
    { when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
inline fun <T> Array<out T>.single(predicate: (T) -> Boolean): T {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as T
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun ByteArray.single(predicate: (Byte) -> Boolean): Byte {
    var single: Byte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Byte
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun ShortArray.single(predicate: (Short) -> Boolean): Short {
    var single: Short? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Short
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun IntArray.single(predicate: (Int) -> Boolean): Int {
    var single: Int? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Int
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun LongArray.single(predicate: (Long) -> Boolean): Long {
    var single: Long? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Long
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun FloatArray.single(predicate: (Float) -> Boolean): Float {
    var single: Float? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Float
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {
    var single: Double? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Double
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun BooleanArray.single(predicate: (Boolean) -> Boolean): Boolean {
    var single: Boolean? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Boolean
}
matching element

*\*anonymous inline fun CharArray.single(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.first(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.findFirst(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.findFirstOrNull(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.last(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.lastOrNull(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.findLast(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray.findLastOrNull(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findFirstMatchingPredicate(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findFirstOrNullMatchingPredicate(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findLastMatchingPredicate(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findLastOrNullMatchingPredicate(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findFirstMatchingElementMatchingPredicate(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findFirstOrNullMatchingElementMatchingPredicate(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findLastMatchingElementMatchingPredicate(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findLastOrNullMatchingElementMatchingPredicate(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findFirstMatchingElementMatchingPredicateCaseInsensitive(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findFirstOrNullMatchingElementMatchingPredicateCaseInsensitive(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findLastMatchingElementMatchingPredicateCaseInsensitive(predicate: (Char) -> Boolean): Char {
    \n    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}

*\*anonymous inline fun CharArray戆 findLastOrNullMatchingElementMatchingPredicateCaseInsensitive(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            \n            single = element
            found = true
        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    } \n    @Suppress("UNCHECKED_CAST")
    return single as Char
}
if (predicate(element)) {\n    \n    single = element\n    found = true\n    if (!found) return null\n    return single\n}
\n//public inline fun

/**
 * Returns the single element matching the given [predicate], or 'null' if element was not found or more than one element was found.
 */
public inline fun BooleanArray.singleOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    var single = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns the single element matching the given [predicate], or 'null' if element was not found or more than one element was found.
 */
public inline fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? {
    var single = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun <T> Array<out T>.drop(n: Int): List<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun ByteArray.drop(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun ShortArray.drop(n: Int): List<Short> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun IntArray.drop(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun LongArray.drop(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun FloatArray.drop(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun DoubleArray.drop(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun BooleanArray.drop(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun CharArray.drop(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun LongArray.drop(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun FloatArray.drop(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun DoubleArray.drop(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun BooleanArray.drop(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun CharArray.drop(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun ShortArray.drop(n: Int): List<Short> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun IntArray.drop(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun ByteArray.drop(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun ByteArray.drop(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun ByteArray.drop(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample
 */
public fun ByteArray.drop(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}

samples.collections.Collections.Transformations.drop\n */pubic fun <T> Array<out T>.dropLast(n: Int): List<T> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return take((size - n).coerceAtLeast(0))\n}\n/**\n * Returns a list containing all elements except last [n] elements.\n * @throws IllegalArgumentExcepcion if [n] is negative.\n * \n */
public fun <T> Array<out T>.dropLast(n: Int): List<T> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    return take((size - n).coerceAtLeast(0))\n}
```java
{\n    list.add(item)\n    yielding = true\n} \nreturn list\n\n/**\n* Returns a list containing all elements except first elements that satisfy the given \[predicate\].\n* @param \[predicate\] function that takes the index of an element and the element itself as parameters and returns the result of predicate evaluation on the element.\n*/
public inline fun <T> Array<out T>.filterIndexed(predicate: (T, Int, T) -> Boolean): List<T> {
    return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n* Returns a list containing only elements matching the given \[predicate\].\n* @param \[predicate\] function that takes the index of an element and the element itself as parameters and returns the result of predicate evaluation on the element.\n* \n*/
public inline fun <T> ByteArray.filterIndexed(predicate: (T, Int, T) -> Boolean): List<T> {
    return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n* Returns a list containing only elements matching the given \[predicate\].\n* \n*/
public inline fun <T> ShortArray.filterIndexed(predicate: (T, Int, T) -> Boolean): List<T> {
    return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n* Returns a list containing only elements matching the given \[predicate\].\n* \n*/
public inline fun <T> IntArray.filterIndexed(predicate: (T, Int, T) -> Boolean): List<T> {
    return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n* Returns a list containing only elements matching the given \[predicate\].\n* \n*/
public inline fun <T> LongArray.filterIndexed(predicate: (T, Int, T) -> Boolean): List<T> {
    return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n* Returns a list containing only elements matching the given \[predicate\].\n* \n*/
public inline fun <T> FloatArray.filterIndexed(predicate: (T, Int, T) -> Boolean): List<T> {
    return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n* Returns a list containing only elements matching the given \[predicate\].\n* \n*/
public inline fun <T> DoubleArray.filterIndexed(predicate: (T, Int, T) -> Boolean): List<T> {
    return filterIndexedTo(ArrayList<T>(), predicate)\n}
```

Open Source Used In DNAC Platform Antelope 7851
filterIndexedTo(ArrayList<Short>(), predicate)\n\n/**
 * Returns a list containing only elements matching the
given [predicate].\n @param [predicate] function that takes the index of an element and the element itself\n and returns the result of predicate evaluation on the element.\n */\n\n\nprivate fun Array<out T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {
    forEachIndexed {
        index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}\n
\nprivate fun ByteArray.filterIndexedTo(destination: C, predicate: (index: Int, Byte) -> Boolean): C {
    forEachIndexed {
        index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}\n
private fun ShortArray.filterIndexedTo(destination: C, predicate: (index: Int, Short) -> Boolean): C {
    forEachIndexed {
        index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}\n
private fun IntArray.filterIndexedTo(destination: C, predicate: (index: Int, Int) -> Boolean): C {
    forEachIndexed {
        index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}\n
private fun LongArray.filterIndexedTo(destination: C, predicate: (index: Int, Long) -> Boolean): C {
    forEachIndexed {
        index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}\n
private fun DoubleArray.filterIndexedTo(destination: C, predicate: (index: Int, Double) -> Boolean): C {
    forEachIndexed {
        index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}\n
private fun BooleanArray.filterIndexedTo(destination: C, predicate: (index: Int, Boolean) -> Boolean): C {
    forEachIndexed {
        index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}\n
private fun CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {
    forEachIndexed {
        index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}
the element itself * and returns the result of predicate evaluation on the element.

```kotlin
public inline fun <C : MutableCollection<in Float>> FloatArray.filterIndexedTo(destination: C, predicate: (index: Int, Float) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}
```

```kotlin
/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself * and returns the result of
 * predicate evaluation on the element.
 */
public inline fun <C : MutableCollection<in Double>>
DoubleArray.filterIndexedTo(destination: C, predicate: (index: Int, Double) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}
```

```kotlin
/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself * and returns the result of
 * predicate evaluation on the element.
 */
public inline fun <C : MutableCollection<in Char>>
CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}
```

```kotlin
/**
 * Returns a list containing all elements that are instances of specified type parameter R.
 */
public inline fun <reified R> Array<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {
    return filterIsInstanceTo(ArrayList<R>())
}
```

```kotlin
/**
 * Returns a list containing all elements that are instances of specified type parameter R to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself * and returns the result of
 * predicate evaluation on the element.
 */
public inline fun ByteArray.filterNot(predicate: (Byte) -> Boolean): List<Byte> {
    return filterNotTo(ArrayList<Byte>(), predicate)
}
```

```kotlin
/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun ShortArray.filterNot(predicate: (Short) -> Boolean): List<Short> {
    return filterNotTo(ArrayList<Short>(), predicate)
}
```

```kotlin
/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun IntArray.filterNot(predicate: (Int) -> Boolean): List<Int> {
    return filterNotTo(ArrayList<Int>(), predicate)
}
```

```kotlin
/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun LongArray.filterNot(predicate: (Long) -> Boolean): List<Long> {
    return filterNotTo(ArrayList<Long>(), predicate)
}
```

```kotlin
/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun FloatArray.filterNot(predicate: (Float) -> Boolean): List<Float> {
    return filterNotTo(ArrayList<Float>(), predicate)
}
```

```kotlin
/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun DoubleArray.filterNot(predicate: (Double) -> Boolean): List<Double> {
    return filterNotTo(ArrayList<Double>(), predicate)
}
```

```kotlin
/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun BooleanArray.filterNot(predicate: (Boolean) -> Boolean): List<Boolean> {
    return filterNotTo(ArrayList<Boolean>(), predicate)
}
```

```kotlin
/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun CharArray.filterNot(predicate: (Char) -> Boolean): List<Char> {
    return filterNotTo(ArrayList<Char>(), predicate)
}
```
Boolean): List<Char> {
    return filterNotTo(ArrayList<Char>(), predicate)
}

/**
 * Returns a list containing all elements that are not null.
 * @sample samples.collections.Collections.Filtering
 */
public fun <T : Any> Array<out T?>.filterNullTo(): List<T> {
    return filterNotTo(ArrayList<T>(), predicate)
}

/**
 * Returns a list containing all elements that are not `null` to the given [destination].
 */
private fun <C : MutableCollection<in T>, T : Any> Array<out T?>.filterNullTo(destination: C): C {
    for (element in this) if (element != null)
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Byte>> ByteArray.filterNotTo(destination: C, predicate: (Byte) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Short>> ShortArray.filterNotTo(destination: C, predicate: (Short) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Int>> IntArray.filterNotTo(destination: C, predicate: (Int) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Long>> LongArray.filterNotTo(destination: C, predicate: (Long) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Float>> FloatArray.filterNotTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Double>> DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Boolean>> BooleanArray.filterNotTo(destination: C, predicate: (Boolean) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Char>> CharArray.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Byte>> ByteArray.filterNotTo(destination: C, predicate: (Byte) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Short>> ShortArray.filterNotTo(destination: C, predicate: (Short) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Int>> IntArray.filterNotTo(destination: C, predicate: (Int) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Long>> LongArray.filterNotTo(destination: C, predicate: (Long) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Float>> FloatArray.filterNotTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Double>> DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Boolean>> BooleanArray.filterNotTo(destination: C, predicate: (Boolean) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
public fun <C : MutableCollection<in Char>> CharArray.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}
MutableCollection<in Float>> FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <C : MutableCollection<in Float>> FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <C : MutableCollection<in Double>> DoubleArray.filterTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterTo(destination: C, predicate: (Boolean) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
public inline fun <C : MutableCollection<in Char>> CharArray.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun <T> Array<out T>.slice(indices: IntRange): List<T> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun ByteArray.slice(indices: IntRange): List<Byte> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun ShortArray.slice(indices: IntRange): List<Short> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun IntArray.slice(indices: IntRange): List<Int> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun LongArray.slice(indices: IntRange): List<Long> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun FloatArray.slice(indices: IntRange): List<Float> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun DoubleArray.slice(indices: IntRange): List<Double> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun BooleanArray.slice(indices: IntRange): List<Boolean> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun CharArray.slice(indices: IntRange): List<Char> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun <T> Array<out T>.slice(indices: Iterable<Int>): List<T> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
public fun ByteArray.slice(indices: Iterable<Int>): List<Byte> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
public fun ShortArray.slice(indices: Iterable<Int>): List<Short> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
public fun IntArray.slice(indices: Iterable<Int>): List<Int> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
public fun LongArray.slice(indices: Iterable<Int>): List<Long> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
public fun FloatArray.slice(indices: Iterable<Int>): List<Float> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
public fun DoubleArray.slice(indices: Iterable<Int>): List<Double> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
public fun BooleanArray.slice(indices: Iterable<Int>): List<Boolean> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
public fun CharArray.slice(indices: Iterable<Int>): List<Char> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}
for (index in indices) { /* list.add(get(index)) */
  } return list
}

/** Returns an array containing elements at specified [indices]. */
*public fun <T> slice(indices: Collection<Int>): Array<T> {
  val result = arrayOfNulls(this, indices.size)
  var targetIndex = 0
  for (sourceIndex in indices) {
    result[targetIndex++] = this[sourceIndex]
  }
  return result
}

/** Returns an array containing elements of this array at specified [indices]. */
public fun ByteArray.slice(indices: IntRange): ByteArray {
  if (indices.isEmpty()) return ByteArray(0)
  return copyOfRange(indices.start, indices.endInclusive + 1)
}

/** Returns an array containing elements at specified [indices]. */
*public fun <T> slice(indices: IntRange): List<T> {
  var list = ArrayList<T>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns a list containing elements at specified [indices]. */
List<CharArray>.slice(indices: Iterable<Int>): List<Char> {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<Char>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns a list containing elements at specified [indices]. */
List<BooleanArray>.slice(indices: Iterable<Int>): List<Boolean> {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<Boolean>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns a list containing elements at specified [indices]. */
List<DoubleArray>.slice(indices: Iterable<Int>): List<Double> {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<Double>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns a list containing elements at specified [indices]. */
List<FloatArray>.slice(indices: Iterable<Int>): List<Float> {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<Float>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns a list containing elements at specified [indices]. */
List<LongArray>.slice(indices: Iterable<Int>): List<Long> {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<Long>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns a list containing elements at specified [indices]. */
List<IntArray>.slice(indices: Iterable<Int>): List<Int> {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<Int>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
ByteArray.slice(indices: Collection<Int>): ByteArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<ByteArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
CharArray.slice(indices: Collection<Int>): CharArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<CharArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
BooleanArray.slice(indices: Collection<Int>): BooleanArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<BooleanArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
CharArray.slice(indices: Collection<Int>): CharArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<CharArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
DoubleArray.slice(indices: Collection<Int>): DoubleArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<DoubleArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
FloatArray.slice(indices: Collection<Int>): FloatArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<FloatArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
IntArray.slice(indices: Collection<Int>): IntArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<IntArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
ShortArray.slice(indices: Collection<Int>): ShortArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<ShortArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
ByteArray.slice(indices: Collection<Int>): ByteArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<ByteArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
CharArray.slice(indices: Collection<Int>): CharArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<CharArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
BooleanArray.slice(indices: Collection<Int>): BooleanArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<BooleanArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
DoubleArray.slice(indices: Collection<Int>): DoubleArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<DoubleArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
FloatArray.slice(indices: Collection<Int>): FloatArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<FloatArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
IntArray.slice(indices: Collection<Int>): IntArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<IntArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns an array containing elements of this array at specified [indices]. */
ShortArray.slice(indices: Collection<Int>): ShortArray {
  val size = indices.collectionSizeOrDefault(10)
  if (size == 0) return emptyList()
  val list = ArrayList<ShortArray>(size)
  for (index in indices) {
    list.add(get(index))
  }
  return list
}

/** Returns a list containing elements at specified [indices]. */
*public fun <T> slice(indices: IntRange): List<T> {
  if (indices.isEmpty()) return copyOfRange(0, 0)
  return copyOfRange(indices.start, indices.endInclusive + 1)
}

/** Returns an array containing elements at specified [indices]. */
*public fun <T> slice(indices: IntRange): List<T> {
  if (indices.isEmpty()) return ShortArray(0)
  return copyOfRange(indices.start, indices.endInclusive + 1)
}

/** Returns an array containing elements at specified [indices]. */
*public fun <T> slice(indices: IntRange): List<T> {
  if (indices.isEmpty()) return EmptyList(0)
  return copyOfRange(indices.start, indices.endInclusive + 1)
}

/** Returns an array containing elements at specified [indices]. */
*public fun <T> slice(indices: IntRange): List<T> {
  if (indices.isEmpty()) return EmptyList(0)
  return copyOfRange(indices.start, indices.endInclusive + 1)
}

*/}
in the specified [indices] range.

```kotlin
fun IntArray.sliceArray(indices: IntRange): IntArray {
    if (indices.isEmpty()) return IntArray(0)
    return copyOfRange(indices.start, indices.endInclusive + 1)
}
```

@throws IllegalArgumentException if [n] is negative.

```kotlin
fun BooleanArray.sliceArray(indices: IntRange): BooleanArray {
    if (indices.isEmpty()) return BooleanArray(0)
    return copyOfRange(indices.start, indices.endInclusive + 1)
}
```

@throws IllegalArgumentException if [n] is negative.

```kotlin
fun CharArray.sliceArray(indices: IntRange): CharArray {
    if (indices.isEmpty()) return CharArray(0)
    return copyOfRange(indices.start, indices.endInclusive + 1)
}
```

@throws IllegalArgumentException if [n] is negative.

```kotlin
fun DoubleArray.sliceArray(indices: IntRange): DoubleArray {
    if (indices.isEmpty()) return DoubleArray(0)
    return copyOfRange(indices.start, indices.endInclusive + 1)
}
```

@throws IllegalArgumentException if [n] is negative.

```kotlin
fun FloatArray.sliceArray(indices: IntRange): FloatArray {
    if (indices.isEmpty()) return FloatArray(0)
    return copyOfRange(indices.start, indices.endInclusive + 1)
}
```

@throws IllegalArgumentException if [n] is negative.

```kotlin
fun IntArray.sliceArray(indices: IntRange): IntArray {
    if (indices.isEmpty()) return IntArray(0)
    return copyOfRange(indices.start, indices.endInclusive + 1)
}
```

@throws IllegalArgumentException if [n] is negative.

```kotlin
fun LongArray.sliceArray(indices: IntRange): LongArray {
    if (indices.isEmpty()) return LongArray(0)
    return copyOfRange(indices.start, indices.endInclusive + 1)
}
```

@throws IllegalArgumentException if [n] is negative.
size) return toList() 
if (n == 1) return listOf(this[0])
var count = 0
val list = ArrayList<Double>(n)
for (item in this) { list.add(item)
if (++count == n) break
}
return list
}

/**
 * Returns a list containing last [n] elements.
 * 
 * @throws IllegalArgumentException if \[n\] is negative.
 * 
 * @sample
 */
public fun DoubleArray.takeLast(n: Int): List<Double> {
    require(n >= 0) { "Requested element count \$n\ is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    var count = 0
    val list = ArrayList<Double>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}

/**
 * Returns a list containing last [n] elements.
 * 
 * @throws IllegalArgumentException if \[n\] is negative.
 * 
 * @sample
 */
public fun FloatArray.takeLast(n: Int): List<Float> {
    require(n >= 0) { "Requested element count \$n\ is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    var count = 0
    val list = ArrayList<Float>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}

/**
 * Returns a list containing last [n] elements.
 * 
 * @throws IllegalArgumentException if \[n\] is negative.
 * 
 * @sample
 */
public fun LongArray.takeLast(n: Int): List<Long> {
    require(n >= 0) { "Requested element count \$n\ is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    var count = 0
    val list = ArrayList<Long>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}

/**
 * Returns a list containing last [n] elements.
 * 
 * @throws IllegalArgumentException if \[n\] is negative.
 * 
 * @sample
 */
public fun IntArray.takeLast(n: Int): List<Int> {
    require(n >= 0) { "Requested element count \$n\ is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    var count = 0
    val list = ArrayList<Int>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}

/**
 * Returns a list containing last [n] elements.
 * 
 * @throws IllegalArgumentException if \[n\] is negative.
 * 
 * @sample
 */
public fun ShortArray.takeLast(n: Int): List<Short> {
    require(n >= 0) { "Requested element count \$n\ is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    var count = 0
    val list = ArrayList<Short>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}

/**
 * Returns a list containing last [n] elements.
 * 
 * @throws IllegalArgumentException if \[n\] is negative.
 * 
 * @sample
 */
public fun ByteArray.takeLast(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count \$n\ is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    var count = 0
    val list = ArrayList<Byte>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}

/**
 * Returns a list containing last [n] elements.
 * 
 * @throws IllegalArgumentException if \[n\] is negative.
 * 
 * @sample
 */
public fun <T> Array<out T>.takeLast(n: Int): List<T> {
    require(n >= 0) { "Requested element count \$n\ is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<T>(n)
    for (item in this) {
        list.add(item)
        if (++count == n) break
    }
    return list
}
Returns a list containing last [n] elements.\n\n* @throws IllegalArgumentException if [n] is negative.\n* @sample samples.collections.Collections.Transformations.take\n* \n\n* \n\n\npublic fun BooleanArray.takeLast(n: Int):\n\nList<Boolean> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<Boolean>(n)\n    for (index in size - n until size)\n        list.add(this[index])\n    return list.joinToString(*\n\n* Returns a list containing last [n] elements.\n* @throws IllegalArgumentException if [n] is negative.\n* @sample samples.collections.Collections.Transformations.take\n* \n\n\npublic fun CharArray.takeLast(n: Int):\n\nList<Char> {\n    require(n >= 0) { "Requested element count $n is less than zero." }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<Char>(n)\n    for (index in size - n until size)\n        list.add(this[index])\n    return list.joinToString(*\n\n* Returns a list containing last elements satisfying the given [predicate].\n* @sample samples.collections.Migrations.Transformations.take\n* \n\n\npublic inline fun <T> Array<out T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n\n* Returns a list containing last elements satisfying the given [predicate].\n* @sample samples.collections.Migrations.Transformations.take\n* \n\n\npublic inline fun IntArray.takeLastWhile(predicate: (Int) -> Boolean): List<Int> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n\n* Returns a list containing last elements satisfying the given [predicate].\n* @sample samples.collections.Migrations.Transformations.take\n* \n\n\npublic inline fun LongArray.takeLastWhile(predicate: (Long) -> Boolean): List<Long> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n\n* Returns a list containing last elements satisfying the given [predicate].\n* @sample samples.collections.Migrations.Transformations.take\n* \n\n\npublic inline fun FloatArray.takeLastWhile(predicate: (Float) -> Boolean): List<Float> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n\n* Returns a list containing last elements satisfying the given [predicate].\n* @sample samples.collections.Migrations.Transformations.take\n* \n\n\npublic inline fun DoubleArray.takeLastWhile(predicate: (Double) -> Boolean): List<Double> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n\n* Returns a list containing last elements satisfying the given [predicate].\n* @sample samples.collections.Migrations.Transformations.take\n* \n\n\npublic inline fun BooleanArray.takeLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n\n* Returns a list containing last elements satisfying the given [predicate].\n* @sample samples.collections.Migrations.Transformations.take\n* \n\n\npublic inline fun CharArray.takeLastWhile(predicate: (Char) -> Boolean): List<Char> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return drop(index + 1)\n        }\n    }\n    return toList()\n\n* Returns a list containing last elements satisfying the given [predicate].\n* @sample samples.collections.Migrations.Transformations.take\n* \n\n\nOpen Source Used In DNAC Platform Antelope 7659
fun <T> Array<T>.reversed(): Array<T> {
    if (isEmpty()) return this
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 * @param <T>
 */
public fun <T> List<T>.reversed(): List<T> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

fun BooleanArray.reversed(): BooleanArray {
    if (isEmpty()) return emptyList()!
    val list = toMutableList()!
    list.reverse()
    return list
}

fun DoubleArray.reversed(): DoubleArray {
    if (isEmpty()) return emptyList()!
    val list = toMutableList()!
    list.reverse()
    return list
}

fun FloatArray.reversed(): FloatArray {
    if (isEmpty()) return emptyList()!
    val list = toMutableList()!
    list.reverse()
    return list
}

fun IntArray.reversed(): IntArray {
    if (isEmpty()) return emptyList()!
    val list = toMutableList()!
    list.reverse()
    return list
}

fun ShortArray.reversed(): ShortArray {
    if (isEmpty()) return emptyList()!
    val list = toMutableList()!
    list.reverse()
    return list
}

fun ByteArray.reversed(): ByteArray {
    if (isEmpty()) return emptyList()!
    val list = toMutableList()!
    list.reverse()
    return list
}

funCharArray.reversed(): List<Char> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funBooleanArray.reversed(): List<Boolean> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funDoubleArray.reversed(): List<Double> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funFloatArray.reversed(): List<Float> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funLongArray.reversed(): List<Long> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funByteArray.reversed(): List<Byte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funIntArray.reversed(): List<Int> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funShortArray.reversed(): List<Short> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funByteArray.reversed(): List<Byte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funCharArray.reversed(): List<Char> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funBooleanArray.reversed(): List<Boolean> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funDoubleArray.reversed(): List<Double> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funFloatArray.reversed(): List<Float> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funIntArray.reversed(): List<Int> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funShortArray.reversed(): List<Short> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funByteArray.reversed(): List<Byte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funCharArray.reversed(): List<Char> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funBooleanArray.reversed(): List<Boolean> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funDoubleArray.reversed(): List<Double> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funFloatArray.reversed(): List<Float> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funLongArray.reversed(): List<Long> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funByteArray.reversed(): List<Byte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funCharArray.reversed(): List<Char> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funBooleanArray.reversed(): List<Boolean> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funDoubleArray.reversed(): List<Double> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funFloatArray.reversed(): List<Float> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funLongArray.reversed(): List<Long> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funByteArray.reversed(): List<Byte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funCharArray.reversed(): List<Char> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funBooleanArray.reversed(): List<Boolean> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

funDoubleArray.reversed(): List<Double> {
Sorts elements in the array in-place according to natural sort order of the value returned by specified [selector] function. * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting. */

public fun LongArray.sortedArray(): LongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun IntArray.sortedArray(): IntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ShortArray.sortedArray(): ShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ByteArray.sortedArray(): ByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun <T : Comparable<T>> Array<T>.sortedArray(): Array<T> {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
    return toTypedArray().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun CharArray.sorted(): List<Char> {
    return toTypedArray().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun IntArray.sorted(): List<Int> {
    return toTypedArray().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ShortArray.sorted(): List<Short> {
    return toTypedArray().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ByteArray.sorted(): List<Byte> {
    return toTypedArray().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun <T : Comparable<T>> Array<out T>.sorted(): List<T> {
    return sortedArray().asList()
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
public fun IntArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
public fun FloatArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
public fun LongArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
public fun ShortArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
public fun <T : Comparable<T>> Array<out T>.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
public inline fun <T, R : Comparable<R>> Array<out T>.sortByDescending(crossinline selector: (T) -> R?): Unit {
    if (size > 1) {
        sortWith(compareByDescending(selector))
    }
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
public inline fun <T, R : Comparable<R>> Array<out T>.sortBy(crossinline selector: (T) -> R?): Unit {
    if (size > 1) {
        sortWith(compareBy(selector))
    }
}
(isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun FloatArray.sortedArray(): FloatArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun DoubleArray.sortedArray(): DoubleArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun CharArray.sortedArray(): CharArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to the specified 
 * comparator.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public <T : Comparable<T>> Array<T>.sortedArrayDescending(): Array<T> {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to the specified 
 * comparator.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun ByteArray.sortedArrayDescending(): ByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to the specified 
 * comparator.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun ShortArray.sortedArrayDescending(): ShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to the specified 
 * comparator.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun IntArray.sortedArrayDescending(): IntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to the specified 
 * comparator.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun LongArray.sortedArrayDescending(): LongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns a list of all elements sorted according to the specified 
 * selector function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public inline fun <T, R : Comparable<R>> Array<out T>.sortedBy(crossinline selector: (T) -> R?): List<T> {
    return sortedWith(compareBy(selector))
}

/**
 * Returns a list of all elements sorted according to the specified 
 * selector function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public inline fun <R : Comparable<R>> ByteArray.sortedBy(crossinline selector: (Byte) -> R?): List<Byte> {
    return sortedWith(compareBy(selector))
}

/**
 * Returns a list of all elements sorted according to the specified 
 * selector function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public inline fun <R : Comparable<R>> ShortArray.sortedBy(crossinline selector: (Short) -> R?): List<Short> {
    return sortedWith(compareBy(selector))
}

/**
 * Returns a list of all elements sorted according to the specified 
 * selector function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public inline fun <R : Comparable<R>> IntArray.sortedBy(crossinline selector: (Int) -> R?): List<Int> {
    return sortedWith(compareBy(selector))
}

/**
 * Returns a list of all elements sorted according to the specified 
 * selector function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public inline fun <R : Comparable<R>> LongArray.sortedBy(crossinline selector: (Long) -> R?): List<Long> {
    return sortedWith(compareBy(selector))
}
FloatArray.sortedBy(crossinline selector: (Float) -> R?): List<Float> {
    return sortedWith(compareBy(selector))
}  

/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */

public inline fun <R : Comparable<R>> DoubleArray.sortedBy(crossinline selector: (Double) -> R?): List<Double> {
    return sortedWith(compareBy(selector))
}  

/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */

public inline fun <R : Comparable<R>> BooleanArray.sortedBy(crossinline selector: (Boolean) -> R?): List<Boolean> {
    return sortedWith(compareBy(selector))
}  

/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */

public inline fun <R : Comparable<R>> CharArray.sortedBy(crossinline selector: (Char) -> R?): List<Char> {
    return sortedWith(compareBy(selector))
}  

/**
 * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.
 */

public inline fun <R : Comparable<R>> Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 */

public fun <T : Comparable<T>> Array<out T>.sortedDescending(): List<T> {
    return sortedWith(reverseOrder())
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun ByteArray.sortedDescending(): List<Byte> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun ShortArray.sortedDescending(): List<Short> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun IntArray.sortedDescending(): List<Int> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun LongArray.sortedDescending(): List<Long> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun FloatArray.sortedDescending(): List<Float> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun DoubleArray.sortedDescending(): List<Double> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun BooleanArray.sortedDescending(): List<Boolean> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun CharArray.sortedDescending(): List<Char> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun IntArray.sortedDescending(): List<Int> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun ByteArray.sortedDescending(): List<Byte> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun ShortArray.sortedDescending(): List<Short> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun IntArray.sortedDescending(): List<Int> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun LongArray.sortedDescending(): List<Long> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun FloatArray.sortedDescending(): List<Float> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun DoubleArray.sortedDescending(): List<Double> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun BooleanArray.sortedDescending(): List<Boolean> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun CharArray.sortedDescending(): List<Char> {
    return sortedWith(compareByDescending(selector))
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun IntArray.sortedDescending(): List<Int> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun ByteArray.sortedDescending(): List<Byte> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun ShortArray.sortedDescending(): List<Short> {
    return copyOf().apply { sort() }.reversed()
}  

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 */

public fun IntArray.sortedDescending(): List<Int> {
    return copyOf().apply { sort() }.reversed()
LongArray.sortedDescending(): List<Long> { return copyOf().apply { sort() }.reversed() } * Returns a list of all elements sorted descending according to their natural sort order.

FloatArray.sortedDescending(): List<Float> { return copyOf().apply { sort() }.reversed() } * Returns a list of all elements sorted descending according to their natural sort order.

DoubleArray.sortedDescending(): List<Double> { return copyOf().apply { sort() }.reversed() } * Returns a list of all elements sorted according to the specified [comparator]. The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.

CharArray.sortedDescending(): List<Char> { return copyOf().apply { sortWith comparator }.asList() } * Returns a list of all elements sorted according to the specified [comparator].

LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> { return toTypedArray().apply { sortWith comparator }.asList() } * Returns a list of all elements sorted according to the specified [comparator].

ByteArray.sortedWith(comparator: Comparator<in Byte>): List<Byte> { return toTypedArray().apply { sortWith comparator }.asList() } * Returns a list of all elements sorted according to the specified [comparator].

ShortArray.sortedWith(comparator: Comparator<in Short>): List<Short> { return toTypedArray().apply { sortWith comparator }.asList() } * Returns a list of all elements sorted according to the specified [comparator].

IntArray.sortedWith(comparator: Comparator<in Int>): List<Int> { return toTypedArray().apply { sortWith comparator }.asList() } * Returns a list of all elements sorted according to the specified [comparator].

LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> { return toTypedArray().apply { sortWith comparator }.asList() } * Returns a list of all elements sorted according to the specified [comparator].

ByteArray.asList(): List<Byte> { return list() } * Returns a [List] that wraps the original array.

ShortArray.asList(): List<Short> { return list() } * Returns a [List] that wraps the original array.

IntArray.asList(): List<Int> { return list() } * Returns a [List] that wraps the original array.

LongArray.asList(): List<Long> { return list() } * Returns a [List] that wraps the original array.

FloatArray.asList(): List<Float> { return list() } * Returns a [List] that wraps the original array.

DoubleArray.asList(): List<Double> { return list() } * Returns a [List] that wraps the original array.

CharArray.asList(): List<Char> { return list() } * Returns a [List] that wraps the original array.

contentDeepEquals(): Boolean { if (another.contentDeepHashCode() == this.contentDeepHashCode()) return true } * Returns `true` if the two specified arrays are _deeply_ equal to one another, i.e. contain the same number of the same elements in the same order. If two corresponding elements are nested arrays, they are also compared deeply. * If any of arrays contains itself on any nesting level the behavior is undefined. * The elements of other types are compared for equality with the [equals] function. * For floating point numbers it means that NaN == NaN is not equal to 0.0, and -0.0 == 0.0. * The Array-out T>.contentDeepEquals(other: Array<in T>): Boolean returns a hash code based on the contents of this array if it is [List].

Nested arrays are treated as lists too. * If any of arrays contains itself on any nesting level the behavior is undefined. * The Array-out T>.contentDeepHashCode(): Int returns a string representation of the contents of this array as if it is a [List].

Nested arrays are treated as lists too. * If any of arrays contains itself on any nesting level that reference is rendered as "\"[...]\"" to prevent
recursion.

```kotlin
// @sample samples.collections.Arrays.ContentOperations.contentDeepToString

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect fun <T> Array<out T>.contentDeepToString(): String

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect infix fun ByteArray.contentEquals(other: ByteArray): Boolean

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect infix fun ShortArray.contentEquals(other: ShortArray): Boolean

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect infix fun IntArray.contentEquals(other: IntArray): Boolean

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect infix fun LongArray.contentEquals(other: LongArray): Boolean

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect infix fun FloatArray.contentEquals(other: FloatArray): Boolean

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect infix fun DoubleArray.contentEquals(other: DoubleArray): Boolean

/**
  * Returns `true` if the two specified arrays are *structurally* equal to one another. i.e. contain the same number of the same elements in the same order.
  * The elements are compared for equality with the [equals][Any.equals] function.
  * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
  */
@SinceKotlin("1.1")
public expect infix fun CharArray.contentEquals(other: CharArray): Boolean

/**
  * Returns a hash code based on the contents of this array as if it is [List].
  */
@SinceKotlin("1.1")
public expect fun <T> Array<out T>.contentHashCode(): Int

/**
  * Returns a hash code based on the contents of this array as if it is [List].
  */
@SinceKotlin("1.1")
public expect fun ByteArray.contentHashCode(): Int

/**
  * Returns a hash code based on the contents of this array as if it is [List].
  */
@SinceKotlin("1.1")
public expect fun ShortArray.contentHashCode(): Int

/**
  * Returns a hash code based on the contents of this array as if it is [List].
  */
@SinceKotlin("1.1")
public expect fun IntArray.contentHashCode(): Int

/**
  * Returns a hash code based on the contents of this array as if it is [List].
  */
@SinceKotlin("1.1")
public expect fun LongArray.contentHashCode(): Int

/**
  * Returns a hash code based on the contents of this array as if it is [List].
  */
@SinceKotlin("1.1")
public expect fun FloatArray.contentHashCode(): Int

/**
  * Returns a hash code based on the contents of this array as if it is [List].
  */
@SinceKotlin("1.1")
public expect fun DoubleArray.contentHashCode(): Int

/**
  * Returns a hash code based on the contents of this array as if it is [List].
  */
@SinceKotlin("1.1")
public expect fun CharArray.contentHashCode(): Int
```
code based on the contents of this array as if it is 

```kotlin
(List)[n] *\n@SinceKotlin("1.1.1")\npublic expect fun DoubleArray.contentHashCode(): Int
```

```kotlin

\n@SinceKotlin("1.1")\npublic expect fun BooleanArray.contentHashCode(): Int
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun CharArray.contentHashCode(): Int
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun fun Array(contentOperations.contentToString

\n@SinceKotlin("1.1")\npublic expect fun <T> Array\n
晋升: T> .contentToString(): String
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun BooleanArray.contentToString(): String
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun CharArray.contentToString(): String
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray
```

```kotlin
\n@SinceKotlin("1.1")\npublic expect fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray
```
subrange into the [destination] array and returns that array.

* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

* @param destination the array to copy to.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.

* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],

* or when that index is out of the [destination] array indices range.

* @return the [destination] array.

*/

@SinceKotlin("1.3")
public expect fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] array to copy to, 0 by default.

* @param destination the array to copy to.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.

* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],

* or when that index is out of the [destination] array indices range.

* @return the [destination] array.

*/

@SinceKotlin("1.3")
public expect fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] array to copy to, 0 by default.

* @param destination the array to copy to.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.

* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],

* or when that index is out of the [destination] array indices range.

* @return the [destination] array.

*/

@SinceKotlin("1.3")
public expect fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] array to copy to, 0 by default.

* @param destination the array to copy to.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.

* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],

* or when that index is out of the [destination] array indices range.

* @return the [destination] array.

*/

@SinceKotlin("1.3")
public expect fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray
> endIndex.
    * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array
    * or when that index is out of the [destination] array indices range.

public expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray

    * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array and even specify the subrange so that it overlaps with the destination range.
    * @return the [destination] array.

@SinceKotlin("1.3")
public expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray

    * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
    * @return the [destination] array.

@SinceKotlin("1.3")
public expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray

    * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array and even specify the subrange so that it overlaps with the destination range.
    * @return the [destination] array.

@SinceKotlin("1.3")
public expect fun BooleanArray.copyOf(): BooleanArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

@SinceKotlin("1.3")
public expect fun ByteArray.copyOf(): ByteArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

@SinceKotlin("1.3")
public expect fun ShortArray.copyOf(): ShortArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

@SinceKotlin("1.3")
public expect fun IntArray.copyOf(): IntArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

@SinceKotlin("1.3")
public expect fun LongArray.copyOf(): LongArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

@SinceKotlin("1.3")
public expect fun FloatArray.copyOf(): FloatArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

@SinceKotlin("1.3")
public expect fun DoubleArray.copyOf(): DoubleArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

@SinceKotlin("1.3")
public expect fun BooleanArray.copyOf(): BooleanArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

@SinceKotlin("1.3")
public expect fun CharArray.copyOf(): CharArray

    * @sample samples.collections.Arrays.CopyOfOperations.copyOf

    * Returns new array which is a copy of the original array, resized to the given [newSize].
    * The copy is either truncated or padded at the end with zero values if necessary.
    * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
    * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
samples.collections.Arrays.copyOfOperations.resizedPrimitiveCopyOf

*\n
@public fun ByteArray.copyOf(newSize: Int): ByteArray

*\n
* The copy is either truncated or padded at the end with zero values if necessary.
* \n
* \n
* If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* \n
* If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
* \n
* \n
* @sample samples.collections.Arrays.copyOfOperations.resizedPrimitiveCopyOf

*\n
*\n
@public expect fun ShortArray.copyOf(newSize: Int): ShortArray

*\n
* The copy is either truncated or padded at the end with zero values if necessary.
* \n
* \n
* If [newSize] is less than the size of the original array, the extra elements in the copy array are filled with zero values.
* \n
* \n
* If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
* \n
* \n
* @sample samples.collections.Arrays.copyOfOperations.resizedPrimitiveCopyOf

*\n
*\n
@public expect fun IntArray.copyOf(newSize: Int): IntArray

*\n
* The copy is either truncated or padded at the end with zero values if necessary.
* \n
* \n
* If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* \n
* If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
* \n
* \n
* @sample samples.collections.Arrays.copyOfOperations.resizedPrimitiveCopyOf

*\n
*\n
@public expect fun LongArray.copyOf(newSize: Int): LongArray

*\n
* The copy is either truncated or padded at the end with zero values if necessary.
* \n
* \n
* If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* \n
* If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
* \n
* \n
* @sample samples.collections.Arrays.copyOfOperations.resizedPrimitiveCopyOf

*\n
*\n
@public expect fun FloatArray.copyOf(newSize: Int): FloatArray

*\n
* The copy is either truncated or padded at the end with zero values if necessary.
* \n
* \n
* If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* \n
* If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
* \n
* \n
* @sample samples.collections.Arrays.copyOfOperations.resizedPrimitiveCopyOf

*\n
*\n
@public expect fun DoubleArray.copyOf(newSize: Int): DoubleArray

*\n
* The copy is either truncated or padded at the end with null char (`\u0000`) values if necessary.
* \n
* \n
* If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* \n
* If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\u0000`) values.
* \n
* \n
* @sample samples.collections.Arrays.copyOfOperations.resizedPrimitiveCopyOf

*\n
*\n
@public expect fun BooleanArray.copyOf(newSize: Int): BooleanArray

*\n
* The copy is either truncated or padded at the end with null char (`\u0000`) values if necessary.
* \n
* \n
* If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* \n
* If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\u0000`) values.
* \n
* \n
* @sample samples.collections.Arrays.copyOfOperations.resizedPrimitiveCopyOf

*\n
*\n
@public expect fun CharArray.copyOf(newSize: Int): CharArray

*\n
* The copy is either truncated or padded at the end with `null` values if necessary.
* \n
* \n
* If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].

* \n
* If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.
* \n
* \n
* @sample samples.collections.Arrays.copyOfOperations.resizedCopyOf

*\n
*\n
@Suppress("NO_ACTUAL_FOR_EXPECT")
@public expect fun Array<T>.copyOf(newSize: Int): Array<T>

*\n
* Returns a new array which is a copy of the specified range of the original array.
* \n
* @param fromIndex the start of the range (inclusive), must be in `0..array.size`
* \n
* @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
* \n
* @sample samples.collections.Arrays.copyOfRange

*\n
*\n
@Suppress("NO_ACTUAL_FOR_EXPECT")
@public expect fun <T> Array<T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<T>

*\n
* Returns a new array which is a copy of the specified range of the original array.
which is a copy of the specified range of the original array.

```kotlin
public expect fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public expect fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public expect fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public expect fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public expect fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public expect fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public expect fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * 
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public expect fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive), 0 by default.
 * @param toIndex the end of the range (exclusive), size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
public expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit
```
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

@throws IllegalArgumentExcepion if [fromIndex] is greater than [toIndex].

public expect fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit

@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

@SinceKotlin("1.3")
public expect fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun <T> Array<out T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun ByteArray.fill(fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun ShortArray.fill(fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun IntArray.fill(fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun LongArray.fill(fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun FloatArray.fill(fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun DoubleArray.fill(fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun BooleanArray.fill(fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public expect fun CharArray.fill(fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public inline fun <T> Array<out T>.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun ByteArray.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun ShortArray.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun IntArray.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun LongArray.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun FloatArray.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun DoubleArray.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun BooleanArray.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun CharArray.isEmpty(): Boolean {
    return size == 0
}

@SinceKotlin("1.3")
public inline fun <T> Array<out T>.contains(element: T): Boolean {
    return index(element) >= 0
}

@SinceKotlin("1.3")
public inline fun ByteArray.contains(element: Byte): Boolean {
    return contains(element.toInt())
}

@SinceKotlin("1.3")
public inline fun ShortArray.contains(element: Short): Boolean {
    return contains(element.toInt())
}

@SinceKotlin("1.3")
public inline fun IntArray.contains(element: Int): Boolean {
    return contains(element)
}

@SinceKotlin("1.3")
public inline fun LongArray.contains(element: Long): Boolean {
    return contains(element)
}

@SinceKotlin("1.3")
public inline fun FloatArray.contains(element: Float): Boolean {
    return contains(element)
}

@SinceKotlin("1.3")
public inline fun DoubleArray.contains(element: Double): Boolean {
    return contains(element)
}

@SinceKotlin("1.3")
public inline fun BooleanArray.contains(element: Boolean): Boolean {
    return contains(element)
}

@SinceKotlin("1.3")
public inline fun CharArray.contains(element: Char): Boolean {
    return contains(element)
}
* Returns `true` if the array is not empty.

```kotlin
*\npublic inline fun <T> Array<out T>.isNotEmpty(): Boolean {
    return !isEmpty()
}

*\npublic inline fun ByteArray.isNotEmpty(): Boolean {
    return !isEmpty()
}

*\npublic inline fun ShortArray.isNotEmpty(): Boolean {
    return !isEmpty()
}

*\npublic inline fun IntArray.isNotEmpty(): Boolean {
    return !isEmpty()
}

*\npublic inline fun LongArray.isNotEmpty(): Boolean {
    return !isEmpty()
}

*\npublic inline fun FloatArray.isNotEmpty(): Boolean {
    return !isEmpty()
}

*\npublic inline fun DoubleArray.isNotEmpty(): Boolean {
    return !isEmpty()
}

*\npublic inline fun BooleanArray.isNotEmpty(): Boolean {
    return !isEmpty()
}

*\npublic inline fun CharArray.isNotEmpty(): Boolean {
    return !isEmpty()
}
```

* Returns the last valid index for the array.

```kotlin
*\nprivate val <T> Array<out T>.lastIndex: Int
    get() = size - 1
```

* Returns an array containing all elements of the original array and then the given `element`.

```kotlin
*\npublic expect operator fun <T> Array<T>.plus(element: T): Array<T>

*\npublic expect operator fun ByteArray.plus(element: Byte): ByteArray

*\npublic expect operator fun ShortArray.plus(element: Short): ShortArray

*\npublic expect operator fun IntArray.plus(element: Int): IntArray

*\npublic expect operator fun LongArray.plus(element: Long): LongArray

*\npublic expect operator fun FloatArray.plus(element: Float): FloatArray

*\npublic expect operator fun DoubleArray.plus(element: Double): DoubleArray

*\npublic expect operator fun CharArray.plus(element: Char): CharArray
```

* Returns an array containing all elements of the original array and then the given `elements`.

```kotlin
*\npublic expect operator fun <T> Array<T>.plus(elements: Collection<T>): Array<T>

*\npublic expect operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray

*\npublic expect operator fun ShortArray.plus(elements: Collection<Short>): ShortArray

*\npublic expect operator fun IntArray.plus(elements: Collection<Int>): IntArray
```

** Open Source Used In DNAC Platform Antelope 7873 **
LongArray.plus(elements: Collection<Long>): LongArray

* Returns an array containing all elements of the original array and all elements of the given [elements] collection.

** Returns an array containing all elements of the given [elements] collection.

### Open Source Used In DNAC Platform

Antelope
Returns an array of Char containing all of the elements of this generic array.

```kotlin
public fun Array<out Char>.toCharArray(): CharArray {
    return CharArray(size) { index -> this[index] }
}
```

Returns an array of Double containing all of the elements of this generic array.

```kotlin
public fun Array<out Double>.toDoubleArray(): DoubleArray {
    return DoubleArray(size) { index -> this[index] }
}
```

Returns an array of Float containing all of the elements of this generic array.

```kotlin
public fun Array<out Float>.toFloatArray(): FloatArray {
    return FloatArray(size) { index -> this[index] }
}
```

Returns an array of Int containing all of the elements of this generic array.

```kotlin
public fun Array<out Int>.toIntArray(): IntArray {
    return IntArray(size) { index -> this[index] }
}
```

Returns an array of Long containing all of the elements of this generic array.

```kotlin
public fun Array<out Long>.toLongArray(): LongArray {
    return LongArray(size) { index -> this[index] }
}
```

Returns an array of Short containing all of the elements of this generic array.

```kotlin
public expect fun ByteArray.toTypedArray(): Array<Byte>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun ShortArray.toTypedArray(): Array<Short>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun IntArray.toTypedArray(): Array<Int>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun LongArray.toTypedArray(): Array<Long>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun FloatArray.toTypedArray(): Array<Float>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun DoubleArray.toTypedArray(): Array<Double>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun BooleanArray.toTypedArray(): Array<Boolean>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun CharArray.toTypedArray(): Array<Char>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun IntArray.toTypedArray(): Array<Int>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun LongArray.toTypedArray(): Array<Long>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun FloatArray.toTypedArray(): Array<Float>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun DoubleArray.toTypedArray(): Array<Double>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun BooleanArray.toTypedArray(): Array<Boolean>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun CharArray.toTypedArray(): Array<Char>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun IntArray.toTypedArray(): Array<Int>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun LongArray.toTypedArray(): Array<Long>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun FloatArray.toTypedArray(): Array<Float>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun DoubleArray.toTypedArray(): Array<Double>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun BooleanArray.toTypedArray(): Array<Boolean>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun CharArray.toTypedArray(): Array<Char>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun IntArray.toTypedArray(): Array<Int>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun LongArray.toTypedArray(): Array<Long>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun FloatArray.toTypedArray(): Array<Float>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun DoubleArray.toTypedArray(): Array<Double>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun BooleanArray.toTypedArray(): Array<Boolean>
```

Returns a typed* object array containing all of the elements of this primitive array.

```kotlin
public expect fun CharArray.toTypedArray(): Array<Char>
```
elements of the given array. If any of two pairs would have the same key the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

```java
public inline fun <K, V> LongArray.associate(transform: (Long) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}
```

```java
public inline fun <K, V> FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}
```

```java
public inline fun <K, V> DoubleArray.associate(transform: (Double) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}
```

```java
public inline fun <K, V> BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}
```

```java
public inline fun <T, K> Array<out T>.associateBy(keySelector: (T) -> K): Map<K, T> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)
}
```

```java
public inline fun <K> ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Byte>(capacity), keySelector)
}
```

```java
public inline fun <K> CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)
}
```

```java
public inline fun <T, K> Array<out T>.associateBy(keyword: (T) -> K): Map<K, T> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, T>(capacity), keyword)
}
```

```java
public inline fun <K> ByteArray.associateBy(keyword: (Byte) -> K): Map<K, Byte> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Byte>(capacity), keyword)
}
```
ShortArray.associateBy(keySelector: (Short) -> K): Map<K, Short> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Short>(capacity), keySelector)  
    Returns a [Map] containing the elements from the given array indexed by the key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array. 
}

IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Int>(capacity), keySelector)  
    Returns a [Map] containing the elements from the given array indexed by the key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array. 
}

LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Long>(capacity), keySelector)  
    Returns a [Map] containing the elements from the given array indexed by the key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array. 
}

FloatArray.associateBy(keySelector: (Float) -> K): Map<K, Float> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Float>(capacity), keySelector)  
    Returns a [Map] containing the elements from the given array indexed by the key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array. 
}

DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Double>(capacity), keySelector)  
    Returns a [Map] containing the elements from the given array indexed by the key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array. 
}

CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)  
    Returns a [Map] containing the elements from the given array indexed by the key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array. 
}
samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

```kotlin
/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
    keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
    keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
    keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
    keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
    keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
    keySelector, valueTransform)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
    keySelector, valueTransform)
}
```

Open Source Used In DNAC Platform Antelope 7878
```
public inline fun <K, V> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, V>
    { /* val capacity = mapCapacity(size).coerceAtLeast(16) */ return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) /* Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. * The returned map preserves the entry iteration order of the original array. */ @sample

public inline fun <K, V> CharArray.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {
    /* val capacity = mapCapacity(size).coerceAtLeast(16) */ return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) /* Populates and returns the [destination] mutable map with key-value pairs, * where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. */ @sample

public inline fun <T, K, M : MutableMap<br>    in T, T'> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination /* Populates and returns the [destination] mutable map with key-value pairs, * where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. */ @sample

public inline fun <K, M : MutableMap<br>    in K, in T> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination /* Populates and returns the [destination] mutable map with key-value pairs, * where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. */ @sample

public inline fun <K, M : MutableMap<br>    in K, in Byte, T> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination /* Populates and returns the [destination] mutable map with key-value pairs, * where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. */ @sample

public inline fun <K, M : MutableMap<br>    in K, in Short, T> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination /* Populates and returns the [destination] mutable map with key-value pairs, * where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. */ @sample

public inline fun <K, M : MutableMap<br>    in K, in Int, T> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination /* Populates and returns the [destination] mutable map with key-value pairs, * where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. */ @sample

public inline fun <K, M : MutableMap<br>    in K, in Long, T> LongArray.associateByTo(destination: M, keySelector: (Long) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination /* Populates and returns the [destination] mutable map with key-value pairs, * where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. */ @sample

public inline fun <K, M : MutableMap<br>    in K, in Float, T> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination /* Populates and returns the [destination] mutable map with key-value pairs, * where key is provided by the [keySelector] function applied to each element of the given array and value is the element itself. */ If any two elements would have the same key returned by [keySelector] the last one gets added to the map. */ @sample
```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array, and value is the element itself. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo

```kotlin
public inline fun <K, M : MutableMap<in K, in Double>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}
```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given array, and value is the element itself. If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo

```kotlin
public inline fun <K, M : MutableMap<in K, in Boolean>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}
```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and and value is provided by the [valueTransform] function applied to elements of the given array, If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

```kotlin
public inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and and value is provided by the [valueTransform] function applied to elements of the given array, If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

```kotlin
public inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and and value is provided by the [valueTransform] function applied to elements of the given array, If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

```kotlin
public inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and and value is provided by the [valueTransform] function applied to elements of the given array, If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform

```kotlin
public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}
```

Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and and value is provided by
the [valueTransform] function applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. * @sample

```kotlin
public inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

public inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

public inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform: (Byte) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}
```

* @sample
inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform: (Double) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateTo(destination: M, transform: (Boolean) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <T, C : MutableCollection<in T>> Array<out T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Byte>> ByteArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Short>> ShortArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Int>> IntArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Long>> LongArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Float>>
FloatArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

MutableCollection<Double>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

CharArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

HashSet<T>: T {
    for (item in this) {
        return item
    }
    return null
}

Array<T>: T {
    for (item in this) {
        return item
    }
    return null
}

Array<Byte>: Byte {
    for (item in this) {
        return item
    }
    return null
}

Array<Short>: Short {
    for (item in this) {
        return item
    }
    return null
}

Array<Int>: Int {
    for (item in this) {
        return item
    }
    return null
}

Array<Long>: Long {
    for (item in this) {
        return item
    }
    return null
}

Array<Float>: Float {
    for (item in this) {
        return item
    }
    return null
}

Array<Double>: Double {
    for (item in this) {
        return item
    }
    return null
}

Array<Boolean>: Boolean {
    for (item in this) {
        return item
    }
    return null
}

Array<Char>: Char {
    for (item in this) {
        return item
    }
    return null
}

List<T>: T {
    for (item in this) {
        return item
    }
    return null
}

MutableList<T>: T {
    for (item in this) {
        return item
    }
    return null
}

MutableList<Byte>: Byte {
    for (item in this) {
        return item
    }
    return null
}

MutableList<Short>: Short {
    for (item in this) {
        return item
    }
    return null
}

MutableList<Int>: Int {
    for (item in this) {
        return item
    }
    return null
}

MutableList<Long>: Long {
    for (item in this) {
        return item
    }
    return null
}

MutableList<Float>: Float {
    for (item in this) {
        return item
    }
    return null
}

MutableList<Double>: Double {
    for (item in this) {
        return item
    }
    return null
}

MutableList<Boolean>: Boolean {
    for (item in this) {
        return item
    }
    return null
}

MutableList<Char>: Char {
    for (item in this) {
        return item
    }
    return null
}

Array<T>: T {
    for (item in this) {
        return item
    }
    return null
}

Array<Byte>: Byte {
    for (item in this) {
        return item
    }
    return null
}

Array<Short>: Short {
    for (item in this) {
        return item
    }
    return null
}

Array<Int>: Int {
    for (item in this) {
        return item
    }
    return null
}

Array<Long>: Long {
    for (item in this) {
        return item
    }
    return null
}

Array<Float>: Float {
    for (item in this) {
        return item
    }
    return null
}

Array<Double>: Double {
    for (item in this) {
        return item
    }
    return null
}

Array<Boolean>: Boolean {
    for (item in this) {
        return item
    }
    return null
}

Array<Char>: Char {
    for (item in this) {
        return item
    }
    return null
}
in this) list.add(item)
} return list

/**
 * Returns a [MutableList] filled with all elements of this array.
 * @return a [MutableList] filled with all elements of this array.
 */
public fun IntArray.toMutableList(): MutableList<Int> {
    val list = ArrayList<Int>(size)
    for (item in this) list.add(item)
    return list
}

/**
 * Returns a [MutableList] filled with all elements of this array.
 * @return a [MutableList] filled with all elements of this array.
 */
public fun LongArray.toMutableList(): MutableList<Long> {
    val list = ArrayList<Long>(size)
    for (item in this) list.add(item)
    return list
}

/**
 * Returns a [MutableList] filled with all elements of this array.
 * @return a [MutableList] filled with all elements of this array.
 */
public fun FloatArray.toMutableList(): MutableList<Float> {
    val list = ArrayList<Float>(size)
    for (item in this) list.add(item)
    return list
}

/**
 * Returns a [MutableList] filled with all elements of this array.
 * @return a [MutableList] filled with all elements of this array.
 */
public fun DoubleArray.toMutableList(): MutableList<Double> {
    val list = ArrayList<Double>(size)
    for (item in this) list.add(item)
    return list
}

/**
 * Returns a [MutableList] filled with all elements of this array.
 * @return a [MutableList] filled with all elements of this array.
 */
public fun BooleanArray.toMutableList(): MutableList<Boo...
samples.collections.Collections.Transformations.flatMap\n*/\n\npublic inline fun <R> ShortArray.flatMap(transform: (Short) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform).map\n}\n\nReturns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n\n@sample samples.collections.Collections.Transformations.flatMap\n\npublic inline fun <R> IntArray.flatMap(transform: (Int) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform).map\n}\n\nReturns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n\n@sample samples.collections.Collections.Transformations.flatMap\n\npublic inline fun <R> LongArray.flatMap(transform: (Long) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform).map\n}\n\nReturns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n\n@sample samples.collections.Collections.Transformations.flatMap\n\npublic inline fun <R> FloatArray.flatMap(transform: (Float) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform).map\n}\n\nReturns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n\n@sample samples.collections.Collections.Transformations.flatMap\n\npublic inline fun <R> DoubleArray.flatMap(transform: (Double) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform).map\n}\n\nReturns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n\n@sample samples.collections.Collections.Transformations.flatMap\n\npublic inline fun <R> BooleanArray.flatMap(transform: (Boolean) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform).map\n}\n\nReturns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n\n@sample samples.collections.Collections.Transformations.flatMap\n\npublic inline fun <R> CharArray.flatMap(transform: (Char) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform).map\n}\n\nReturns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n\n@sample samples.collections.Collections.Transformations.flatMap\n\npublic inline fun <R> ByteArray.flatMap(transform: (Byte) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform).map\n}\n\nReturns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n\n@sample samples.collections.Collections.Transformations.flatMap\n\npublic inline fun <R> ShortArray.flatMapTo(destination: C, transform: (Short) -> Iterable<R>): C {
    val list = transform(element)\n    destination.addAll(list)\n    return destination\n}\n\nAppends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n\n@sample samples.collections.Collections.Transformations.flatMapTo\n\npublic inline fun <R> IntArray.flatMapTo(destination: C, transform: (Int) -> Iterable<R>): C {
    val list = transform(element)\n    destination.addAll(list)\n    return destination\n}\n\nAppends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n\n@sample samples.collections.Collections.Transformations.flatMapTo\n\npublic inline fun <R> LongArray.flatMapTo(destination: C, transform: (Long) -> Iterable<R>): C {
    val list = transform(element)\n    destination.addAll(list)\n    return destination\n}\n\nAppends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n\n@sample samples.collections.Collections.Transformations.flatMapTo\n\npublic inline fun <R> FloatArray.flatMapTo(destination: C, transform: (Float) -> Iterable<R>): C {
    val list = transform(element)\n    destination.addAll(list)\n    return destination\n}\n\nAppends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n\n@sample samples.collections.Collections.Transformations.flatMapTo\n\npublic inline fun <R> DoubleArray.flatMapTo(destination: C, transform: (Double) -> Iterable<R>): C {
    val list = transform(element)\n    destination.addAll(list)\n    return destination\n}\n\nAppends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n\n@sample samples.collections.Collections.Transformations.flatMapTo
transform: (Double) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<R>> CharArray.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 */
public inline fun <T, K> Array<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 */
public inline fun <K> ByteArray.groupBy(keySelector: (Byte) -> K): Map<K, List<Byte>> {
    return groupByTo(LinkedHashMap<K, MutableList<Byte>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 */
public inline fun <K> ShortArray.groupBy(keySelector: (Short) -> K): Map<K, List<Short>> {
    return groupByTo(LinkedHashMap<K, MutableList<Short>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 */
public inline fun <K> IntArray.groupBy(keySelector: (Int) -> K): Map<K, List<Int>> {
    return groupByTo(LinkedHashMap<K, MutableList<Int>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 */
public inline fun <K> LongArray.groupBy(keySelector: (Long) -> K): Map<K, List<Long>> {
    return groupByTo(LinkedHashMap<K, MutableList<Long>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 */
public inline fun <K> FloatArray.groupBy(keySelector: (Float) -> K): Map<K, List<Float>> {
    return groupByTo(LinkedHashMap<K, MutableList<Float>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 */
public inline fun <K> DoubleArray.groupBy(keySelector: (Double) -> K): Map<K, List<Double>> {
    return groupByTo(LinkedHashMap<K, MutableList<Double>>(), keySelector)
}
original array by the key returned by the given [keySelector] function * applied to each element and returns a map where each group key is associated with a list of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the original array. The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupBy

BooleanArray.groupBy(keySelector: (Boolean) -> K): Map<K, List<Boolean>>

CharArray.groupBy(keySelector: (Char) -> K): Map<K, List<Char>>

Array<out T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>>

ByteArray.groupBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, List<V>>

ShortArray.groupBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, List<V>>

IntArray.groupBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>>

LongArray.groupBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, List<V>>

Open Source Used In DNAC Platform Antelope 7887
with a list of corresponding values.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.

The returned map preserves the entry iteration order of the keys produced from the original array.
keySelector: (Int) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Int>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M : MutableMap<Int, MutableList<Long>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Long>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M :.MutableMap<Int, MutableList<Float>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Float>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M : MutableMap<Int, MutableList<Double>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Double>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M : MutableMap<Int, MutableList<Boolean>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Boolean>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M :MutableMap<Int, MutableList<Char>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Char>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function * applied to each element and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <T, K, V, M : MutableMap<T, MutableList<V>>> Array<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
original array* by the key returned by the given [keySelector] function applied to the element* and puts to the [destination] map each group key associated with a list of corresponding values.* @return The [destination] map.* @sample samples.collections.Collections.Transformations.groupByKeysAndValues* /unpublic inline fun <K, V, M : MutableMap<K, MutableList<V>>, ShortArray, groupByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<K, MutableList<V>>, IntArray, groupByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<K, MutableList<V>>, LongArray, groupByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<K, MutableList<V>>, FloatArray, groupByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<K, MutableList<V>>, DoubleArray, groupByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<K, MutableList<V>>, BooleanArray, groupByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

public inline fun <K, V, M : MutableMap<K, MutableList<V>>, CharArray, groupByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}
val list = destination.getOrPut(key) { ArrayList<V>() } list.add(valueTransform(element)) return

/**
 * Creates a [Grouping] source from an array to be used later with one of group-and-fold
 * operations
 *
 * @param keySelector function to extract a key from each element.
 * @param * @param * @param samples.collections.Grouping.groupingByEachCount
 */
public inline fun <T, K> Array<out T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {

 override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
 override fun

 keyOf(element: T): K = keySelector(element)


function that takes the index of an element and the element itself\* and returns the result of the transform applied to the element.

```kotlin
public inline fun <R> IntArray.mapIndexedTo(transform: (index: Int, Int) -> R): List<R> {\n    return mapIndexedTo(ArrayList<R>(size), transform)\n}
```

```kotlin
public inline fun <R> LongArray.mapIndexedTo(transform: (index: Int, Long) -> R): List<R> {\n    return mapIndexedTo(ArrayList<R>(size), transform)\n}
```

```kotlin
public inline fun <R> FloatArray.mapIndexedTo(transform: (index: Int, Float) -> R): List<R> {\n    return mapIndexedTo(ArrayList<R>(size), transform)\n}
```

```kotlin
public inline fun <R> DoubleArray.mapIndexedTo(transform: (index: Int, Double) -> R): List<R> {\n    return mapIndexedTo(ArrayList<R>(size), transform)\n}
```

```kotlin
public inline fun <R> BooleanArray.mapIndexedTo(transform: (index: Int, Boolean) -> R): List<R> {\n    return mapIndexedTo(ArrayList<R>(size), transform)\n}
```

```kotlin
public inline fun <R> CharArray.mapIndexedTo(transform: (index: Int, Char) -> R): List<R> {\n    return mapIndexedTo(ArrayList<R>(size), transform)\n}
```

```kotlin
public inline fun <T, R : Any> Array<out T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> IntArray.mapIndexedTo(destination: C, transform: (index: Int, Int) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> LongArray.mapIndexedTo(destination: C, transform: (index: Int, Long) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> FloatArray.mapIndexedTo(destination: C, transform: (index: Int, Float) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> DoubleArray.mapIndexedTo(destination: C, transform: (index: Int, Double) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```

```kotlin
public inline fun <R, C : MutableCollection<in R>> CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {\n    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}
```
this
    destination.add(transform(index++, item))
} return destination

/**
 * Applies the given [transform] function to each element and its index in the original array
 * and appends the results to the given [destination].
 */
@<transform> function that takes the index of an element and the element itself and returns the result of the transform applied to the element.

IntArray.mapIndexedTo(destination: C, transform: (index: Int, Int) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

LongArray.mapIndexedTo(destination: C, transform: (index: Int, Long) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

FloatArray.mapIndexedTo(destination: C, transform: (index: Int, Float) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

DoubleArray.mapIndexedTo(destination: C, transform: (index: Int, Double) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C
{
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C
{
    var index = 0
    for (item in this)
C : MutableCollection<in R>> IntArray.mapTo(destination: C, transform: (Int) -> R): C {
    for (item in this)
        return destination
    return destination
}

/**
 * Applies the given [transform] function to each
 * element of the original array* and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C {
    for (item in this)
        return destination
    return destination
}

/**
 * Applies the given [transform] function to each
 * element of the original array* and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C {
    for (item in this)
        return destination
    return destination
}

/**
 * Applies the given [transform] function to each
 * element of the original array* and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> DoubleArray.mapTo(destination: C, transform: (Double) -> R): C {
    for (item in this)
        return destination
    return destination
}

/**
 * Applies the given [transform] function to each
 * element of the original array* and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean) -> R): C {
    for (item in this)
        return destination
    return destination
}

/**
 * Applies the given [transform] function to each
 * element of the original array* and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> CharArray.mapTo(destination: C, transform: (Char) -> R): C {
    for (item in this)
        return destination
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an
 * [IndexedValue] containing the index of that element and the element itself.*
 */
public fun <T> Array<out T>.withIndex(): Iterable<IndexedValue<T>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an
 * [IndexedValue] containing the index of that element and the element itself.*
 */
public fun ByteArray.withIndex(): Iterable<IndexedValue<Byte>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an
 * [IndexedValue] containing the index of that element and the element itself.*
 */
public fun ShortArray.withIndex(): Iterable<IndexedValue<Short>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an
 * [IndexedValue] containing the index of that element and the element itself.*
 */
public fun IntArray.withIndex(): Iterable<IndexedValue<Int>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an
 * [IndexedValue] containing the index of that element and the element itself.*
 */
public fun LongArray.withIndex(): Iterable<IndexedValue<Long>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an
 * [IndexedValue] containing the index of that element and the element itself.*
 */
public fun FloatArray.withIndex(): Iterable<IndexedValue<Float>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an
 * [IndexedValue] containing the index of that element and the element itself.*
 */
public fun DoubleArray.withIndex(): Iterable<IndexedValue<Double>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array* into an
 * [IndexedValue] containing the index of that element and the element itself.*
 */
public fun BooleanArray.withIndex(): Iterable<IndexedValue<Boolean>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun <T> Array<out T>.distinct(): List<T> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun ByteArray.distinct(): List<Byte> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun ShortArray.distinct(): List<Short> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun IntArray.distinct(): List<Int> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun LongArray.distinct(): List<Long> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun FloatArray.distinct(): List<Float> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun DoubleArray.distinct(): List<Double> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun BooleanArray.distinct(): List<Boolean> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.* The elements in the resulting list are in the same order as they were in the source array.*
 */
public fun CharArray.distinct(): List<Char> {
    return this.toMutableSet().toList()
}
only distinct elements from the given array.
The elements in the resulting list are in the same order as they were in the source array.

```kotlin
samples.collections.Collections.Transformations.distinctAndDistinctBy

public fun ShortArray.distinct(): List<Short> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun IntArray.distinct(): List<Int> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun LongArray.distinct(): List<Long> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun FloatArray.distinct(): List<Float> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun DoubleArray.distinct(): List<Double> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun BooleanArray.distinct(): List<Boolean> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun CharArray.distinct(): List<Char> {
    return this.toMutableSet().toList()
}

```
fun <K> LongArray.distinctBy(selector: (Long) -> K): List<Long> {
    val set = HashSet<K>(array)
    val list = ArrayList<Long>(array)
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

fun <K> FloatArray.distinctBy(selector: (Float) -> K): List<Float> {
    val set = HashSet<K>(array)
    val list = ArrayList<Float>(array)
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

fun <K> DoubleArray.distinctBy(selector: (Double) -> K): List<Double> {
    val set = HashSet<K>(array)
    val list = ArrayList<Double>(array)
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

fun <K> BooleanArray.distinctBy(selector: (Boolean) -> K): List<Boolean> {
    val set = HashSet<K>(array)
    val list = ArrayList<Boolean>(array)
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

fun <T> Array<out T>.intersect(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}

fun <T> ByteArray.intersect(other: Iterable<Byte>): Set<Byte> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}

fun <T> ShortArray.intersect(other: Iterable<Short>): Set<Short> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}

fun <T> IntArray.intersect(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}

fun <T> LongArray.intersect(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}

fun <T> IntArray.union(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T> LongArray.union(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T>ByteArray.union(other: Iterable<Byte>): Set<Byte> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T> ShortArray.union(other: Iterable<Short>): Set<Short> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T> IntArray.union(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T> LongArray.union(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T> ByteArray.union(other: Iterable<Byte>): Set<Byte> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T> ShortArray.union(other: Iterable<Short>): Set<Short> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T> IntArray.union(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

fun <T> LongArray.union(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
contained by both this array and the specified collection. The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.

The returned set preserves the element iteration order of the original array.
Open Source Used In DNAC Platform Antelope
Open Source Used In DNAC Platform Antelope

```kotlin
/**
 * Returns a set containing all distinct elements from both collections.
 * Those elements of the [other] collection that are unique are iterated in the end
 * in the order of the [other] collection.
 * To get a set containing all elements
 * that are contained in both collections use [intersect].
 */
public infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.
 * Those elements of the [other] collection that are unique are iterated in the end
 * in the order of the [other] collection.
 * To get a set containing all elements
 * that are contained in both collections use [intersect].
 */
public infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.
 * Those elements of the [other] collection that are unique are iterated in the end
 * in the order of the [other] collection.
 * To get a set containing all elements
 * that are contained in both collections use [intersect].
 */
public infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.
 * Those elements of the [other] collection that are unique are iterated in the end
 * in the order of the [other] collection.
 * To get a set containing all elements
 * that are contained in both collections use [intersect].
 */
public infix fun CharArray.union(other: Iterable<Char>): Set<Char> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun <T> Array<out T>.all(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun ByteArray.all(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun ShortArray.all(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun IntArray.all(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun LongArray.all(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun FloatArray.all(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun DoubleArray.all(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun BooleanArray.all(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

doubleArray
```
return size

} Returns the number of elements in this array.

 */

@kotlin.internal.InlineOnly
public inline fun DoubleArray.count(): Int {
    return size
}

/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.count(): Int {
    return size
}

/**
 * Returns the number of elements matching the given [predicate].
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.count(): Int {
    return size
}

/**
 * Returns the number of elements matching the given [predicate].
 */
public inline fun <T> Array<T>.count(predicate: (T) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 */
public inline fun ByteArray.count(predicate: (Byte) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 */
public inline fun ShortArray.count(predicate: (Short) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 */
public inline fun IntArray.count(predicate: (Int) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 */
public inline fun LongArray.count(predicate: (Long) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 */
public inline fun BooleanArray.count(predicate: (Boolean) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 */
public inline fun CharArray.count(predicate: (Char) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <T, R> Array<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
public inline fun <R> Array<out T>.fold(initial: R, operation: (acc: R, T) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
to current accumulator value and each element.

```kotlin
fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <R> ByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <R> LongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <R> FloatArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <R> DoubleArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```
Open Source Used In DNAC Platform Antelope  7902
```
an element, the element itself * and current accumulator value, and calculates the next accumulator value.

```kotlin
public inline fun <R> ByteArray.foldRightIndexed(initial: R, operation: (index: Int, Byte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to
 * each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun <R> ShortArray.foldRightIndexed(initial: R, operation: (index: Int, Short, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to
 * each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun <R> IntArray.foldRightIndexed(initial: R, operation: (index: Int, Int, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to
 * each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun <R> LongArray.foldRightIndexed(initial: R, operation: (index: Int, Long, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to
 * each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun <R> FloatArray.foldRightIndexed(initial: R, operation: (index: Int, Float, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to
 * each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun <R> DoubleArray.foldRightIndexed(initial: R, operation: (index: Int, Double, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to
 * each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun <R> BooleanArray.foldRightIndexed(initial: R, operation: (index: Int, Boolean, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to
 * each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself * and current accumulator value, and calculates the next accumulator value.
 */
public inline fun <R> CharArray.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
```

Open Source Used In DNAC Platform Antelope 7904
null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun <T : Comparable<T>> Array<T>.max(): T? {
    if (isEmpty()) return null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i] if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun ByteArray.max(): Byte? {
    if (isEmpty()) return null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i] if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun ShortArray.max(): Short? {
    if (isEmpty()) return null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i] if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun IntArray.max(): Int? {
    if (isEmpty()) return null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i] if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun LongArray.max(): Long? {
    if (isEmpty()) return null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i] if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun FloatArray.max(): Float? {
    if (isEmpty()) return null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i] if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun DoubleArray.max(): Double? {
    if (isEmpty()) return null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i] if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
public fun CharArray.max(): Char? {
    if (isEmpty()) return null
    var max = this[0] if (max.isNaN()) return max
    for (i in 1..lastIndex) {
        val e = this[i] if (max < e) max = e
    }
    return max
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
public fun <T, R : Comparable<R>> Array<T>.maxBy(selector: (T) -> R): T? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
public fun ByteArray.maxBy(selector: (Byte) -> R): Byte? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
public fun ShortArray.maxBy(selector: (Short) -> R): Short? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
public fun IntArray.maxBy(selector: (Int) -> R): Int? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
public fun LongArray.maxBy(selector: (Long) -> R): Long? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
public fun FloatArray.maxBy(selector: (Float) -> R): Float? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
public fun DoubleArray.maxBy(selector: (Double) -> R): Double? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 */
public fun CharArray.maxBy(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
samples.collections.Collections.Aggregates.maxBy

*\!/public inline fun <R : Comparable>R>\nLongArray.maxBy(selector: (Long) -> R): Long? {\n    \n    if (isEmpty()) return \n    var maxElem = this[0] \n    val lastIndex = this.lastIndex \n    if (lastIndex == 0) return maxElem \n    var varMaxValue = selector(maxElem) \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      val v = selector(e) \n      if (maxValue < v) {\n        maxValue = e \n        maxElem = e \n      } \n    } \n    return maxElem
  }

  /\n  * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
  * @sample
  * samples.collections.Collections.Aggregates.maxBy
  *
  FloatArray.maxBy(selector: (Float) -> R): Float? {\n    \n    if (isEmpty()) return \n    var maxElem = this[0] \n    val lastIndex = this.lastIndex \n    if (lastIndex == 0) return maxElem \n    var varMaxValue = selector(maxElem) \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      val v = selector(e) \n      if (maxValue < v) {\n        maxValue = e \n        maxElem = e \n      } \n    } \n    return maxElem
  }

  /\n  * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
  * @sample
  * samples.collections.Collections.Aggregates.maxBy
  *
  DoubleArray.maxBy(selector: (Double) -> R): Double? {\n    \n    if (isEmpty()) return \n    var maxElem = this[0] \n    val lastIndex = this.lastIndex \n    if (lastIndex == 0) return maxElem \n    var varMaxValue = selector(maxElem) \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      val v = selector(e) \n      if (maxValue < v) {\n        maxValue = e \n        maxElem = e \n      } \n    } \n    return maxElem
  }

  /\n  * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
  * @sample
  * samples.collections.Collections.Aggregates.maxBy
  *
  BooleanArray.maxBy(selector: (Boolean) -> R): Boolean? {\n    \n    if (isEmpty()) return \n    var maxElem = this[0] \n    val lastIndex = this.lastIndex \n    if (lastIndex == 0) return maxElem \n    var varMaxValue = selector(maxElem) \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      val v = selector(e) \n      if (maxValue < v) {\n        maxValue = e \n        maxElem = e \n      } \n    } \n    return maxElem
  }

  /\n  * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
  * @sample
  * samples.collections.Collections.Aggregates.maxBy
  *
  CharArray.maxBy(selector: (Char) -> R): Char? {\n    \n    if (isEmpty()) return \n    var maxElem = this[0] \n    val lastIndex = this.lastIndex \n    if (lastIndex == 0) return maxElem \n    var varMaxValue = selector(maxElem) \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      val v = selector(e) \n      if (maxValue < v) {\n        maxValue = e \n        maxElem = e \n      } \n    } \n    return maxElem
  }

  /\n  * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
  * @sample
  * samples.collections.Collections.Aggregates.maxBy
  *
  Array<out T>.maxWith(comparator: Comparator<in T>): T? {\n    \n    if (isEmpty()) return \n    var max = this[0] \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      if (comparator.compare(max, e) < 0) max = e \n    } \n    return max
  }

  /\n  * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
  * @public
  * Array<Byte>.maxWith(comparator: Comparator<in Byte>): Byte? {\n    \n    if (isEmpty()) return \n    var max = this[0] \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      if (comparator.compare(max, e) < 0) max = e \n    } \n    return max
  }

  /\n  * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
  * @public
  * ShortArray.maxWith(comparator: Comparator<in Short>): Short? {\n    \n    if (isEmpty()) return \n    var max = this[0] \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      if (comparator.compare(max, e) < 0) max = e \n    } \n    return max
  }

  /\n  * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
  * @public
  * LongArray.maxWith(comparator: Comparator<in Long>): Long? {\n    \n    if (isEmpty()) return \n    var max = this[0] \n    for (i in 1..lastIndex) {\n      val e = this[i] \n      if (comparator.compare(max, e) < 0) max = e \n    } \n    return max
  }

  /\n  * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
  * @public
  * FloatArray.maxWith(comparator: Comparator<in Float>): Float?
if (isEmpty()) return null
        var max = this[0]
        for (i in 1..lastIndex) {
            val e = this[i]
            if (comparator.compare(max, e) < 0) max = e
        }
        return max
    }
    return null
}

/**
 * Returns the first element having the smallest element or `null` if there are no elements.
 * 
 * If any of elements is `NaN` returns `NaN`.
 */
public fun DoubleArray.min(): Double? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun CharArray.maxWith(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element yielding the largest value according to the provided [comparator] or `null` if there are no elements.
 */
public fun ByteArray.maxWith(comparator: Comparator<in Byte>): Byte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public fun samples.collections.Collections.Aggregates.minByOrNull(selector: (T) -> R): T? {
    if (isEmpty()) return null
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(minValue, e) < 0) minValue = e
    }
    return minValue
}

@SinceKotlin("1.1")
public fun Array<out Double>.min(): Double? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

@SinceKotlin("1.1")
public fun Array<out Float>.min(): Float? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

@SinceKotlin("1.1")
public fun Array<out Long>.min(): Long? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

@SinceKotlin("1.1")
public fun Array<out Int>.min(): Int? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

@SinceKotlin("1.1")
public fun Array<out Short>.min(): Short? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

@SinceKotlin("1.1")
public fun Array<out Byte>.min(): Byte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public fun samples.collections.Collections.Aggregates.minByNull(selector: (T) -> R): T? {
    if (isEmpty()) return null
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(minValue, e) < 0) minValue = e
    }
    return minValue
}
samples.collections.Collections.Aggregates.minBy

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
public fun <R : Comparable<R>>

ByteArray.minBy(selector: (Byte) -> R): Byte? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

ShortArray.minBy(selector: (Short) -> R): Short? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

IntArray.minBy(selector: (Int) -> R): Int? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

LongArray.minBy(selector: (Long) -> R): Long? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

CharArray.minBy(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
T.minWith(comparator: Comparator<in T>): T? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or
 * ‘null’ if there are no elements.
 */
public fun ByteArray.minWith(comparator: Comparator<in Byte>): Byte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or
 * ‘null’ if there are no elements.
 */
public fun ShortArray.minWith(comparator: Comparator<in Short>): Short? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or
 * ‘null’ if there are no elements.
 */
public fun IntArray.minWith(comparator: Comparator<in Int>): Int? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or
 * ‘null’ if there are no elements.
 */
public fun LongArray.minWith(comparator: Comparator<in Long>): Long? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or
 * ‘null’ if there are no elements.
 */
public fun FloatArray.minWith(comparator: Comparator<in Float>): Float? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or
 * ‘null’ if there are no elements.
 */
public fun DoubleArray.minWith(comparator: Comparator<in Double>): Double? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or
 * ‘null’ if there are no elements.
 */
public fun BooleanArray.minWith(comparator: Comparator<in Boolean>): Boolean? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or
 * ‘null’ if there are no elements.
 */
public fun CharArray.minWith(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
isEmpty(): Returns `true` if no elements match the given [predicate].

```kotlin
public inline fun <T> Array<out T>.none(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun ByteArray.none(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun ShortArray.none(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun IntArray.none(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun LongArray.none(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun FloatArray.none(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun DoubleArray.none(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun CharArray.none(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun <S, T : S> Array<out T>.reduce(operation: (acc: S, T) -> S): S {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator: S = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
public inline fun ByteArray.reduce(operation: (acc: Byte, Byte) -> Byte): Byte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
public inline fun ShortArray.reduce(operation: (acc: Short, Short) -> Short): Short {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
public inline fun IntArray.reduce(operation: (acc: Int, Int) -> Int): Int {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
public inline fun LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
public inline fun FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
public inline fun DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
public inline fun BooleanArray.reduce(operation: (acc: Boolean, Boolean) -> Boolean): Boolean {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
public inline fun CharArray.reduce(operation: (acc: Char, Char) -> Char): Char {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```
value and each element.

```kotlin
LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

CharArray.reduce(operation: (acc: Char, Char) -> Char): Char {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

ByteArray.reduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

ShortArray.reduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): Short {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

`Open Source Used In DNAC Platform Antelope 7912`
next accumulator value.

* @param [operation] function that takes the index of an element, current accumulator value and each element with its index in the original array.

* @sample samples.collections.Collections.Aggregates.reduce

*/
public inline fun IntArray.reduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): Int {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes the index of an element, current accumulator value and each element with its index in the original array.

* @sample samples.collections.Collections.Aggregates.reduce

*/
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes the index of an element, current accumulator value and each element with its index in the original array.

* @sample samples.collections.Collections.Aggregates.reduce

*/
public inline fun FloatArray.reduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): Float {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes the index of an element, current accumulator value and each element with its index in the original array.

* @sample samples.collections.Collections.Aggregates.reduce

*/
public inline fun DoubleArray.reduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): Double {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes the index of an element, current accumulator value and each element with its index in the original array.

* @sample samples.collections.Collections.Aggregates.reduce

*/
public inline fun BooleanArray.reduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.

* Returns null if the array is empty.

* @sample samples.collections.Collections.Aggregates.reduceOrNull

*/
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <S, T : S> Array<out T>.reduceOrNull(operation: (acc: S, T) -> S): S? {
    if (isEmpty())
        return null
    var accumulator: S = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.

* Returns null if the array is empty.

* @sample samples.collections.Collections.Aggregates.reduceOrNull

*/
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun CharArray.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

*\@SinceKotlin("1.3")*\@ExperimentalStdlibApi*\public inline fun S, T : S> Array<&gt;T&gt;.reduceOrNull(operation: (acc: S, T) -&gt; S): S? { if (isEmpty()) return null
var accumulator: S = this[0]
for (index in 1..lastIndex) { var accumulator = operation(accumulator, this[index])
} return accumulator
}*\@SinceKotlin("1.3")*\@ExperimentalStdlibApi*\public inline fun S, T : S> Array<&gt;T&gt;.reduceOrNull(operation: (acc: S, T) -&gt; S): S? { if (isEmpty()) return null
var accumulator: S = this[0]
for (index in 1..lastIndex) { var accumulator = operation(accumulator, this[index])
} return accumulator
*
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun ByteArray.reduceOrNull(operation: (acc: Byte, Byte) -> Byte): Byte? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator
 * value and each element. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun ShortArray.reduceOrNull(operation: (acc: Short, Short) -> Short): Short? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator
 * value and each element. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun IntArray.reduceOrNull(operation: (acc: Int, Int) -> Int): Int? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator
 * value and each element. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator
 * value and each element. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun FloatArray.reduceOrNull(operation: (acc: Float, Float) -> Float): Float? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator
 * value and each element. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun DoubleArray.reduceOrNull(operation: (acc: Double, Double) -> Double): Double? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator
 * value and each element. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun BooleanArray.reduceOrNull(operation: (acc: Boolean, Boolean) -> Boolean): Boolean? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator
 * value and each element. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun CharArray.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator
 * value and each element. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <S, T : S> Array<out T>.reduceRight(operation: (T, acc: S) -> S): S {
    var index = lastIndex
    if (index <
0) throw UnsupportedOperationException("Empty array can't be reduced.
} } return accumulator
/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun ByteArray.reduceRight(operation: (Byte, acc: Byte) -> Byte): Byte {
    var index = lastIndex
    if (index < 0) throw
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun ShortArray.reduceRight(operation: (Short, acc: Short) -> Short): Short {
    var index = lastIndex
    if (index < 0) throw
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun IntArray.reduceRight(operation: (Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) throw
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun LongArray.reduceRight(operation: (Long, acc: Long) -> Long): Long {
    var index = lastIndex
    if (index < 0) throw
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun FloatArray.reduceRight(operation: (Float, acc: Float) -> Float): Float {
    var index = lastIndex
    if (index < 0) throw
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun DoubleArray.reduceRight(operation: (Double, acc: Double) -> Double): Double {
    var index = lastIndex
    if (index < 0) throw
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean {
    var index = lastIndex
    if (index < 0) throw
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun CharArray.reduceRight(operation: (Char, acc: Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
index in the original array and current accumulator value.\n  * @param [operation] function that takes the index of an element, the element itself\n  * and current accumulator value, and calculates the next accumulator value.\n  * @sample samples.collections.Collections.Aggregates.reduceRight\n  */
public inline fun <S, T : S> Array<out
T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.\n")\n    var accumulator: S = get(index--)\n    while (index >= 0) {\n        \n        \n        }\n        return accumulator\n    }\n\n    @sample samples.collections.Collections.Aggregates.reduceRight\n    */
public inline fun ByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.\n")\n    var \n        \n        }\n        return accumulator\n    }\n\n    @sample samples.collections.Collections.Aggregates.reduceRight\n    */
public inline fun ShortArray.reduceRightIndexed(operation: (index: Int, Short, acc: Short) -> Short): Short {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.\n")\n    var \n        \n        }\n        return accumulator\n    }\n\n    @sample samples.collections.Collections.Aggregates.reduceRight\n    */
public inline fun IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.\n")\n    var \n        \n        }\n        return accumulator\n    }\n\n    @sample samples.collections.Collections.Aggregates.reduceRight\n    */
public inline fun LongArray.reduceRightIndexed(operation: (index: Int, Long, acc: Long) -> Long): Long {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.\n")\n    var \n        \n        }\n        return accumulator\n    }\n\n    @sample samples.collections.Collections.Aggregates.reduceRight\n    */
public inline fun FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.\n")\n    var \n        \n        }\n        return accumulator\n    }\n\n    @sample samples.collections.Collections.Aggregates.reduceRight
samples.collections.Aggregates.reduceRight

DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double

var index = lastIndex
if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
var accumulator = get(index--)
while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
}

return accumulator

/**
 * Accumulates value starting with last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRight
 */
public inline fun samples.collections.Aggregates.reduceRight

BooleanArray.reduceRightIndexed(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean

var index = lastIndex
if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
var accumulator = get(index--)
while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
}

return accumulator

/**
 * Accumulates value starting with last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRight
 */
public inline fun samples.collections.Aggregates.reduceRight

CharArray.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char

var index = lastIndex
if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
var accumulator = get(index--)
while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    --index
}

return accumulator

/**
 * Accumulates value starting with last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRight
 */
public inline fun samples.collections.Aggregates.reduceRight

ByteArray.reduceRightOrNull(operation: (Byte, acc: Byte) -> Byte): Byte?

var index = lastIndex
if (index < 0) return null
var accumulator = get(index--)
while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
}

return accumulator

/**
 * Accumulates value starting with last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRight
 */
public inline fun samples.collections.Aggregates.reduceRight

ShortArray.reduceRightOrNull(operation: (Short, acc: Short) -> Short): Short?

var index = lastIndex
if (index < 0) return null
var accumulator = get(index--)
while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
}

return accumulator

/**
 * Accumulates value starting with last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRight
 */
public inline fun samples.collections.Aggregates.reduceRight

IntArray.reduceRightOrNull(operation: (Int, acc: Int) -> Int): Int?

var index = lastIndex
if (index < 0) return null
var accumulator = get(index--)
while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
}

return accumulator

/**
 * Accumulates value starting with last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRight
 */
public inline fun samples.collections.Aggregates.reduceRight

LongArray.reduceRightOrNull(operation:
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left to
each element and current accumulator value. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun FloatArray.reduceRightOrNull(operation: 
    (Float, acc: Float) -> Float): Float? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left to
each element and current accumulator value. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun DoubleArray.reduceRightOrNull(operation: 
    (Double, acc: Double) -> Double): Double? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left to
each element and current accumulator value. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun BooleanArray.reduceRightOrNull(operation: 
    (Boolean, acc: Boolean) -> Boolean): Boolean? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left to
each element and current accumulator value. Returns null if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun CharArray.reduceRightOrNull(operation: 
    (Char, acc: Char) -> Char): Char? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right to each element and current accumulator value that starts with
 * [initial] value.
 * Note that [acc] value passed to [operation] function should not be mutated;
 * otherwise it would affect the
 * previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an
 * element, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <T, R> Array<out T>.scan(initial: R, operation: 
    (acc: R, T) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right to each element and current accumulator value that starts with
 * [initial] value.
 * Note that [acc] value passed to [operation] function should not be mutated;
 * otherwise it would affect the
 * previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an
 * element, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <R> ByteArray.scan(initial: R, operation: 
    (acc: R, Byte) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}
[operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@io.kotlin.internal.InlineOnly
public inline fun <R> ShortArray.scan(initial: R, operation: (acc: R, Short) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@io.kotlin.internal.InlineOnly
public inline fun <R> IntArray.scan(initial: R, operation: (acc: R, Int) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@io.kotlin.internal.InlineOnly
public inline fun <R> LongArray.scan(initial: R, operation: (acc: R, Long) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@io.kotlin.internal.InlineOnly
public inline fun <R> FloatArray.scan(initial: R, operation: (acc: R, Float) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@io.kotlin.internal.InlineOnly
public inline fun <R> DoubleArray.scan(initial: R, operation: (acc: R, Double) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}
BooleanArray.scan(initial: R, operation: (acc: R, Boolean) -> R): List<R> {
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

CharArray.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

Array<out T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

ByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

ShortArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
run

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <R>
IntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>({size + 1}).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right to each element, its index in the original array and current accumulator value that
 * starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <R>
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>({size + 1}).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right to each element, its index in the original array and current accumulator value that
 * starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <R>
FloatArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>({size + 1}).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right to each element, its index in the original array and current accumulator value that
 * starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <R>
DoubleArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>({size + 1}).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right to each element, its index in the original array and current accumulator value that
 * starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <R>
BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>({size + 1}).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

Open Source Used In DNAC Platform Antelope 7921
function that takes the index of an element, current accumulator value and the element itself, and calculates the
next accumulator value. * `n * n * @sample samples.collections.Collections.Aggregates.scan

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <R>
 CharArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {
    var accumulator = initial;
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right
* to each element and current accumulator value that starts with the first element of this array. * `n * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would
affect the previous value in resulting list. * `n * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value. * `n * @sample samples.collections.Collections.Aggregates.scanReduce

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <L, T : L> Array<out T>.scanReduce(operation: (acc: L, T) -> L): List<L> {
    var accumulator = this[0];
    val result = ArrayList<L>(size); add(accumulator)
    result.add(accumulator)
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right
* to each element and current accumulator value that starts with the first element of this array. * `n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value. * `n * @sample samples.collections.Collections.Aggregates.scanReduce

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun 
 ByteArray.scanReduce(operation: (acc: Byte, Byte) -> Byte): List<Byte> {
    var accumulator = this[0];
    val result = ArrayList<Byte>(size); add(accumulator)
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right
* to each element and current accumulator value that starts with the first element of this array. * `n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value. * `n * @sample samples.collections.Collections.Aggregates.scanReduce

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun 
 ShortArray.scanReduce(operation: (acc: Short, Short) -> Short): List<Short> {
    var accumulator = this[0];
    val result = ArrayList<Short>(size); add(accumulator)
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right
* to each element and current accumulator value that starts with the first element of this array. * `n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value. * `n * @sample samples.collections.Collections.Aggregates.scanReduce

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun 
 IntArray.scanReduce(operation: (acc: Int, Int) -> Int): List<Int> {
    var accumulator = this[0];
    val result = ArrayList<Int>(size); add(accumulator)
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right
* to each element and current accumulator value that starts with the first element of this array. * `n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value. * `n * @sample samples.collections.Collections.Aggregates.scanReduce

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun 
 LongArray.scanReduce(operation: (acc: Long, Long) -> Long): List<Long> {
    var accumulator = this[0];
    val result = ArrayList<Long>(size); add(accumulator)
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

Returns a list containing successive accumulation values generated by applying [operation] from left to right
* to each element and current accumulator value that starts with the first element of this array. * `n *
Returns a list containing successive accumulation values generated by applying [operation] from left to right on each element and current accumulator value that starts with the first element of this array.

Returns a list containing successive accumulation values generated by applying [operation] from left to right on each element and current accumulator value that starts with the first element of this array.

Returns a list containing successive accumulation values generated by applying [operation] from left to right on each element and current accumulator value that starts with the first element of this array.

Returns a list containing successive accumulation values generated by applying [operation] from left to right on each element and current accumulator value that starts with the first element of this array.

Returns a list containing successive accumulation values generated by applying [operation] from left to right on each element and current accumulator value that starts with the first element of this array.

Returns a list containing successive accumulation values generated by applying [operation] from left to right on each element and current accumulator value that starts with the first element of this array.

Returns a list containing successive accumulation values generated by applying [operation] from left to right on each element and current accumulator value that starts with the first element of this array.

Returns a list containing successive accumulation values generated by applying [operation] from left to right on each element and current accumulator value that starts with the first element of this array.
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun ByteArray.scanReduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): List<Byte> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Byte>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size)
        result.add(operation(index, accumulator, this[index])
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.scanReduce
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun ShortArray.scanReduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): List<Short> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Short>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size)
        result.add(operation(index, accumulator, this[index])
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.scanReduce
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun IntArray.scanReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Int>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size)
        result.add(operation(index, accumulator, this[index])
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.scanReduce
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Long>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size)
        result.add(operation(index, accumulator, this[index])
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.scanReduce
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun FloatArray.scanReduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): List<Float> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Float>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size)
        result.add(operation(index, accumulator, this[index])
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.scanReduce
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun DoubleArray.scanReduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): List<Double> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Double>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size)
        result.add(operation(index, accumulator, this[index])
    return result
}
add(accumulator) } for (index in 1 until size) { accumulator = operation(index, accumulator, this[index])
result.add(accumulator) } return result

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulation value that starts with the first element
 * of this array. Requires that the index of an element, current accumulation value and the element itself, and calculates
 * the next accumulation value.
 */
@SinceKotlin("1.3") @ExperimentalStdlibApi
public inline fun BooleanArray.scanReduceIndexed(operation: (Int, acc: Boolean, Boolean) -> Boolean): List<Boolean> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Boolean>(size).apply {
        add(accumulator) } for (index in 1 until size) { accumulator = operation(index, accumulator, this[index])
result.add(accumulator) } return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulation value that starts with the first element
 * of this array. Requires that the index of an element, current accumulation value and the element itself, and calculates
 * the next accumulation value.
 */
@SinceKotlin("1.3") @ExperimentalStdlibApi
public inline fun CharArray.scanReduceIndexed(operation: (Int, acc: Char, Char) -> Char): List<Char> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Char>(size).apply {
        add(accumulator) } for (index in 1 until size) { accumulator = operation(index, accumulator, this[index])
result.add(accumulator) } return result
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun <T> Array<out T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ByteArray.sumBy(selector: (Byte) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ShortArray.sumBy(selector: (Short) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun IntArray.sumBy(selector: (Int) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun LongArray.sumBy(selector: (Long) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun FloatArray.sumBy(selector: (Float) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun DoubleArray.sumBy(selector: (Double) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun BooleanArray.sumBy(selector: (Boolean) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun CharArray.sumBy(selector: (Char) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun <T> Array<out T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ByteArray.sumByDouble(selector: (Byte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ShortArray.sumByDouble(selector: (Short) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun IntArray.sumByDouble(selector: (Int) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun LongArray.sumByDouble(selector: (Long) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun FloatArray.sumByDouble(selector: (Float) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun DoubleArray.sumByDouble(selector: (Double) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
[selector] function applied to each element in the array.

```kotlin
public inline fun ShortArray.sumByDouble(selector: (Short) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

public inline fun IntArray.sumByDouble(selector: (Int) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

public inline fun LongArray.sumByDouble(selector: (Long) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

public inline fun FloatArray.sumByDouble(selector: (Float) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

public inline fun DoubleArray.sumByDouble(selector: (Double) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

public inline fun BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

public inline fun CharArray.sumByDouble(selector: (Char) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

public fun <T : Any> Array<T?>.requireNoNulls(): Array<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as Array<T>
}

public inline fun <T> Array<out T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {
    val first = ArrayList<T>()
    val second = ArrayList<T>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

public inline fun ByteArray.partition(predicate: (Byte) -> Boolean): Pair<List<Byte>, List<Byte>> {
    val first = ArrayList<Byte>()
    val second = ArrayList<Byte>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

public inline fun ShortArray.partition(predicate: (Short) -> Boolean): Pair<List<Short>, List<Short>> {
    val first = ArrayList<Short>()
    val second = ArrayList<Short>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

public inline fun IntArray.partition(predicate: (Int) -> Boolean): Pair<List<Int>, List<Int>> {
    val first = ArrayList<Int>()
    val second = ArrayList<Int>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

Open Source Used In DNAC Platform Antelope 7926
*second* list contains elements for which [predicate] yielded `false`.

```kotlin
public inline fun LongArray.partition(predicate: (Long) -> Boolean): Pair<List<Long>, List<Long>> {
    val first = ArrayList<Long>()
    val second = ArrayList<Long>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

```kotlin
public inline fun FloatArray.partition(predicate: (Float) -> Boolean): Pair<List<Float>, List<Float>> {
    val first = ArrayList<Float>()
    val second = ArrayList<Float>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

```kotlin
public inline fun DoubleArray.partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>> {
    val first = ArrayList<Double>()
    val second = ArrayList<Double>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

```kotlin
public inline fun BooleanArray.partition(predicate: (Boolean) -> Boolean): Pair<List<Boolean>, List<Boolean>> {
    val first = ArrayList<Boolean>()
    val second = ArrayList<Boolean>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

```kotlin
public inline fun CharArray.partition(predicate: (Char) -> Boolean): Pair<List<Char>, List<Char>> {
    val first = ArrayList<Char>()
    val second = ArrayList<Char>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

```kotlin
public infix fun <T, R> Array<out T>.zip(other: Array<out R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}
```

```kotlin
public infix fun <R> ByteArray.zip(other: Array<out R>): List<Pair<Byte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}
```

```kotlin
public infix fun <R> ShortArray.zip(other: Array<out R>): List<Pair<Short, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}
```

```kotlin
public infix fun <R> IntArray.zip(other: Array<out R>): List<Pair<Int, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}
```

```kotlin
public infix fun <R> LongArray.zip(other: Array<out R>): List<Pair<Long, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}
```
collection.

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable

```kotlin
public infix fun <R>
    FloatArray.zip(other: Array<out R>): List<Pair<Float, R>>
    { return zip(other) { t1, t2 -> t1 to t2 } }
}
```

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable

```kotlin
public infix fun <R>
    DoubleArray.zip(other: Array<out R>): List<Pair<Double, R>>
    { return zip(other) { t1, t2 -> t1 to t2 } }
}
```

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
* The returned list has length of the shortest collection.
* @sample samples.collections.Iterables.Operations.zipIterable

```kotlin
public infix fun <R>
    BooleanArray.zip(other: Array<out R>): List<Pair<Boolean, R>>
    { return zip(other) { t1, t2 -> t1 to t2 } }
}
```

Returns a list of values built from the elements of `this` array and the [other] array with the same index.
* Using the provided [transform] function applied to each pair of elements.

```kotlin
public inline fun <T, R, V>
    Array<out T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V>
    { val size = minOf(size, other.size)
      val list = ArrayList<V>(size)
      for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
      }
      return list
}
```

Returns a list of values built from the elements of `this` array and the [other] array with the same index.
* Using the provided [transform] function applied to each pair of elements.

```kotlin
public inline fun <R, V>
    ByteArray.zip(other: Array<out R>, transform: (a: Byte, b: R) -> V): List<V>
    { val size = minOf(size, other.size)
      val list = ArrayList<V>(size)
      for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
      }
      return list
}
```

Returns a list of values built from the elements of `this` array and the [other] array with the same index.
* Using the provided [transform] function applied to each pair of elements.

```kotlin
public inline fun <R, V>
    ShortArray.zip(other: Array<out R>, transform: (a: Short, b: R) -> V): List<V>
    { val size = minOf(size, other.size)
      val list = ArrayList<V>(size)
      for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
      }
      return list
}
```

Returns a list of values built from the elements of `this` array and the [other] array with the same index.
* Using the provided [transform] function applied to each pair of elements.

```kotlin
public inline fun <R, V>
    IntArray.zip(other: Array<out R>, transform: (a: Int, b: R) -> V): List<V>
    { val size = minOf(size, other.size)
      val list = ArrayList<V>(size)
      for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
      }
      return list
}
```

Returns a list of values built from the elements of `this` array and the [other] array with the same index.
* Using the provided [transform] function applied to each pair of elements.

```kotlin
public inline fun <R, V>
    LongArray.zip(other: Array<out R>, transform: (a: Long, b: R) -> V): List<V>
    { val size = minOf(size, other.size)
      val list = ArrayList<V>(size)
      for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
      }
      return list
}
```

Returns a list of values built from the elements of `this` array and the [other] array with the same index.
* Using the provided [transform] function applied to each pair of elements.

```kotlin
public inline fun <R, V>
    FloatArray.zip(other: Array<out R>, transform: (a: Float, b: R) -> V): List<V>
    { val size = minOf(size, other.size)
      val list = ArrayList<V>(size)
      for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
      }
      return list
}
```

Returns a list of values built from the elements of `this` array and the [other] array with the same index.
* Using the provided [transform] function applied to each pair of elements.

* @sample samples.collections.Iterables.Operations.zipIterable

```kotlin
public infix fun <R>
    CharArray.zip(other: Array<out R>): List<Pair<Char, R>>
    { return zip(other) { t1, t2 -> t1 to t2 } }
}
```

Returns a list of values built from the elements of `this` array and the [other] array with the same index.
* Using the provided [transform] function applied to each pair of elements.

```
* @sample samples.collections.Iterables.Operations.zipIterable
```
list has length of the shortest collection.

* @sample

samples.collections.Iterables.Operations.zipIterableWithTransform

*/

public inline fun <R, V>

DoubleArray.zip(other: Array<R>, transform: (a: Double, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

*/

public inline fun <R, V>

BooleanArray.zip(other: Array<R>, transform: (a: Boolean, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

*/

public inline fun <R, V>

CharArray.zip(other: Array<R>, transform: (a: Char, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

*/

public infix fun <T, R> Array<T>.zip(other: Iterable<R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/

public infix fun <R>

ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/

public infix fun <R> ShortArray.zip(other: Iterable<R>): List<Pair<Short, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/

public infix fun <R> IntArray.zip(other: Iterable<R>): List<Pair<Int, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/

public infix fun <R> LongArray.zip(other: Iterable<R>): List<Pair<Long, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/

public infix fun <R> FloatArray.zip(other: Iterable<R>): List<Pair<Float, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/

public infix fun <R> DoubleArray.zip(other: Iterable<R>): List<Pair<Double, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/

public infix fun <R> BooleanArray.zip(other: Iterable<R>): List<Pair<Boolean, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/

public infix fun <R> CharArray.zip(other: Iterable<R>): List<Pair<Char, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

*/
elements of 'this' array and the [other] collection with the same index
* The returned list has length of the shortest collection.
* @sample samples.collections.Operables.Operations.zipIterableWithTransform

/**
 * Returns a list of values built from the elements of 'this' array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operables.Operations.zipIterableWithTransform
 */
public inline fun <T, R, V> Array<out T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {
    var i = 0
    var arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operables.Operations.zipIterableWithTransform
 */
public inline fun <T, R, V> IntArray.zip(other: Iterable<R>, transform: (a: Int, b: R) -> V): List<V> {
    var i = 0
    var arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operables.Operations.zipIterableWithTransform
 */
    var i = 0
    var arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operables.Operations.zipIterableWithTransform
 */
public inline fun <T, R, V> ByteArray.zip(other: Iterable<R>, transform: (a: Byte, b: R) -> V): List<V> {
    var i = 0
    var arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operables.Operations.zipIterableWithTransform
 */
public inline fun <T, R, V> ShortArray.zip(other: Iterable<R>, transform: (a: Short, b: R) -> V): List<V> {
    var i = 0
    var arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operables.Operations.zipIterableWithTransform
 */
public inline fun <T, R, V> FloatArray.zip(other: Iterable<R>, transform: (a: Float, b: R) -> V): List<V> {
    var i = 0
    var arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operables.Operations.zipIterableWithTransform
 */
public inline fun <T, R, V> DoubleArray.zip(other: Iterable<R>, transform: (a: Double, b: R) -> V): List<V> {
    var i = 0
    var arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operables.Operations.zipIterableWithTransform
 */
public inline fun <T, R, V>
BooleanArray.zip(other: Iterable<R>, transform: (a: Boolean, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    val var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <V> BooleanArray.zipIterableWithTransform(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    val var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public inline fun <R, V> CharArray.zip(other: Iterable<R>, transform: (a: Char, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public inline fun <R, V> ShortArray.zip(other: ShortArray, transform: (a: Short, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public inline fun <R, V> IntArray.zip(other: IntArray, transform: (a: Int, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public inline fun <R, V> LongArray.zip(other: LongArray, transform: (a: Long, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public inline fun <R, V> FloatArray.zip(other: FloatArray, transform: (a: Float, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.
 */
public inline fun <R, V> DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}
returns a list of values built from the elements of `this` array and the [other] array with the same index
* using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest array.

```kotlin
fun <V> IntArray.zip(other: IntArray, transform: (a: Int, b: Int) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
```

```kotlin
fun <V> LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
```

```kotlin
fun <V> FloatArray.zip(other: FloatArray, transform: (a: Float, b: Float) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
```

```kotlin
fun <V> DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
```

```kotlin
fun <V> BooleanArray.zip(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
```

```kotlin
fun <V> CharArray.zip(other: CharArray, transform: (a: Char, b: Char) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
```

```kotlin
fun <T, A : Appendable> Array<out T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            buffer.appendElement(element, transform)
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}
```
Appendable> ByteArray.joinTo(buffer: A, separator: CharSequence = ",", prefix: CharSequence = "\n", postfix: CharSequence = ",\n", limit: Int = -1, truncated: CharSequence = "...\n", transform: ((Byte) -> CharSequence)? = null): A { 
    var count = 0
    if (++count > 1) return

    var n = *n*
    if (prefix) n = *n*
    if (postfix) n = *n*
    if (limit < 0 || count <= limit) { 
        if (transform != null) return
    } else if (limit >= 0 && count > limit) return

    buffer.copy(n, 0, n, 0, buffer, separator) 

    if (count > 1) buffer.append(separator)
    buffer.append(transform(element))
    } else break

    if (transform != null) buffer.append(transform(element))
    else buffer.append(element.toString())

    buffer.append(separator)
    if (limit < 0 || count <= limit) { 
        if (transform != null) return
    } else if (limit >= 0 && count > limit) buffer.append(truncated)

    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix) 
}
separator, prefix, postfix, limit, truncated, transform).toString().

```kotlin
/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix]
 * if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to '...'...').
 */
public fun IntArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Int) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString().

```
public fun ShortArray.asIterable(): Iterable<Short> { if (isEmpty()) return emptyList() return
    this.iterator() }\n
    *\n    * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
    */

    public fun IntArray.asIterable(): Iterable<Int> { if (isEmpty()) return emptyList() return
    this.iterator() }\n
    *\n    * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
    */

    public fun LongArray.asIterable(): Iterable<Long> { if (isEmpty()) return emptyList() return
    this.iterator() }\n
    *\n    * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
    */

    public fun FloatArray.asIterable(): Iterable<Float> { if (isEmpty()) return emptyList() return
    this.iterator() }\n
    *\n    * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
    */

    public fun DoubleArray.asIterable(): Iterable<Double> { if (isEmpty()) return emptyList() return
    this.iterator() }\n
    *\n    * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
    */

    public fun BooleanArray.asIterable(): Iterable<Boolean> { if (isEmpty()) return emptyList() return
    this.iterator() }\n
    *\n    * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
    */

    public fun CharArray.asIterable(): Iterable<Char> { if (isEmpty()) return emptyList() return
    this.iterator() }\n
    *\n    * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
    */

    @kotlin.jvm.JvmName("averageOfByte")
    public fun Array<out Byte>.average(): Double { var sum:
    Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return
    if (count == 0) Double.NaN else sum / count
    }\n
    @kotlin.jvm.JvmName("averageOfShort")
    public fun Array<out Short>.average(): Double { var sum:
    Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return
    if (count == 0) Double.NaN else sum / count
    }\n
    @kotlin.jvm.JvmName("averageOfInt")
    public fun Array<out Int>.average(): Double { var sum:
    Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return
    if (count == 0) Double.NaN else sum / count
    }\n
    @kotlin.jvm.JvmName("averageOfLong")
    public fun Array<out Long>.average(): Double { var sum:
    Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return
    if (count == 0) Double.NaN else sum / count
    }\n
    @kotlin.jvm.JvmName("averageOfFloat")
    public fun Array<out Float>.average(): Double { var sum:
    Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return
    if (count == 0) Double.NaN else sum / count
    }\n
    @kotlin.jvm.JvmName("averageOfDouble")
    public fun Array<out Double>.average(): Double { var sum:
    Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return
    if (count == 0) Double.NaN else sum / count
    }\n
    @kotlin.jvm.JvmName("averageOfBoolean")
    public fun Array<out Boolean>.average(): Double { var sum:
    Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return
    if (count == 0) Double.NaN else sum / count
    }\n
    @kotlin.jvm.JvmName("averageOfChar")
    public fun Array<out Char>.average(): Double { var sum:
    Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return
    if (count == 0) Double.NaN else sum / count
    }

    // Other methods for arithmetic operations on arrays...
if (count == 0) Double.NaN else sum / count

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfInt")
public fun Array<out Int>.average(): Double {
    var sum: Int = 0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Array<out Long>.average(): Double {
    var sum: Long = 0L
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Array<out Float>.average(): Double {
    var sum: Float = 0.0f
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfDouble")
public fun Array<out Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun ByteArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun ShortArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun IntArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun LongArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun FloatArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun DoubleArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Array<out Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Array<out Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Array<out Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Array<out Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Array<out Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Array<out Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Array<out Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Array<out Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Array<out Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Array<out Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Array<out Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Array<out Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Array<out Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Array<out Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Array<out Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Array<out Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Array<out Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Array<out Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}
sum of all elements in the array. */
public fun FloatArray.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun DoubleArray.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("RangesKt")

package kotlin.ranges

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun IntRange.random(): Int {
    return random(Random)
}

/**
 * Returns a random element from this range using the specified source of randomness.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
public fun IntRange.random(random: Random): Int {
    try {
        return random.nextInt(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.
 */
@SinceKotlin("1.3")
public fun IntRange.randomOrNull(): Int? {
    if (isEmpty())
        return null
    return random.nextInt(this)
}

/**
 * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.
 */
@SinceKotlin("1.3")
public fun LongRange.random(): Long {
    try {
        return random.nextLong(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.
 */
@SinceKotlin("1.3")
public fun LongRange.randomOrNull(): Long? {
    if (isEmpty())
        return null
    return random.nextLong(this)
}

/**
 * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.
 */
@SinceKotlin("1.3")
public fun CharRange.random(): Char {
    try {
        return random.nextInt(first.toInt(), last.toInt() + 1).toChar()
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.
 */
@SinceKotlin("1.3")
public fun CharRange.randomOrNull(): Char? {
    if (isEmpty())
        return null
    return random.nextInt(first.toInt(), last.toInt() + 1).toChar()
}

/**
 * Returns `true` if this range contains the specified [element].
 * 
 * Always returns `false` if the [element] is `null`.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun IntRange.contains(element: Int): Boolean {
    return contains(element)
}

@SinceKotlin("1.3")
public fun IntRange.contains(random: Random): Boolean {
    try {
        return contains(random.nextInt(this))
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

@SinceKotlin("1.3")
public fun IntRange.containsOrNull(random: Random): Boolean? {
    if (isEmpty())
        return null
    return contains(random.nextInt(this))
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun LongRange.contains(element: Long): Boolean {
    return contains(element)
}

@SinceKotlin("1.3")
public fun LongRange.contains(random: Random): Boolean {
    try {
        return contains(random.nextLong(this))
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

@SinceKotlin("1.3")
public fun LongRange.containsOrNull(random: Random): Boolean? {
    if (isEmpty())
        return null
    return contains(random.nextLong(this))
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun CharRange.contains(element: Char): Boolean {
    return contains(element)
}

@SinceKotlin("1.3")
public fun CharRange.contains(random: Random): Boolean {
    try {
        return contains(random.nextInt(first.toInt(), last.toInt() + 1).toChar())
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

@SinceKotlin("1.3")
public fun CharRange.containsOrNull(random: Random): Boolean? {
    if (isEmpty())
        return null
    return contains(random.nextInt(first.toInt(), last.toInt() + 1).toChar())
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun IntRange.containsOrEqual(element: Int): Boolean {
    return containsOrEqual(element)
}

@SinceKotlin("1.3")
public fun IntRange.containsOrEqual(random: Random): Boolean {
    try {
        return containsOrEqual(random.nextInt(this))
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

@SinceKotlin("1.3")
public fun IntRange.containsOrEqualOrNull(random: Random): Boolean? {
    if (isEmpty())
        return null
    return containsOrEqual(random.nextInt(this))
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun LongRange.containsOrEqual(element: Long): Boolean {
    return containsOrEqual(element)
}

@SinceKotlin("1.3")
public fun LongRange.containsOrEqual(random: Random): Boolean {
    try {
        return containsOrEqual(random.nextLong(this))
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

@SinceKotlin("1.3")
public fun LongRange.containsOrEqualOrNull(random: Random): Boolean? {
    if (isEmpty())
        return null
    return containsOrEqual(random.nextLong(this))
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun CharRange.containsOrEqual(element: Char): Boolean {
    return containsOrEqual(element)
}

@SinceKotlin("1.3")
public fun CharRange.containsOrEqual(random: Random): Boolean {
    try {
        return containsOrEqual(random.nextInt(first.toInt(), last.toInt() + 1).toChar())
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

@SinceKotlin("1.3")
public fun CharRange.containsOrEqualOrNull(random: Random): Boolean? {
    if (isEmpty())
        return null
    return containsOrEqual(random.nextInt(first.toInt(), last.toInt() + 1).toChar())
}
IntRange.contains(element: Int?): Boolean {
    return element != null && contains(element)  // Returns 'true' if this range contains the specified [element] and 'false' if the [element] is 'null'.
}

/**
 * Returns `true` if this range contains the specified [element]. Always returns 'false' if the [element] is 'null'.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline operator fun LongRange.contains(element: Long?): Boolean {
    return element != null && contains(element)
}

/**
 * Returns `true` if this range contains the specified [element]. Always returns 'false' if the [element] is 'null'.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline operator fun CharRange.contains(element: Char?): Boolean {
    return element != null && contains(element)
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Byte): Boolean {
    return contains(value.toInt())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Byte): Boolean {
    return contains(value.toLong())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Byte): Boolean {
    return contains(value.toShort())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("doubleRangeContains")
public operator fun ClosedRange<Double>.contains(value: Byte): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Byte): Boolean {
    return contains(value.toFloat())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Double): Boolean {
    return contains(value.toInt())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Double): Boolean {
    return contains(value.toLong())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("byteRangeContains")
public operator fun ClosedRange<Byte>.contains(value: Double): Boolean {
    return contains(value.toByte())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Double): Boolean {
    return contains(value.toShort())
}

/**
 * Checks if the specified [value] belongs to this range. Always returns 'false' if the specified [value] is 'null'.
 */
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Double): Boolean {
    return contains(value.toFloat())
}
and is going to be removed.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Double>.contains(value: Short): Boolean {\n    return contains(value.toDouble())\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Float>.contains(value: Short): Boolean {\n    return contains(value.toFloat())\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Byte>.contains(value: Short): Boolean {\n    return contains(value.toInt())\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Char>.contains(value: Short): Boolean {\n    return contains(value)\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Int>.contains(value: Short): Boolean {\n    return contains(value.toInt())\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Long>.contains(value: Short): Boolean {\n    return contains(value.toLong())\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Byte>.contains(value: Int): Boolean {\n    return contains(value.toInt())\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Char>.contains(value: Int): Boolean {\n    return contains(value)\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Int>.contains(value: Int): Boolean {\n    return contains(value)\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Long>.contains(value: Int): Boolean {\n    return contains(value.toLong())\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Byte>.contains(value: Long): Boolean {\n    return contains(value.toLong())\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Char>.contains(value: Long): Boolean {\n    return contains(value)\n}\n
/**
 * Checks if the specified \[value\] belongs to this range.\n}@kotlin.jvm.JvmName("doubleRangeContains")\npublic operator fun
ClosedRange<Int>.contains(value: Long): Boolean {\n    return contains(value)\n}
* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Short.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1L)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Int.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Byte.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Char.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Fun.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Double.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toDouble(), to.toDouble(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Float.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toFloat(), to.toFloat(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Fun.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toDouble(), to.toDouble(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Double.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toDouble(), to.toDouble(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Float.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toFloat(), to.toFloat(), -1)
}
```

* If the [to] value is greater than `this` value the returned progression is empty.

```kotlin
public infix fun Fun.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toDouble(), to.toDouble(), -1)
}
```
Open Source Used In DNAC Platform Antelope  7943

```kotlin
Int.toShortExactOrNull(): Short? {
    return if (this in Short.MIN_VALUE.toInt()..Short.MAX_VALUE.toInt())
        this.toShort() else null
}

internal fun Long.toShortExactOrNull(): Short? {
    return if (this in Short.MIN_VALUE.toLong()..Short.MAX_VALUE.toLong())
        this.toShort() else null
}

internal fun Double.toShortExactOrNull(): Short? {
    return if (this in Short.MIN_VALUE.toDouble()..Short.MAX_VALUE.toDouble())
        this.toShort() else null
}

internal fun Float.toShortExactOrNull(): Short? {
    return if (this in Short.MIN_VALUE.toFloat()..Short.MAX_VALUE.toFloat())
        this.toShort() else null
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Int.until(to: Byte): IntRange {
    return this..(to.toInt() - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Long.until(to: Byte): LongRange {
    return this..(to.toLong() - 1).toLong()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Byte.until(to: Byte): IntRange {
    return this.toInt()..(to.toInt() - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Short.until(to: Byte): IntRange {
    return this.toInt()..(to.toInt() - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Char.until(to: Char): CharRange {
    return CharRange.EMPTY if (to <= '\u0000')
        return CharRange(return this..(to - 1).toChar())
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Int.until(to: Int): IntRange {
    if (to <= Int.MIN_VALUE) return IntRange.EMPTY
    return this..(to - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * If the [to] value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Long.until(to: Int): LongRange {
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this..(to - 1).toLong()
}
```

- Antelope
- DNAC
- Open Source
1) toLong() Returns a range from this value up to but excluding the specified [to] value. If the [to] value is less than or equal to 'this' value, then the returned range is empty. 

public infix fun Byte.until(to: Short): IntRange {
    return this.toInt(). until(to.toInt() - 1)
}

/**
 * Returns a range from this value up to but excluding the specified [to] value. If the [to] value is less than or equal to 'this' value, then the returned range is empty. 

 * @sample samples.comparisons.ComparableOps.coerceAtLeastComparable *\n * @sample samples.comparisons.ComparableOps.coerceAtMostComparable
 */
public fun Byte.coerceAtLeast(maximumValue: Byte): Byte {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue]. If the [maximumValue] is less than or equal to the [maximumValue] or the [maximumValue] otherwise.

 * @sample samples.comparisons.ComparableOps.coerceAtMost
 */
public fun Byte.coerceAtMost(maximumValue: Byte): Byte {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue]. If the [maximumValue] is less than or equal to the [maximumValue] or the [maximumValue] otherwise.

 * @sample samples.comparisons.ComparableOps.coerceAtMost
 */
public fun Short.coerceAtMost(maximumValue: Short): Short {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue]. If the [maximumValue] is less than or equal to the [maximumValue] or the [maximumValue] otherwise.

 * @sample samples.comparisons.ComparableOps.coerceAtMost
 */
public fun Int.coerceAtMost(maximumValue: Int): Int {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue]. If the [maximumValue] is less than or equal to the [maximumValue] or the [maximumValue] otherwise.

 * @sample samples.comparisons.ComparableOps.coerceAtMost
 */
public fun Long.coerceAtMost(maximumValue: Long): Long {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue]. If the [maximumValue] is less than or equal to the [maximumValue] or the [maximumValue] otherwise.

 * @sample samples.comparisons.ComparableOps.coerceAtMost
 */
public fun Float.coerceAtMost(maximumValue: Float): Float {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue]. If the [maximumValue] is less than or equal to the [maximumValue] or the [maximumValue] otherwise.

 * @sample samples.comparisons.ComparableOps.coerceAtMost
 */
public fun Double.coerceAtMost(maximumValue: Double): Double {
    return if (this > maximumValue) maximumValue else this
}
the [maximumValue] otherwise.

```kotlin
public fun Long.coerceAtMost(maximumValue: Long): Long { return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue].
 * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.
 */
public fun <T : Comparable<T>> T.coerceIn(minimumValue: T?, maximumValue: T?): T {
    if (minimumValue !== null && maximumValue !== null) {
        if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
        if (this < minimumValue) return minimumValue
        if (this > maximumValue) return maximumValue
    }
    else {
        if (minimumValue !== null && this < minimumValue) return minimumValue
        if (maximumValue !== null && this > maximumValue) return maximumValue
    }
    return this
}

/**
 * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].
 * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Byte.coerceIn(minimumValue: Byte, maximumValue: Byte): Byte {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}

/**
 * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].
 * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Short.coerceIn(minimumValue: Short, maximumValue: Short): Short {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}

/**
 * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].
 * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Int.coerceIn(minimumValue: Int, maximumValue: Int): Int { return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].
 * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Long.coerceIn(minimumValue: Long, maximumValue: Long): Long { return if (this > maximumValue) maximumValue else this
}
```

Open Source Used In DNAC Platform Antelope 7945
is less than \([minimumValue]\), or \([maximumValue]\) if this value is greater than \([maximumValue]\).

```java
public fun Float.coerceIn(minimumValue: Float,
maximumValue: Float): Float {
    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// Auto-generated file. DO NOT EDIT!

package kotlin

import kotlin.experimental.*

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@ExperimentalUnsignedTypes
public inline class UByte @PublishedApi internal
constructor(@PublishedApi internal val data: Byte) : Comparable<UByte> {

    companion object {
        /**
         * A constant holding the minimum value an instance of UByte can have.
         */
        public const val
```
MIN_VALUE: UByte = UByte(0)
/**
 * A constant holding the maximum value an instance of UByte can
 * have.
 */
public const val MAX_VALUE: UByte = UByte(-1)
/**
 * The number of bytes used to represent an instance of UByte in a binary
 * form.
 */
public const val SIZE_BYTES: Int = 1
/**
 * The number of bits used to represent an instance of UByte in a binary
 * form.
 */
public const val SIZE_BITS: Int = 8

/**
 * Compares this value with the specified value for order.
 * Returns zero if this value is equal to the specified other value, a negative number if it's
 * less than other, or a positive number if it's greater than other.
 */
public override inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())

/**
 * Compares this value with the specified value for order.
 * Returns zero if this value is equal to the specified other value, a negative number if it's
 * less than other, or a positive number if it's greater than other.
 */
public inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())

/**
 * Compares this value with the specified value for order.
 * Returns zero if this value is equal to the specified other value, a negative number if it's
 * less than other, or a positive number if it's greater than other.
 */
public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)

/**
 * Compares this value with the specified value for order.
 * Returns zero if this value is equal to the specified other value, a negative number if it's
 * less than other, or a positive number if it's greater than other.
 */
public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)

/** Adds the other value to this value.
 */
public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())

/** Adds the other value to this value.
 */
public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())

/** Adds the other value to this value.
 */
public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)

/** Adds the other value to this value.
 */
public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)

/** Subtracts the other value from this value.
 */
public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())

/** Subtracts the other value from this value.
 */
public inline operator fun minus(other: UShort): UInt = this.toUInt().minus(other.toUInt())

/** Subtracts the other value from this value.
 */
public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)

/** Subtracts the other value from this value.
 */
public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)

/** Multiplies this value by the other value.
 */
public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt())

/** Multiplies this value by the other value.
 */
public inline operator fun times(other: UShort): UInt = this.toUInt().times(other.toUInt())

/** Multiplies this value by the other value.
 */
public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)

/** Multiplies this value by the other value.
 */
public inline operator fun times(other: ULong): ULong = this.toULong().times(other)

/** Calculates the remainder of dividing this value by the other value.
 */
public inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other.toUInt())

/** Calculates the remainder of dividing this value by the other value.
 */
public inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())
binary representation as this `Byte` value."
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun Byte.toUByte(): UByte = UByte\(this\)\n/**\n* Converts this [Short] value to [UByte].\n*\n* If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n* the same numerical value as this `Short` value.\n*\n* The resulting `UByte` value is represented by the least significant 8 bits of this `Short` value.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun Short.toUByte(): UByte = UByte\(this\)\n/**\n* Converts this [Int] value to [UByte].\n*\n* If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n* the same numerical value as this `Int` value.\n*\n* The resulting `UByte` value is represented by the least significant 8 bits of this `Int` value.\n*\n* The resulting `UByte` value is represented by the least significant 8 bits of this `Long` value.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npackage kotlin

import kotlin.experimental.*

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UInt @PublishedApi internal constructor(@PublishedApi internal val data: Int) : Comparable<UInt> {

    companion object {
        /**
         * A constant holding the minimum value an instance of UInt can have.\n         */
        public const val MIN_VALUE: UInt = UInt(0)

        /**
         * A constant holding the maximum value an instance of UInt can have.\n         */
        public const val MAX_VALUE: UInt = UInt(-1)

        /**
         * The number of bytes used to represent an instance of UInt in a binary form.\n         */
        public const val SIZE_BYTES: Int = 4

        /**
         * The number of bits used to represent an instance of UInt in a binary form.\n         */
        public const val SIZE_BITS: Int = 32
    }

    /**
     * Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if it’s less than other,\n     * or a positive number if it’s greater than other.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UInt): Int = uintCompare\(this.data, other.data\)

    /**
     * Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if it’s less than other,\n     * or a positive number if it’s greater than other.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = this\(\)\n
    /** Adds the other value to this value.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun plus\(other: UByte\): UInt = this\(\)\n
    /** Adds the other value to this value.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun plus\(other: UShort\): UInt = this\(\)\n
    /** Adds the other value to this value.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun plus\(other: UInt\): UInt = this\(\)\n
    /** Adds the other value to this value.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun plus\(other: ULong\): UInt = this\(\)\n
    /** Subtracts the other value from this value.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun minus\(other: UByte\): UInt = this\(\)\n
    /** Subtracts the other value from this value.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun minus\(other: UShort\): UInt = this\(\)\n
    /** Subtracts the other value from this value.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun minus\(other: UInt\): UInt = this\(\)\n
    /** Subtracts the other value from this value.\n     */
    @kotlin.internal.InlineOnly
    public inline operator fun minus\(other: ULong\): UInt = this\(\)\n
   ယောက်သညှိသည်။
this.minus(other.toUInt()) /* Precedes the other value from this value. */
    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UShort): UInt = this.minus(other.toUInt())
    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UInt): UInt = this.minus(other.toUInt())
    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: ULong): ULong = this.minus(other.toULong())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UByte): UInt = this.times(other.toUInt())
    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UShort): UInt = this.times(other.toUInt())
    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UInt): UInt = UInt(this.data.times(other.data))
    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: ULong): ULong = this.times(other.toULong())

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UByte): UInt = this.div(other.toUInt())
    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UShort): UInt = this.div(other.toUInt())
    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UInt): UInt = uintDivide(this, other)
    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: ULong): ULong = this.div(other.toULong())

    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UByte): UInt = this.rem(other.toUInt())
    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UShort): UInt = this.rem(other.toUInt())
    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UInt): UInt = uintRemainder(this, other)
    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: ULong): ULong = this.rem(other.toULong())

    /** Increments this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun inc(): UInt = UInt(data.inc())
    /** Decrements this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun dec(): UInt = UInt(data.dec())

    /** Creates a range from this value to the specified [other] value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rangeTo(other: UInt): UIntRange = UIntRange(this, other)

    /** Shifts this value left by the [bitCount] number of bits. */
    @kotlin.internal.InlineOnly
    public inline infix fun shl(bitCount: Int): UInt = UInt(data shl bitCount)
    /** Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros. */
    @kotlin.internal.InlineOnly
    public inline infix fun shr(bitCount: Int): UInt = UInt(data ushr bitCount)
    /** Performs a bitwise AND operation between the two values. */
    @kotlin.internal.InlineOnly
    public inline infix fun and(other: UInt): UInt = UInt(this.data and other.data)
    /** Performs a bitwise OR operation between the two values. */
    @kotlin.internal.InlineOnly
    public inline infix fun or(other: UInt): UInt = UInt(this.data or other.data)
    /** Performs a bitwise XOR operation between the two values. */
    @kotlin.internal.InlineOnly
    public inline infix fun xor(other: UInt): UInt = UInt(this.data xor other.data)
    /** Inverts the bits in this value. */
    @kotlin.internal.InlineOnly
    public inline fun inv(): UInt = UInt(data.inv())

    /** Converts this [UInt] value to [Byte]. */
    @kotlin.internal.InlineOnly
    public inline fun toByte(): Byte = data.toByte()
    /** Converts this [UInt] value to [Short]. */
    @kotlin.internal.InlineOnly
    public inline fun toShort(): Short = data.toShort()
    /** Converts this [UInt] value to [Int]. */
    @kotlin.internal.InlineOnly
    public inline fun toInt(): Int = data.toInt()
*\n    * The resulting `Int` value has the same binary representation as this `UInt` value.
*\n    @kotlin.internal.InlineOnly
    public inline fun toInt(): Int = data
    /**
     * Converts this [UInt] value to [Long].
     * The resulting `Long` value represents the same numerical value as this `UInt`
     * whereas the least significant 32 bits of the resulting `Long` value are the same as the bits of this `UInt` value.
     */
    @kotlin.internal.InlineOnly
    public inline fun toLong(): Long = data.toLong() and 0xFFFF_FFFF
    /**
     * Converts this [UInt] value to [UByte].
     * The resulting `UByte` value represents the least significant 8 bits of this `UInt`
     * The resulting `UByte` value is represented by the least significant 8 bits of this `UInt` value.
     */
    @kotlin.internal.InlineOnly
    public inline fun toUByte(): UByte = data.toUByte()
    /**
     * Converts this [UInt] value to [UShort].
     * The resulting `UShort` value represents the least significant 16 bits of this `UInt` value.
     */
    @kotlin.internal.InlineOnly
    public inline fun toUShort(): UShort = data.toUShort()
    /** Returns this value. */
    @kotlin.internal.InlineOnly
    public inline fun toUInt(): UInt = this
    /**
     * Converts this [UInt] value to [ULong].
     * The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `UInt` value.
     * whereas the most significant 32 bits are filled with zeros.
     */
    @kotlin.internal.InlineOnly
    public inline fun toULong(): ULong = ULong(data.toLong() and 0xFFFF_FFFF)
    /**
     * Converts this [UInt] value to [Float].
     * The fractional part, if any, is rounded down towards zero.
     * Returns zero if this `Float` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.
     */
    @kotlin.internal.InlineOnly
    public inline fun toFloat(): Float = this.toDouble().toFloat()
    /**
     * Converts this [UInt] value to [Double].
     * The fractional part, if any, is rounded down towards zero.
     * Returns zero if this `Double` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.
     */
    @kotlin.internal.InlineOnly
    public inline fun toDouble(): Double = uintToDouble(data)

    @SinceKotlin("1.3")
    @ExperimentalUnsignedTypes
    @kotlin.internal.InlineOnly
    public inline fun Byte.toUInt(): UInt = UInt(this.toInt())
    /**
     * Converts this [Short] value to [UInt].
     * The least significant 16 bits of the resulting `UInt` value are the same as the bits of this `Short` value.
     */
    @SinceKotlin("1.3")
    @ExperimentalUnsignedTypes
    @kotlin.internal.InlineOnly
    public inline fun Short.toUInt(): UInt = UInt(this.toInt())
    /**
     * Converts this [Int] value to [UInt].
     * The resulting `UInt` value has the same binary representation as this `Int` value.
     */
    @SinceKotlin("1.3")
    @ExperimentalUnsignedTypes
    @kotlin.internal.InlineOnly
    public inline fun Int.toUInt(): UInt = UInt(this)
    /**
     * Converts this [Long] value to [UInt].
     * If this value is positive and less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents the same numerical value as this `Long` value.
     */
    @SinceKotlin("1.3")
    @ExperimentalUnsignedTypes
    @kotlin.internal.InlineOnly
    public inline fun Long.toUInt(): UInt = UInt(this.toLong())

    public override fun toString(): String = toLong().toString()
A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.
public inline operator fun div(other: UShort): Uint = this.toUInt().div(other.toUInt())

    /** Divides this value by the other value. */

    public inline operator fun div(other: UInt): Uint = this.toUInt().div(other)

    /** Divides this value by the other value. */

    public inline operator fun div(other: ULong): ULong = this.toULong().div(other)

    /** Calculates the remainder of dividing this value by the other value. */

    public inline operator fun rem(other: UByte): UByte = data.toByte()

    /** Calculates the remainder of dividing this value by the other value. */

    public inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other)

    /** Calculates the remainder of dividing this value by the other value. */

    public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)

    /** Calculates the remainder of dividing this value by the other value. */

    fun toInt(): Int = data.toInt() and 0xFFFF

    /** Converts this [UShort] value to [Int]. */

    fun toUByte(): UByte = data.toUByte()

    /** Converts this value to [UByte]. */

    fun toUShort(): UShort = data

    /** Converts this value to [UShort]. */

    fun toUByte(): UByte = this.toUInt().rem(other.toUInt())

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()

    /** Calculates the remainder of dividing this value by the other value. */

    fun toUByte(): UByte = data.toUByte()
most significant 48 bits are filled with zeros.

```kotlin
/**
 * Converts this [UShort] value to [Float].
 *
 * The resulting `Float` value represents the same numerical value as this `UShort`.
 */
@kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toInt().toFloat()
```

```kotlin
/**
 * Converts this [UShort] value to [Double].
 *
 * The resulting `Double` value represents the same numerical value as this `UShort`.
 */
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double =
```

```kotlin
/**
 * Converts this [Byte] value to [UShort].
 *
 * If this value is positive, the resulting `UShort` value represents the same numerical value as this `Byte`.
 * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this `Byte` value.
 * whereas the most significant 8 bits are filled with the sign bit of this value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Byte.toUShort(): UShort = UShort(this.toShort())
```

```kotlin
/**
 * Converts this [Short] value to [UShort].
 *
 * If this value is positive, the resulting `UShort` value represents the same binary representation as this `Short`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Short.toUShort(): UShort = UShort(this)
```

```kotlin
/**
 * Converts this [Int] value to [UShort].
 *
 * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents the same numerical value as this `Int`.
 * The resulting `UShort` value is represented by the least significant 16 bits of this `Int` value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Int.toUShort(): UShort = UShort(this.toShort())
```

```kotlin
/**
 * Converts this [Long] value to [UShort].
 *
 * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents the same numerical value as this `Long`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Long.toUShort(): UShort = UShort(this.toShort())
```

```

```
override fun toString(): String = "$first..$last"

    override fun toString(): String = "$first..$last"

    companion object {
        /** An empty range of values of type Long. */
        public val EMPTY: LongRange = LongRange(1, 0)

    }

    /** An empty range of values of type Long. */
    public val EMPTY: LongRange = LongRange(1, 0)

    class CollectionsKt

import kotlin.collections

internal object EmptyIterator : ListIterator<Nothing> {
    override fun hasNext(): Boolean = false
    override fun hasPrevious(): Boolean = false
    override fun nextIndex(): Int = 0
    override fun previousIndex(): Int = 0
    override fun next(): Nothing = throw NoSuchElementException()
    override fun previous(): Nothing = throw NoSuchElementException()
}

internal object EmptyList : List<Nothing>, Serializable, RandomAccess {
    private const val serialVersionUID: Long = -7390468764508069838L

    override fun equals(other: Any?): Boolean = other is List<*>
    override fun hashCode(): Int = 1
    override fun toString(): String = "[]"

    override val size: Int get() = 0
    override fun isEmpty(): Boolean = true
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun get(index: Int): Nothing = throw IndexOutOfBoundsException("Empty list doesn't contain element at index $index.")
    override fun indexOf(element: Nothing): Int = -1
    override fun lastIndexOf(element: Nothing): Int = -1

    override fun iterator(): Iterator<Nothing> = EmptyIterator
    override fun listIterator(): ListIterator<Nothing> = EmptyIterator
    override fun listIterator(index: Int): ListIterator<Nothing> {
        if (index != 0) throw IndexOutOfBoundsException("Index: $index")
        return EmptyIterator
    }

    override fun subList(fromIndex: Int, toIndex: Int): List<Nothing> {
        if (fromIndex == 0 && toIndex == 0) return this
    }

    private fun readResolve(): Any = EmptyList
}

/**
 * Returns an empty read-only list.  The returned list is serializable (JVM).
 * @sample samples.collections.Collections.Lists.emptyReadOnlyList
 */
public fun <T> emptyList(): List<T> = EmptyList

/**
 * Returns a new read-only list of given elements.  The returned list is serializable (JVM).
 * @sample samples.collections.Collections.Lists.readOnlyList
 */
public fun <T> emptyList(vararg elements: T): List<T> = if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs = true))

/**
 * Returns a new read-only list either of single given element, if it is not null, or empty list if the element is null.
 */
@SinceKotlin("1.1")
public fun <T> list(): List<T> = ArrayList()
returned list is serializable (JVM).

* @sample samples.collections.Collections.Lists.listOfNotNull

@public fun <T : Any> listOfNotNull(element: T?): List<T> = if (element != null) listOf(element) else emptyList()

Returns a new read-only list only of those given elements, that are not null. The returned list is serializable (JVM).

* @sample samples.collections.Collections.Lists.listOfNotNull

@fun <T : Any> listOfNotNull(vararg elements: T?): List<T> = elements.filterNotNull().toList()

Creates a new read-only list with the specified [size], where each element is calculated by calling the specified [init] function. The function [init] is called for each list element sequentially starting from the first one. It should return the value for a list element given its index. The function [init] is called for each list element sequentially starting from the first one. It should return the value for a list element given its index. The function [init] is called for each list element sequentially starting from the first one. It should return the value for a list element given its index.

* @sample samples.collections.Collections.Lists.readOnlyListFromInitializer

@fun <T : Any> MutableList(size: Int, init: (index: Int) -> T): MutableList<T> {
    val list = ArrayList<T>(size)
    repeat(size) { index -> list.add(init(index)) }
    return list
}

* @sample samples.collections.builders.CollectionsBuilders.listBuilder

@fun <T> buildList(@BuilderInference builderAction: MutableList<E>.() -> Unit): List<E> {
    contract {
        callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE)
    }
    return ArrayList<E>().apply(builderAction)
}

* @sample samples.collections.builders.CollectionsBuilders.listBuilder

@fun <T> buildList(capacity: Int, @BuilderInference builderAction: MutableList<E>.() -> Unit): List<E> {
    contract {
        callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE)
    }
    checkBuilderCapacity(capacity)
    return ArrayList<E>(capacity).apply(builderAction)
}

* @sample samples.collections.Collections.Collections.indicesOfCollection

@collection<*>.indices: IntRange
    get() = 0..size - 1

* @sample samples.collections.Collections.Collections.listFiles

@collection<*>.lastIndex: Int
    get() = this.size - 1

* @sample samples.collections.Collections.Collections.isNotEmpty

@kotlin.internal.InlineOnly
@fun <T> Collection<T>.isNotEmpty(): Boolean = !isEmpty()

* @sample samples.collections.Collections.Collections.isEmpty

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@fun <T> Collection<T>?.orEmpty(): Collection<T> = this ?: emptyList()
/**
 * Checks if all elements in the specified collection are contained in this collection.
 * Allows to overcome type-safety restriction of `containsAll` that requires to pass a collection of type `Collection<E>`.
 * @param elements The collection to check for containment.
 * @return true if all elements are contained, false otherwise.
 * @since 1.3
 */
public inline fun <E, R> Collection<E>.containsAll(elements: Collection<R>): Boolean {
    return this.forEach { element ->
        if (element !in this)
            return false
    }
    return true
}

/**
 * Searches this list or its range for the provided [element] using the binary search algorithm.
 * The list is expected to be sorted into ascending order according to the Comparable natural ordering of its elements.
 * otherwise the result is undefined.
 * If the list contains multiple elements equal to the specified [element], there is no guarantee which one will be found.
 * `null` value is considered to be less than any non-null value.
 * @param index The index of the element to find.
 * @param start The starting index of the range to search.
 * @param end The ending index of the range to search.
 * @return the index of the element, if it is contained in the list within the specified range;
 * otherwise, the inverted insertion point (`-insertion point - 1`).
 * The insertion point is defined as the index at which the element should be inserted,
 * so that the list (or the specified subrange of list) still remains sorted.
 */
public fun <T : Comparable<T>> List<T>.binarySearch(element: T, fromIndex: Int = 0, toIndex: Int = size): Int {
    rangeCheck(size, fromIndex, toIndex)
    var low = fromIndex
    var high = toIndex - 1
    while (low <= high) {
        val mid = (low + high).ushr(1) // safe from overflows
        val cmp = comparator.compare(mid, element)
        if (cmp < 0)
            low = mid + 1
        else if (cmp > 0)
            high = mid - 1
        else
            return mid // key found
    }
    return -(low + 1) // key not found
}

/**
 * Searches this list or its range for an element having the key returned by the specified [selector] function
 * equal to the provided [key] using the binary search algorithm.
 * otherwise the result is undefined.
 * If the list contains multiple elements with the specified [key], there is no guarantee which one will be found.
 * `null` value is considered to be less than any non-null value.
 * @param selector The selector function.
 * @param key The key to find.
 * @return the index of the element with the specified [key], if it is contained in the list within the specified range;
 * otherwise, the inverted insertion point (`-insertion point - 1`).
 * The insertion point is defined as the index at which the element should be inserted,
 * so that the list (or the specified subrange of list) still remains sorted according to the specified [comparator].
 */
public inline fun <T, K : Comparable<K>> List<T>.binarySearchBy(key: K?, fromIndex: Int = 0, toIndex: Int = size, crossinline selector: (T) -> K?){
    return binarySearch(fromIndex, toIndex) { selector(it) == key }
}
fromIndex: Int = 0, toIndex: Int = size(), selector: (T) -> K) Int =\n// binarySearch(fromIndex, toIndex) { comparator.compare(selector(it), key) }\n\n// Searches this list or its range for an element for which the
given [comparison] function returns zero using the binary search algorithm.\n*\n* The list is expected to be sorted so that the signs of the [comparison] function's return values ascend on the list elements; i.e. negative values come before zero and zeroes come before positive values.\n*\n* Otherwise, the result is undefined.\n*\n* If the list contains multiple elements for which [comparison] returns zero, there is no guarantee which one will be found.\n*\n* @param comparison function that returns zero when called on the list element being searched.\n*\n* On the elements coming before the target element, the function must return negative values;\n* on the elements coming after the target element, the function must return positive values.\n*\n* @return the index of the found element, if it is contained in the list within the specified range;\n* otherwise, the inverted insertion point '(insertion point - \n* 1)'.\n*\n* The insertion point is defined as the index at which the element should be inserted;\n* so that the list (or the specified subrange of list) still remains sorted.\n*\n* @param sample samples.collections.Collections.Lists.binarySearchWithComparisonFunction\n*\n
fromIndex: Int = 0, toIndex: Int = size, comparator: (T) -> Int) Int =
//\n//\n\n/**\n * Searches this list or its range for an element for which the\ngiven [comparison] function returns zero using the binary search algorithm.\n *\n * The list is expected to be sorted so that the signs of the [comparison] function's return values ascend on the list elements;\n * i.e. negative values come before zero and zeroes come before positive values.\n * Otherwise, the result is undefined.\n *\n * If the list contains multiple elements for which [comparison] returns zero, there is no guarantee which one will be found.\n * On the elements coming before the target element, the function must return negative values;\n * on the elements coming after the target element, the function must return positive values.\n *\n * @return the index of the found element, if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point '(insertion point - 1)'.\n *\n * The insertion point is defined as the index at which the element should be inserted; so that the list (or the specified subrange of list) still remains sorted.\n *\n * @param comparison function that returns zero when called on the list element being searched.\n * On the elements coming before the target element, the function must return negative values;\n * on the elements coming after the target element, the function must return positive values.\n *\n * @return the index of the found element, if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point '(insertion point - 1)'.\n *\n * The insertion point is defined as the index at which the element should be inserted; so that the list (or the specified subrange of list) still remains sorted.\n *\n * @param sample samples.collections.Collections.Lists.binarySearchWithComparisonFunction\n *\n
List<T>.binarySearch(fromIndex: Int = 0, toIndex: Int = size, comparator: (T) -> Int) Int =
//\n//\n\n/**\n * Searches this list or its range for an element for which the\ngiven [comparison] function returns zero using the binary search algorithm.\n *\n * The list is expected to be sorted so that the signs of the [comparison] function's return values ascend on the list elements;\n * i.e. negative values come before zero and zeroes come before positive values.\n * Otherwise, the result is undefined.\n *\n * If the list contains multiple elements for which [comparison] returns zero, there is no guarantee which one will be found.\n * On the elements coming before the target element, the function must return negative values;\n * on the elements coming after the target element, the function must return positive values.\n *\n * @return the index of the found element, if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point '(insertion point - 1)'.\n *\n * The insertion point is defined as the index at which the element should be inserted; so that the list (or the specified subrange of list) still remains sorted.\n *\n * @param comparison function that returns zero when called on the list element being searched.\n * On the elements coming before the target element, the function must return negative values;\n * on the elements coming after the target element, the function must return positive values.\n *\n * @return the index of the found element, if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point '(insertion point - 1)'.\n *\n * The insertion point is defined as the index at which the element should be inserted; so that the list (or the specified subrange of list) still remains sorted.\n *\n * @param sample samples.collections.Collections.Lists.binarySearchWithComparisonFunction\n *\n
private object EmptyMap : Map<Any?, Nothing>, Serializable {
    private const val serialVersionUID: Long = 8246714829545688274

    override fun equals(other: Any?): Boolean = other is Map<*, *> && other.isEmpty()
    override fun hashCode(): Int = 0
    override fun toString(): String = "{}"

    override val size: Int get() = 0
    override fun isEmpty(): Boolean = true

    override fun containsKey(key: Any?): Boolean = false
    override fun containsValue(value: Nothing): Boolean = false
    override fun containsKeyOrValue(key: Any?): Boolean = false
    override fun containsKeyOrValue(value: Nothing): Boolean = false
    override fun containsKey(key: Any?, value: Nothing): Boolean = false
    override fun containsValue(key: Nothing?, value: Any?): Boolean = false
    override fun containsValue(key: Nothing?, value: Nothing): Boolean = false

    override fun get(key: Any?): Nothing? = null
    override fun get(key: Any?): Nothing? = null
    override fun get(key: Nothing?): Nothing? = null


    override fun remove(key: Any?): Nothing? = null
    override fun remove(key: Nothing?): Nothing? = null


    override fun putAll(map: Map<Any?, Nothing>): Nothing? = null
    override fun putAll(map: Map<Any?, Nothing>): Nothing? = null

    override fun keySet(): Set<Any> = EmptySet
    override fun keySet(): Set<Nothing> = EmptySet

    override fun values(): Collection<Nothing> = EmptyList

    private fun readResolve(): Any = EmptyMap
}

/**\n * Returns an empty read-only map of specified type.\n *\n * The returned map is serializable (JVM).\n *\n * @param other Any other Map\n *\n * @return an empty read-only map of specified type that is\n * serializable (JVM).\n *\n * @throws IllegalArgumentException if the specified\n * type is not the Map\n *\n * @throws UnsupportedOperationException if the implementation\n * does not support the\n *\n * @since 1.3\n */

fun <K, V> emptyMap(): Map<K, V> = EmptyMap
emptyMap()\n/**\n * Returns an empty read-only map.\n * The returned map is serializable (JVM).\n */
@sample samples.collections.Maps.Instantiation.emptyReadOnlyMap\n
/**\n * Returns a new [MutableMap] with the specified contents, given as a list of pairs\n * where the first component is the key and the second is the value.\n * If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs.\n */
@sample samples.collections.Maps.Instantiation.mutableMap\n
/**\n * Returns an empty new [HashMap].\n */
@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> hashMapOf(): HashMap<K, V> = HashMap<K, V>()

/**\n * Returns a new [HashMap] with the specified contents, given as a list of pairs\n * where the first component is the key and the second is the value.\n * If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs.\n */
public fun <K, V> hashMapOf(vararg pairs: Pair<K, V>): HashMap<K, V> = hashMapOf(mapCapacity(pairs.size)).apply { putAll(pairs) }

/**\n * Builds a new read-only [Map] by populating a [MutableMap] using the given [builderAction] and returning a read-only map with the same key-value pairs.\n * The map passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n * Entries of the map are iterated in the order they were added by the [builderAction].\n */
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> buildMap(<caret>builderAction: @BuilderInference MutableMap<K, V>.() -> Unit): Map<K, V> {
    contract { callsInPlace(<caret>builderAction, InvocationKind.EXACTLY_ONCE) }\n    return LinkedHashMap<K, V>().apply(<caret>builderAction)
}

/**\n * Calculates the initial capacity of a map.\n */
@PublishedApi\ninternal expect fun mapCapacity(expectedSize: Int): Int

Open Source Used In DNAC Platform Antelope 7959
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@PublishedApi
internal expect fun

checkBuilderCapacity(capacity: Int): Boolean * Returns `true` if this map is not empty. * @sample samples.collections.Maps_USAGE.mapIsNotEmpty

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@PublishedApi
internal inline fun

Map<out K, V>.isNotEmpty(): Boolean = !isEmpty() * Returns `true` if this map is not empty. * @sample samples.collections.Maps_USAGE.mapIsNotEmpty

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@PublishedApi
internal inline fun

Map<out K, V>.isNullOrEmpty(): Boolean {
    return this == null || !isEmpty()
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@PublishedApi
internal inline fun

Map<out K, V>.orElse(defaultValue: () -> R): R where M : Map<k, v>, M : R = R {
    if (isNullOrEmpty())
        return defaultValue() else this
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@PublishedApi
internal inline fun

Map<out K, V>.contains(key: K): Boolean = containsKey(key)

@kotlin.internal.InlineOnly
```kotlin
internal inline fun <K, V> Map<K, V>.getOrElseNullable(key: K, defaultValue: () -> V): V {
    return when {
        value != null -> value
        else -> defaultValue()
    }
}

/**
 * Returns the value for the given [key] or throws an exception if there is no such key in the map.
 * If the map was created by [withDefault], resorts to its `defaultValue` function instead of throwing an exception.
 */
@SinceKotlin("1.1")
public fun <K, V> Map<K, V>.getValue(key: K): V = getOrImplicitDefault(key)

/**
 * Returns the value for the given key. If the key is not found in the map, calls the [defaultValue] function.
 * Note that the operation is not guaranteed to be atomic if the map is being modified concurrently.
 */
@sample collections.Maps.Usage.getOrPut
    return if (value != null) {
        put(key, answer)
        answer
    } else {
        getOrImplicitDefault(key)
    }
}

/**
 * Returns an [Iterator] over the entries in the [Map].
 */
@kotlin.internal.InlineOnly

/**
 * Returns a [MutableIterator] over the mutable entries in the [Map].
 */
@kotlin.jvm.JvmName("mutableIterator")
@kotlin.internal.InlineOnly

/**
 * Puts all the given [pairs] into this [MutableMap] with the first component in the pair being the key and the second the value.
 */
public fun <K, V> MutableMap<in K, in V>.putAll(pairs: Array<out Pair<K, V>>): Unit {
    for ((key, value) in pairs) {
        put(key, value)
    }
}

/**
 * Puts all the elements of the given collection into this [MutableMap] with the first component in the pair being the key and the second the value.
 */
public fun <K, V> MutableMap<in K, in V>.putAll(pairs: Iterable<Pair<K, V>>): Unit {
    for ((key, value) in pairs) {
        put(key, value)
    }
}

/**
 * Puts all the elements of the given sequence into this [MutableMap] with the first component in the pair being the key and the second the value.
 */
public fun <K, V> MutableMap<in K, in V>.putAll(pairs: Sequence<Pair<K, V>>): Unit {
    for ((key, value) in pairs) {
        put(key, value)
    }
}
```

---

Open Source Used In DNAC Platform Antelope 7961
*/
public inline fun <K, V> Map<out K, V>.filterKeys(predicate: (K) -> Boolean): Map<K, V> {
    var result = LinkedHashMap<K, V>()
    for (entry in this) {
        if (predicate(entry.key)) {
            result.put(entry.key, entry.value)
        }
    }
    return result
}

/**
 * Returns a map containing all key-value pairs matching  *
 * the given [predicate] parameter.
 */
public inline fun <K, V> Map<out K, V>.filterValues(predicate: (V) -> Boolean): Map<K, V> {
    var result = LinkedHashMap<K, V>()
    for (entry in this) {
        if (predicate(entry.value)) {
            result.put(entry.key, entry.value)
        }
    }
    return result
}

/**
 * Returns a new map containing all key-value pairs matching  *
 * the given [predicate] parameter.
 */
public inline fun <K, V> Map<out K, V>.filterTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {
    var result = destination
    for (entry in this) {
        if (predicate(entry)) {
            result.put(entry.key, entry.value)
        }
    }
    return result
}

/**
 * Returns a new map containing all key-value pairs from the  *
 * given collection of pairs.
 */
    return filterTo(LinkedHashMap<K, V>(), predicate)
}

/**
 * Returns a new map containing all key-value pairs from the  *
 * given sequence of pairs.
 */
public inline fun <K, V> Sequence<Pair<K, V>>.toMap(): Map<K, V> = toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()

/**
 * Populates and returns the [destination] mutable map with  *
 * key-value pairs from the given sequence of pairs.
 */
    destination.apply { putAll(this@toMap) }

/**
 * Returns a new map containing all key-value pairs from the  *
 * original map.
 */
public fun <K, V> Map<out K, V>.toMap(): Map<K, V> = when (size) {
    0 -> emptyMap()
    1 -> toSingletonMap()
    else -> toMutableMap()
}

/**
 * Returns a new map containing all key-value pairs from the  *
 * given array of pairs.
 */
public fun <K, V> Array<out Pair<K, V>>.toMap(): Map<K, V> = when (size) {
    0 -> emptyMap()
    1 -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))
    else -> toMap(LinkedHashMap<K, V>(mapCapacity(size))).optimizeReadOnlyMap()
}

/**
 * Populates and returns the [destination] mutable map with  *
 * key-value pairs from the given array of pairs.
 */
    destination.apply { putAll(this@toMap) }

/**
 * Returns a new read-only map containing all key-value pairs  *
 * from the original map.
 */
@SinceKotlin("1.1")
public fun <K, V> Map<out K, V>.toMap(): Map<K, V> = when (size) {
    0 -> emptyMap()
    1 -> toSingletonMap()
    else -> toMutableMap()
}
new mutable map containing all key-value pairs from the original map. The returned map preserves the entry iteration order of the original map.

* @SinceKotlin("1.1")
**public fun <K, V> Map<out K, V>.toMutableMap(): MutableMap<K, V> = LinkedHashMap(this)
**
* Populates and returns the [destination] mutable map with key-value pairs from the given map.

* @SinceKotlin("1.1")
* The returned map preserves the entry iteration order of the original map.

* Those [pairs] with unique keys are iterated in the end in the order of [pairs] collection.

* @SinceKotlin("1.1")
* If (this.isEmpty()) returns a new read-only map by replacing or adding entries to this map from a given collection of key-value [pairs].

* The returned map preserves the entry iteration order of the original map.

* Those [pairs] with unique keys are iterated in the end in the order of [pairs] collection.

* The returned map preserves the entry iteration order of the original map. Those [pairs] with unique keys are iterated in the end in the order of [pairs] collection.

* Those entries of another [map] that are missing in this map are iterated in the end in the order of that [map].

* Appends or replaces all entries from the given collection of [pairs] in this mutable map.

* Returns a map containing all entries of the original map except the entry with the given [key].

* The returned map preserves the entry iteration order of the original map.

* Returns a map containing all entries of the original map except those entries

* the keys of which are contained in the given [keys] collection.

* The returned map preserves the entry iteration order of the original map.

* Returns a map containing all entries of the original map except those entries

* the keys of which are contained in the given [keys] collection.

* The returned map preserves the entry iteration order of the original map.

* Returns a map containing all entries of the original map except those entries

* the keys of which are contained in the given [keys] collection.

* The returned map preserves the entry iteration order of the original map.

* Returns a map containing all entries of the original map except those entries

* the keys of which are contained in the given [keys] collection.

* The returned map preserves the entry iteration order of the original map.

* Returns a map containing all entries of the original map except those entries

* the keys of which are contained in the given [keys] collection.

* The returned map preserves the entry iteration order of the original map.

* Returns a map containing all entries of the original map except those entries

* the keys of which are contained in the given [keys] collection.

* The returned map preserves the entry iteration order of the original map.
```kotlin
/**
 * Returns a map containing all entries of the original map except those entries
 * the keys of which are contained in the given [keys] sequence.
 * The returned map preserves the entry iteration order of the original map.
 */
@SinceKotlin("1.1.")
public fun <K, V> Map<K, V> remove(keys: Sequence<K>): Map<K, V> =
    this.toMutableMap().apply {
        minusAssign(keys)
    }.optimizeReadOnlyMap()

/**
 * Removes all entries the keys of which are contained in the given [keys] array from this mutable map.
 */
@SinceKotlin("1.1.")
@kotlin.internal.InlineOnly
public inline fun <K, V> MutableMap<K, V>.removeAssign(keys: Array<out K>) {
    this.keys.removeAll(keys)
}

/**
 * Removes all entries the keys of which are contained in the given [keys] sequence from this mutable map.
 */
@SinceKotlin("1.1.")
@kotlin.internal.InlineOnly
public inline fun <K, V> MutableMap<K, V>.removeAssign(keys: Sequence<K>) {
    this.keys.removeAll(keys)
}

@PublishedApi
internal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) {
    0 -> emptyMap()
    1 -> toSingletonMapOrSelf()
    else -> this
}

// do not expose for now
@PublishedApi
internal fun <K, V> Map<K, V>.toSingletonMapOrSelf() = when (size) {
    0 -> emptyMap()
    else -> this
}
```

```kotlin
import kotlin.contracts.*

internal object EmptySet : Set<Nothing>, Serializable {
    private const val serialVersionUID: Long = 340660374387020532

    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()
    override fun hashCode(): Int = 0
    override fun toString(): String = "["]"

    override val size: Int get() = 0
    override fun isEmpty(): Boolean = true
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun iterator(): Iterator<Nothing> = EmptyIterator

    private fun readResolve(): Any = EmptySet
}

/**
 * Returns an empty read-only set. The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.emptyReadOnlySet
 */
public fun <T> emptySet(): Set<T> = EmptySet

/**
 * Returns a new read-only set with the given elements.
 * Elements of the set are iterated in the order they were specified.
 * The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.readOnlySet
 */
public fun <T> setOf(vararg elements: T): Set<T> = if (elements.size > 0) elements.toSet() else emptySet()

/**
 * Returns an empty new [HashSet].
 */
@SinceKotlin("1.1.")
@kotlin.internal.InlineOnly
public inline fun <T> hashSetOf(): HashSet<T> = HashSet()

/** Returns a new [HashSet] with the given elements. */
public fun <T> hashSetOf(vararg elements: T): HashSet<T> = elements.toCollection(HashSet(mapCapacity(elements.size)))

/**
 * Returns an empty new [LinkedHashSet].
 */
@SinceKotlin("1.1.")
@kotlin.internal.InlineOnly
public inline fun <T> linkedSetOf(): LinkedHashSet<T> = LinkedHashSet()
```

```kotlin
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SetsKt")
@file:UseExperimental(kotlin.experimental.ExperimentalTypeInference::class)

package kotlin.collections

import kotlin.contracts.*

internal object EmptySet : Set<Nothing>, Serializable {
    private const val serialVersionUID: Long = 340660374387020532

    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()
    override fun hashCode(): Int = 0
    override fun toString(): String = "["]"

    override val size: Int get() = 0
    override fun isEmpty(): Boolean = true
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun iterator(): Iterator<Nothing> = EmptyIterator

    private fun readResolve(): Any = EmptySet
}

/**
 * Returns an empty read-only set. The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.emptyReadOnlySet
 */
public fun <T> emptySet(): Set<T> = EmptySet

/**
 * Returns a new read-only set with the given elements.
 * Elements of the set are iterated in the order they were specified.
 * The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.readOnlySet
 */
public fun <T> setOf(vararg elements: T): Set<T> = if (elements.size > 0) elements.toSet() else emptySet()

/** Returns an empty new [HashSet].
 */
@SinceKotlin("1.1.")
@kotlin.internal.InlineOnly
public inline fun <T> hashSetOf(): HashSet<T> = HashSet()

/** Returns a new [HashSet] with the given elements. */
public fun <T> hashSetOf(vararg elements: T): HashSet<T> = elements.toCollection(HashSet(mapCapacity(elements.size)))

/**
 * Returns an empty new [LinkedHashSet].
 */
@SinceKotlin("1.1.")
@kotlin.internal.InlineOnly
public inline fun <T> linkedSetOf(): LinkedHashSet<T> = LinkedHashSet()
```
protected open fun <T> linkedSetOf(vararg elements: T): LinkedHashSet<T> =
    elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))

/**
 * Returns a new [LinkedHashSet] with the given elements.
 * Elements of the set are iterated in the order they were specified.
 * @sample samples.collections.Collections.Sets.linkedHashSet
 */
public fun <T> linkedSetOf(vararg elements: T): LinkedHashSet<T> =
    elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))

/**
 * Builds a new read-only [Set] by populating a [MutableSet] using the given [builderAction] and
 * returning a read-only set with the same elements.
 * The set passed as a receiver to the [builderAction] is valid only inside that function.
 * Using it outside of the function produces an unspecified behavior.
 * Elements of the set are iterated in the order they were added by the [builderAction].
 * The set passed as a receiver to the [builderAction] is valid only inside that function.
 * Using it outside of the function produces an unspecified behavior.
 * @throws IllegalArgumentException if the given [capacity] is negative.
 * @sample samples.collections.Builders.Sets.buildSetSample
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <E> buildSet(@BuilderInference builderAction: MutableSet<E>.() -> Unit): Set<E> {
    contract {
        callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE)
    }
    return
    LinkedHashSet<E>().apply(builderAction)
}

/**
 * Builds a new read-only [Set] by populating a [MutableSet]
 * and returning a read-only set with the same elements.
 * Elements of the set are iterated in the order they were added by the [builderAction].
 * @throws IllegalArgumentException if the given [capacity] is negative.
 * @sample samples.collections.Builders.Sets.buildSetSample
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <E> buildSet(capacity: Int, @BuilderInference builderAction: MutableSet<E>.() -> Unit): Set<E> {
    contract {
        callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE)
    }
    checkBuilderCapacity(capacity)
    return
    LinkedHashSet<E>(mapCapacity(capacity)).apply(builderAction)
}

/** Returns this Set if it's not `null` and the empty set otherwise. */
@kotlin.internal.InlineOnly
public inline fun <T> Set<T>?.orEmpty(): Set<T> = this

internal fun <T> Set<T>.optimizeReadOnlySet() = when (size) {
    0 -> emptySet()
    1 ->
        setOf(iterator().next())
    else -> this
}

Open Source Used In DNAC Platform Antelope 7965
```kotlin
fun parseHexOrDecimalOrOctal(num: String): Long? {
    // Possible leading sign
    if (num.length == 1) return null // non-digit (possible sign) only, no digits after
    var length = num.length
    length = if (firstChar < '0') { // Possible leading sign
        length == 1 // non-digit (possible sign) only, no digits after
        return null
    } else { // if (firstChar == '+' or '-' or '0')
        var start = 0
        if (firstChar == '-') {
            isNegative = true
            limit = Long.MIN_VALUE
        } else if (firstChar == '+') {
            isNegative = false
            limit = -Long.MAX_VALUE
        } else
            return null
    } else {
        start = 0
        isNegative = false
        limit = -Long.MAX_VALUE
    }

    val limitForMaxRadix = (-Long.MAX_VALUE) / 36
    var limitBeforeMul = limitForMaxRadix * val limitBeforeMul = limitForMaxRadix * var result = 0
    for (i in start until length) {
        val digit = digitOf(this[i], radix)
        if (digit < 0) return null
        if (result < limitBeforeMul) {
            if (limitBeforeMul == limitForMaxRadix) {
                limitBeforeMul = limit / radix
                if (result < limitBeforeMul) {
                    return null
                }
            } else {
                return null
            }
        }
        result *= radix
        if (result < limit + digit) return null
        result -= digit
    }
    return if (isNegative) result else -result
}

/**
 * Parses the string as a Long number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 * * @throws NumberFormatException when pl type \"Invalid number format: \"input\"\" is not a valid representation of a number.
 * * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * * @since Kotlin(\"1.1\")\n */
@file:kotlin.jvm.JvmName("UnsignedKt")
@file:UseExperimental(ExperimentalUnsignedTypes::class)
package kotlin

@PublishedApi
fun uintCompare(v1: Int, v2: Int): Int = (v1 xor Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)
@PublishedApi

@PublishedApi
fun uintDivide(v1: UInt, v2: UInt): UInt = (v1.toLong() / v2.toLong()).toUInt()
@PublishedApi
fun ulongDivide(v1: ULong, v2: ULong): ULong {
    val dividend = v1.toLong()
    val divisor = v2.toLong()
    if (divisor < 0) { // i.e., divisor >= 2^63
        return if (v1 < v2) {
            ULong(0)
        } else {
            ULong(v1 + v2)
        }
    }
    if (divisor >= 2^63) {
        return if (v1 < v2) {
            ULong(dividend / divisor)
        } else {
            ULong(dividend % divisor)
        }
    }
    val quotient = ((dividend ushr 1) / divisor) shl 1
    val rem = dividend - quotient * divisor
    return if (isIntegerResult) result else -(quotient ushr 1) + remainder
}

@PublishedApi
fun uintRemainder(v1: UInt, v2: UInt): UInt = (v1.toLong() % v2.toLong()).toUInt()
@PublishedApi
fun ulongRemainder(v1: ULong, v2: ULong): ULong {
    val dividend = v1.toLong()
    val divisor = v2.toLong()
    if (divisor < 0) { // i.e., divisor >= 2^63
        return if (v1 < v2) {
            ULong(0)
        } else {
            ULong(v1 + v2)
        }
    }
    if (divisor >= 2^63) {
        return if (v1 < v2) {
            ULong(dividend / divisor)
        } else {
            ULong(dividend % divisor)
        }
    }
    val quotient = ((dividend ushr 1) / divisor) shl 1
    val rem = dividend - quotient * divisor
    return if (isIntegerResult) result else -(quotient ushr 1) + remainder
}
```

/ divisor) shl 1 \n val rem = dividend - quotient * divisor\n return ULong(rem - if (ULong(rem) >= ULong.MAX_VALUE) toDouble() - ULong.MAX_VALUE) \n v >= ULong.MAX_VALUE.toDouble() - ULong.MAX_VALUE \n v <= Int.MAX_VALUE.toDouble() toInt(), toInt() \n else -> (v - Int.MAX_VALUE).toInt() + Int.MAX_VALUE.toUInt() // Int.MAX_VALUE < v < ULong.MAX_VALUE\n)\n@PublishedApi\ninternal fun doubleToUInt(v: Double): UInt = when {
    v.isNaN() -> 0u
    v <= UInt.MIN_VALUE.toDouble() ->(UInt.MIN_VALUE\n    v < Long.MAX_VALUE.toDouble() -> v.toLong().toUInt()\n    // Real values from Long.MAX_VALUE to (Long.MAX_VALUE + 1) are not representable in Double, so don't handle them.\n    else -> (v - 9223372036854775808.0).toLong().toUInt() + Int.MAX_VALUE\n    // Int.MAX_VALUE + 1 < v < ULong.MAX_VALUE\n)\n@PublishedApi\ninternal fun doubleToULong(v: Double): ULong = when {
    v.isNaN() -> 0u
    v <= ULong.MIN_VALUE.toDouble() -> ULong.MIN_VALUE\n    v < ULong.MAX_VALUE.toDouble() -> v.toULong()\n    else -> (v - 9223372036854775808.0).toLong().toULong() + 9223372036854775808UL\n    // Long.MAX_VALUE + 1 < v < ULong.MAX_VALUE\n)\n@PublishedApi\ninternal fun uintToDouble(v: Int): Double = (v and Int.MAX_VALUE).toDouble() + (v ushr 31 shl 30).toDouble() * 2\n@PublishedApi\ninternal fun ulongToDouble(v: Long): Double = (v ushr 11).toDouble() * 2048 + (v and 2047)\ninternal fun ulongToString(v: Long): String = ulongToString(v, 10)\ninternal fun ulongToString(v: Long, base: Int): String {
    var quotient = ((v ushr 1) / base) shl 1
    var rem = v - quotient * base
    if (rem >= base) {
        rem -= base
        quotient += 1
    }
    return quotient.toString(base) + rem.toString(base)\n}"
ArrayList<T>(expectedSize)
    val listR = ArrayList<R>(expectedSize)
    for (pair in this) {
        listT.add(pair.first)
        listR.add(pair.second)
    }
    return listT to listR
}

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SequencesKt")

package

/**
 * Given an [iterator] function constructs a [Sequence] that returns values through the
 * [Iterator] provided by that function. The values are evaluated lazily, and the sequence is potentially
 * infinite.
 */
@sample samples.collections.Sequences.Building.sequenceFromIterator

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SequencesKt")

package

/**
 * Given an [iterator] function constructs a [Sequence] that returns values through the
 * [Iterator] provided by that function. The values are evaluated lazily, and the sequence is potentially
 * infinite.
 */
@sample samples.collections.Sequences.Building.sequenceFromIterator

iterator<T>.asSequence(): Sequence<T> = Sequence { this }.constrainOnce()

/**
 * Creates a sequence that returns the specified values.
 */
@sample samples.collections.Sequences.Building.sequenceOfValues

/**
 * Returns an empty sequence.
 */
public fun <T> emptySequence(): Sequence<T> = EmptySequence

private object EmptySequence : Sequence<Nothing>,

/**
 * Creates a sequence that iterates through the elements either of
 * this sequence
 * or, if this sequence turns out to be empty, of the sequence returned by [defaultValue] function.
 */
public fun <T> Sequence<T>.ifEmpty(defaultValue: () -> Sequence<T>): Sequence<T> = sequence {
    val iterator = this@ifEmpty.iterator()
    if (iterator.hasNext()) {
        yieldAll(iterator)
    } else {
        yieldAll(defaultValue())
    }
}

/**
 * Returns a sequence of all elements from all sequences in this sequence.
 */
public fun <T> Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

/**
 * Returns a sequence of all
 * elements from all iterables in this sequence.
 */
@kotlin.jvm.JvmName("flattenSequenceOfIterable")
public fun <T> Sequence<Iterable<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

private fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>):
    Sequence<R> {
        if (this is TransformingSequence<*, *>).
            return (this as TransformingSequence<*,
        T>).flatten(iterator)
    return FlatteningSequence(this, { it }, iterator)
}

/**
 * Returns a pair of lists,
 * where
 * *first* list is built from the first values of each pair from this sequence,
 * *second* list is built from the second values of each pair from this sequence.
 */
public fun <T, R> Sequence<Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {
    val listT = ArrayList<T>()
    val listR = ArrayList<R>()
    for (pair in this) {
        listT.add(pair.first)
        listR.add(pair.second)
    }
    return listT to listR
}

internal class

/**
 * A sequence that returns the values from the underlying [sequence] that either match or do not
 * match the specified [predicate].
 */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    if (this is TransformingSequence<*, *>).
    return this as TransformingSequence<*,
}

/**
 * Returns a sequence that returns the values from the underlying [sequence] that either match or do not
 * match the specified [predicate].
 */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    if (this is TransformingSequence<*, *>).
    return this as TransformingSequence<*,
}

/**
 * Returns a sequence that returns the values from the underlying [sequence] that either match or do not
 * match the specified [predicate].
 */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    if (this is TransformingSequence<*, *>).
    return this as TransformingSequence<*,
}

/**
 * Returns a sequence that returns the values from the underlying [sequence] that either match or do not
 * match the specified [predicate].
 */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    if (this is TransformingSequence<*, *>).
    return this as TransformingSequence<*,
}

/**
 * Returns a sequence that returns the values from the underlying [sequence] that either match or do not
 * match the specified [predicate].
 */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    if (this is TransformingSequence<*, *>).
    return this as TransformingSequence<*,
}

/**
 * Returns a sequence that returns the values from the underlying [sequence] that either match or do not
 * match the specified [predicate].
 */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    if (this is TransformingSequence<*, *>).
    return this as TransformingSequence<*,
}
return
            if (dropState == -1)
                return done
            dropState = 1
        }

        @Suppress("UNCHECKED_CAST")
        if (dropState == 1)
            iterator.

        dropState = 0
    }

        override fun next(): T {
            if (dropState == -1)
                drop()

            if (dropState == 1) {
                val result = nextItem as T
                nextItem = null
                dropState = 0
                return result
            }
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (dropState == -1)
                drop()
            return dropState == 1 || iterator.hasNext()
        }
    }

    internal

    class DistinctSequence<T, K>(private val source: Sequence<T>,
    private val keySelector: (T) -> K) : Sequence<T> {
        override fun iterator(): Iterator<T> =
            DistinctIterator(source.iterator(), keySelector)
    }

    private class DistinctIterator<T, K>(private val source:
        Iterator<T>, private val keySelector: (T) -> K) : AbstractIterator<T>() {
        private val observed =
            HashSet<K>()

        override fun computeNext() {
            while (source.hasNext()) {
                val next =
                    source.next()
                val key = keySelector(next)

                if (observed.add(key)) {
                    setNext(next)
                    return
                }
            }

            done()
        }

        override fun hasNext(): Boolean {
            return !skip && (!observed.contains(nextItem) && nextItem != null)
        }
    }

    private class GeneratorSequence<T : Any>(private val
        getInitialValue: () -> T?, private val getNextValue: (T) -> T?) : Sequence<T> {
        override fun iterator(): Iterator<T> =
            object : Iterator<T> {
                var nextItem: T? = null
                var nextState: Int = -2 // -2 for initial unknown, -1 for next unknown, 0 for done, 1 for continue

                private fun calcNext() {
                    nextItem = if (nextState == -2) getInitialValue() else getNextValue(nextItem!!)
                    nextState = if (nextState == 0) 0 else
                        if (nextState == -1) return
                    return
                }

                override fun next(): T {
                    if (nextState < 0)
                        calcNext()

                    if (nextState == 0)
                        throw NoSuchElementException()
                    val result = nextItem as T
                    // Do not clean nextItem (to avoid keeping reference on yielded instance) -- need to keep state for getNextValue
                    nextState = -1
                    return result
                }

                override fun hasNext(): Boolean {
                    if (nextState < 0)
                        calcNext()
                    return nextState == 1
                }
            }
    }

    /**
     * Returns a wrapper sequence that provides
     * values of this sequence, but ensures it can be iterated only one time.
     * The operation is _intermediate_ and _stateless_.
     * * @throws IllegalStateException on iterating the returned sequence from the second time.
     * * @throws NoSuchElementException on iterating the returned sequence from the second time.
     * * @see kotlin.sequences.sequence
     * * @sample samples.collections.Sequences.Building.generateSequence
     */
    public fun <T> Sequence<T>.constrainOnce(): Sequence<T> {
        return this as?
        ConstrainedOnceSequence<T> ?: ConstrainedOnceSequenceImpl(this)
    }

    /**
     * Returns a sequence which
     * invokes the function to calculate the next value on each iteration until it returns `null`.
     * If the sequence produces values until it encounters first `null` value.
     * The sequence can be iterated multiple times, each time
     * starting with [seed].
     * * @see kotlin.sequences.sequence
     * * @sample samples.collections.Sequences.Building.generateSequenceWithSeed
     */
    public fun <T : Any>
        generateSequence(seed: T?, nextFunction: (T) -> T?): Sequence<T> =
            GeneratorSequence(seed, nextFunction)

    /**
     * Returns a sequence defined by the function [seedFunction], which is invoked to produce the starting value,
     * and the function [nextFunction], which is invoked to calculate the next value based on the previous one on each iteration.
     * The sequence produces values until it encounters first `null` value.
     * The sequence can be iterated multiple times, each time
     * starting with [seed].
     * * @see kotlin.sequences.sequence
     * * @Sample
     */
    public fun <T : Any>
        generateSequenceWithSeed(
            seedFunction: () -> T?,
            nextFunction: (T) -> T?): Sequence<T> =
            GeneratorSequence(seedFunction, nextFunction)

    /*
     * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
     * Use of this source code is governed by the Apache 2.0 license that can be
     * found in the license/LICENSE.txt file.
     */

package kotlin

import kotlin.contracts.contract

/**
 * Throws an \[IllegalArgumentException\] if the \[value\] is false.
 * @sample samples.misc.Preconditions.failRequireWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun require(value: Boolean): Unit {
    contract {
        returns() implies value
    }
    require(value) { "Failed requirement.\" }
}

/**
 * Throws an \[IllegalArgumentException\] with the result of calling
 * \[lazyMessage\] if the \[value\] is false.
 * @sample samples.misc.Preconditions.failRequireWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun require(value: Boolean, lazyMessage: () -> Any): Unit {
    contract {
        returns() implies value
    }
    if (!value) {
        val message = lazyMessage()
        throw IllegalArgumentException(message.toString())
    }
}

/**
 * Throws an \[IllegalArgumentException\] if the \[value\] is null. Otherwise \[returns\] the not null value.
 * @sample samples.misc.Preconditions.failNotNullWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun requireNotNull(value: T?): T {
    contract {
        returns() implies (value != null)
    }
    return requireNotNull(value) { "Required value was null.\" }
}

/**
 * Throws an \[IllegalArgumentException\] with the result of calling \[lazyMessage\] if the \[value\] is null. Otherwise \[returns\] the not null value.
 * @sample samples.misc.Preconditions.failNotNullWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun requireNotNull(value: T?, lazyMessage: () -> Any): T {
    contract {
        returns() implies (value != null)
    }
    if (value == null) {
        val message = lazyMessage()
        throw IllegalStateException(message.toString())
    } else {
        return value
    }
}

/**
 * Throws an \[IllegalStateException\] if the \[value\] is false.
 * @sample samples.misc.Preconditions.failCheckWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun check(value: Boolean): Unit {
    contract {
        returns() implies value
    }
    check(value) { "Check failed.\" }
}

/**
 * Throws an \[IllegalStateException\] with the result of calling \[lazyMessage\] if the \[value\] is false.
 * @sample samples.misc.Preconditions.failCheckWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun check(value: Boolean, lazyMessage: () -> Any): Unit {
    contract {
        returns() implies value
    }
    if (!value) {
        val message = lazyMessage()
        throw IllegalStateException(message.toString())
    }
}

/**
 * Throws an \[IllegalStateException\] if the \[value\] is null. Otherwise \[returns\] the not null value.
 * @sample samples.misc.Preconditions.failCheckNotNullWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun checkNotNull(value: T?): T {
    contract {
        returns() implies (value != null)
    }
    return checkNotNull(value) { "Required value was null.\" }
}

/**
 * Throws an \[IllegalStateException\] with the result of calling \[lazyMessage\] if the \[value\] is null. Otherwise \[returns\] the not null value.
 * @sample samples.misc.Preconditions.failCheckNotNullWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun checkNotNull(value: T?, lazyMessage: () -> Any): T {
    contract {
        returns() implies (value != null)
    }
    if (value == null) {
        val message = lazyMessage()
        throw IllegalStateException(message.toString())
    } else {
        return value
    }
}

/**
 * Throws an \[IllegalStateException\] with the given \[message\].
 * @sample samples.misc.Preconditions.failWithError
 */
@kotlin.internal.InlineOnly
public inline fun error(message: Any): Nothing = throw IllegalStateException(message.toString())

import kotlin.js.*
import primitiveArrayConcat
import withType
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns an element at the given \[index\] or throws an \[IndexOutOfBoundsException\] if the \[index\] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun <T> Array<out T>.elementAt(index: Int): T {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size\") }
}

/**
 * Returns an element at the given \[index\] or throws an \[IndexOutOfBoundsException\] if the \[index\] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun ByteArray.elementAt(index: Int): Byte {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size\") }
}
[index] or throws a [IndexOutOfBoundsException] if the [index] is out of bounds of this array.

```kotlin
public actual fun ShortArray.elementAt(index: Int): Short {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size\})
}
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
 * out of bounds of this array.
 * 
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun IntArray.elementAt(index: Int): Int {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size\})
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
 * out of bounds of this array.
 * 
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun LongArray.elementAt(index: Int): Long {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size\})
}
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is
 * out of bounds of this array.
 * 
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun FloatArray.elementAt(index: Int): Float {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size\})
}
}

/**
 * Returns a [List] that wraps the original array.
 */
public actual fun <T> Array<out T>.asList(): List<T> {
    return ArrayList<T>(this.unsafeCast<Array<Any?>>())
}

/**
 * Returns a [List] that wraps the original array.
 */
@kotlin.internal.InlineOnly
public actual inline fun ByteArray.asList(): List<Byte> {
    return this.unsafeCast<Array<Byte>>().asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
@kotlin.internal.InlineOnly
public actual inline fun ShortArray.asList(): List<Short> {
    return this.unsafeCast<Array<Short>>().asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
@kotlin.internal.InlineOnly
public actual inline fun IntArray.asList(): List<Int> {
    return this.unsafeCast<Array<Int>>().asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
@kotlin.internal.InlineOnly
public actual inline fun LongArray.asList(): List<Long> {
    return this.unsafeCast<Array<Long>>().asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
@kotlin.internal.InlineOnly
public actual inline fun FloatArray.asList(): List<Float> {
    return this.unsafeCast<Array<Float>>().asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
@kotlin.internal.InlineOnly
public actual inline fun DoubleArray.asList(): List<Double> {
    return this.unsafeCast<Array<Double>>().asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
@kotlin.internal.InlineOnly
public actual inline fun BooleanArray.asList(): List<Boolean> {
    return this.unsafeCast<Array<Boolean>>().asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
```
elements in the same order. If two corresponding elements are nested arrays, they are also compared deeply. If any of arrays contains itself on any nesting level the behavior is undefined. The elements of other types are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayDeepEquals\")*
    *public actual infix fun \(<T> Array<out T> .contentDeepEquals\(other: Array<out T\>): Boolean { definedExternally\n        \n        * Returns a hash code based on the contents of this array as if it is a [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level the behavior is undefined.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayDeepHashCode\")*
    *public actual fun \(<T> Array<out T>.contentDeepHashCode\(): Int { definedExternally\n        \n        * Returns a string representation of the contents of this array as if it is a [List]. Nested arrays are treated as lists too. If any of arrays contains itself on any nesting level that reference is rendered as \"[...\"] to prevent recursion.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayDeepToString\")*
    *public actual fun \(<T> Array<out T>.contentDeepToString\(): String { definedExternally\n        \n        * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayEquals\")*
    *public actual infix fun \(<T> Array<out T>.contentEquals\(other: Array<out T\>): Boolean { definedExternally\n        \n        * Returns `true` if the two specified arrays are equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayEquals\")*
    *public actual infix fun \(<T> ByteArray.contentEquals\(other: ByteArray\): Boolean { definedExternally\n        \n        * Returns `true` if the two specified arrays are equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayEquals\")*
    *public actual infix fun \(<T> ShortArray.contentEquals\(other: ShortArray\): Boolean { definedExternally\n        \n        * Returns `true` if the two specified arrays are equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayEquals\")*
    *public actual infix fun \(<T> IntArray.contentEquals\(other: IntArray\): Boolean { definedExternally\n        \n        * Returns `true` if the two specified arrays are equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayEquals\")*
    *public actual infix fun \(<T> LongArray.contentEquals\(other: LongArray\): Boolean { definedExternally\n        \n        * Returns `true` if the two specified arrays are equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

    *\@SinceKotlin(\"1.1\")*\n    *\@library(\"arrayEquals\")*
    *public actual infix fun \(<T> FloatArray.contentEquals\(other: FloatArray\): Boolean { definedExternally\n        \n        * Returns `true` if the two specified arrays are equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
- `@SinceKotlin("1.1")`\n  `@library("arrayEquals")`\n  `public actual infix fun DoubleArray.contentEquals(other: DoubleArray): Boolean {\n    definedExternally\n    \n    * Returns ‘true’ if the two specified arrays are structurally equal to one another, i.e. contain the same number of the same elements in the same order.\n    * The elements are compared for equality with the [equals][Any.equals] function.\n    * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayEquals")`\n  `public actual infix fun BooleanArray.contentEquals(other: BooleanArray): Boolean {\n    definedExternally\n    \n    * Returns ‘true’ if the two specified arrays are structurally equal to one another, i.e. contain the same number of the same elements in the same order.\n    * The elements are compared for equality with the [equals][Any.equals] function.\n    * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayEquals")`\n  `public actual infix fun CharArray.contentEquals(other: CharArray): Boolean {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun <T> Array<out T>.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun ByteArray.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun ShortArray.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun IntArray.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun LongArray.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun FloatArray.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun DoubleArray.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun BooleanArray.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayHashCode")`\n  `public actual fun CharArray.contentHashCode(): Int {\n    definedExternally\n    \n    * Returns a hash code based on the contents of this array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayToString")`\n  `public actual fun <T> Array<out T>.contentToString(): String {\n    definedExternally\n    \n    * Returns a string representation of the contents of the specified array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayToString")`\n  `public actual fun ByteArray.contentToString(): String {\n    definedExternally\n    \n    * Returns a string representation of the contents of the specified array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayToString")`\n  `public actual fun ShortArray.contentToString(): String {\n    definedExternally\n    \n    * Returns a string representation of the contents of the specified array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayToString")`\n  `public actual fun IntArray.contentToString(): String {\n    definedExternally\n    \n    * Returns a string representation of the contents of the specified array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayToString")`\n  `public actual fun LongArray.contentToString(): String {\n    definedExternally\n    \n    * Returns a string representation of the contents of the specified array as if it is [List].\n  }

- `@SinceKotlin("1.1")`\n  `@library("arrayToString")`\n  `public actual fun FloatArray.contentToString(): String {\n    definedExternally\n    \n    * Returns a string representation of the contents of the specified array as if it is [List].\n  }
Returns a string representation of the contents of the specified array as if it is
[]

public actual fun DoubleArray.contentToString(): String
{
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
[]

@SinceKotlin("1.1")
@library("arrayToString")
public actual fun DoubleArray.contentToString(): String
{
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
[]

@SinceKotlin("1.1")
@library("arrayToString")
public actual fun BooleanArray.contentToString(): String
{
    definedExternally
}

/**
 * Returns a string representation of the contents of the specified array as if it is
[]

@SinceKotlin("1.1")
@library("arrayToString")
public actual fun CharArray.contentToString(): String
{
    definedExternally
}

/**
 * Copies this array or its subrange into the [destination] array and returns that
array.

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset:
Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T>
{
    arrayCopy(this, destination, destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that
array.

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): ByteArray
{
    arrayCopy(this.unsafeCast<Array<Byte>>(), destination.unsafeCast<Array<Byte>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that
array.

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0,
startIndex: Int = 0, endIndex: Int = size): ShortArray
{
    arrayCopy(this.unsafeCast<Array<Short>>(), destination.unsafeCast<Array<Short>>(), destinationOffset, startIndex, endIndex)
    return destination
}
destination.unsafeCast<Array<Short>>()
    return destination

* Copies this array or its subrange into the [destination] array and returns that array.
* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

* @param destination the array to copy to.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`. This @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* An @SinceKotlin("1.3") inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray
  Copies this array or its subrange into the [destination] array and returns that array.
  It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

* @param destination the array to copy to.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

* An @SinceKotlin("1.3") inline fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray
  Copies this array or its subrange into the [destination] array and returns that array.
  It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

* @param destination the array to copy to.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

* An @SinceKotlin("1.3") inline fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray
  Copies this array or its subrange into the [destination] array and returns that array.
  It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

* @param destination the array to copy to.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
[@SinceKotlin("1.3")]
public actual inline fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray {
    arrayCopy(this.unsafeCast<Array<Double>>(), destination.unsafeCast<Array<Double>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the
destination range.
 * @param destination the array to copy to.
 * @param destinationOffset position in the [destination] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when `startIndex > endIndex`.
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
public actual inline fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {
    arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the
destination range.
 * @param destination the array to copy to.
 * @param destinationOffset position in the [destination] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when `startIndex > endIndex`.
 * @return the [destination] array.
 */
public actual fun LongArray.copyOf(): LongArray {
    return withType("LongArray", this.unsafeCast<Array<Long>>().asDynamic().slice(0, size))
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline fun ByteArray.copyOf(): ByteArray {
    return this.asDynamic().slice(0, size)
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline fun ShortArray.copyOf(): ShortArray {
    return this.asDynamic().slice(0, size)
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline fun IntArray.copyOf(): IntArray {
    return this.asList().slice(0, size)
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline fun LongArray.copyOf(): LongArray {
    return withType("LongArray", this.asList().slice(0, size)).asDynamic().slice(0, size)
}
actual inline fun FloatArray.copyOf(): FloatArray {
    return this.asDynamic().slice() as FloatArray
}

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample
 */
public actual inline fun DoubleArray.copyOf(): DoubleArray {
    return this.asDynamic().slice() as DoubleArray
}

actual inline fun FloatArray.copyOf(): FloatArray {
    return this.asDynamic().slice() as FloatArray
}

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample
 */
public actual inline fun DoubleArray.copyOf(): DoubleArray {
    return this.asDynamic().slice() as DoubleArray
}

actual inline fun FloatArray.copyOf(): FloatArray {
    return this.asDynamic().slice() as FloatArray
}

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample
 */
public actual inline fun DoubleArray.copyOf(): DoubleArray {
    return this.asDynamic().slice() as DoubleArray
}
a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with 'false' values if necessary. If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with 'false' values.

```kotlin
public actual fun BooleanArray.copyOf(newSize: Int): BooleanArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("BooleanArray", arrayCopyResize(this, newSize, false))
}
```

```kotlin
public actual fun CharArray.copyOf(newSize: Int): CharArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("CharArray", fillFrom(this, CharArray(newSize)))
}
```

```kotlin
public actual fun <T> Array<out T>.copyOf(newSize: Int): Array<T?> {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return arrayCopyResize(this, newSize, null)
}
```

```kotlin
public actual fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}
```

```kotlin
public actual fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}
```

```kotlin
public actual fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}
```

```kotlin
public actual fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return withType("LongArray", this.asDynamic().slice(fromIndex, toIndex))
}
```

```kotlin
public actual fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}
```
AbstractList.checkRangeIndexes(fromIndex, toIndex, size) return this.asDynamic().slice(fromIndex, toIndex)\n\n/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public actual fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size) return this.asDynamic().slice(fromIndex, toIndex)\n\n/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size) return withType("BooleanArray", this.asDynamic().slice(fromIndex, toIndex))\n\n/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive), must be in `0..array.size`
 * @param toIndex the end of the range (exclusive), must be in `fromIndex..array.size`
 */
public actual fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size) return withType("CharArray", this.asDynamic().slice(fromIndex, toIndex))\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size) this.asDynamic().fill(element, fromIndex, toIndex)\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size) this.asDynamic().fill(element, fromIndex, toIndex)\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size) this.asDynamic().fill(element, fromIndex, toIndex)\n\n@SinceKotlin("1.3")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size) this.asDynamic().fill(element, fromIndex, toIndex)
fromIndex the start of the range (inclusive), 0 by default.\n* @param toIndex the end of the range (exclusive), size of this array by default.\n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
/**
 * Fills this array or its subrange with the specified [element] value, 0 by default.\n* @param fromIndex the start of the range (inclusive), 0 by default.\n* @param toIndex the end of the range (exclusive), size of this array by default.\n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex, toIndex)\n}

/**
 * Fills this array or its subrange with the specified [element] value, 0 by default.\n* @param fromIndex the start of the range (inclusive), 0 by default.\n* @param toIndex the end of the range (exclusive), size of this array by default.\n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex, toIndex)\n}

/**
 * Fills this array or its subrange with the specified [element] value, 0 by default.\n* @param fromIndex the start of the range (inclusive), 0 by default.\n* @param toIndex the end of the range (exclusive), size of this array by default.\n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex, toIndex)\n}

/**
 * Fills this array or its subrange with the specified [element] value, 0 by default.\n* @param fromIndex the start of the range (inclusive), 0 by default.\n* @param toIndex the end of the range (exclusive), size of this array by default.\n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    this.asDynamic().fill(element, fromIndex, toIndex)\n}

/**
 * Returns an array containing all elements of the original array and then the given [element].\n
*/
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline operator fun <T> Array<out T>.plus(element: T): Array<T> {\n    return this.asDynamic().concat(arrayOf(element))\n}

/**
 * Returns an array containing all elements of the original array and then the given [element].\n
*/
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ByteArray.plus(element: Byte): ByteArray {\n    return plus(byteArrayOf(element))\n}

/**
 * Returns an array containing all elements of the original array and then the given [element].\n
*/
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ShortArray.plus(element: Short): ShortArray {\n    return plus(shortArrayOf(element))\n}
original array and then the given [element].
*/
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun IntArray.plus(element: Int): IntArray {
    return plus(intArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun LongArray.plus(element: Long): LongArray {
    return plus(longArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun FloatArray.plus(element: Float): FloatArray {
    return plus(floatArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun DoubleArray.plus(element: Double): DoubleArray {
    return plus(doubleArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun CharArray.plus(element: Char): CharArray {
    return plus(charArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline operator fun <T> Array<out T>.plus(elements: Collection<T>): Array<T> {
    return arrayPlusCollection(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ShortArray.plus(elements: Collection<Short>): ShortArray {
    return fillFromCollection(this.copyOf(size + elements.size), size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun IntArray.plus(elements: Collection<Int>): IntArray {
    return fillFromCollection(this.copyOf(size + elements.size), size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun LongArray.plus(elements: Collection<Long>): LongArray {
    return arrayPlusCollection(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun FloatArray.plus(elements: Collection<Float>): FloatArray {
    return fillFromCollection(this.copyOf(size + elements.size), size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray {
    return fillFromCollection(this.copyOf(size + elements.size), size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun CharArray.plus(elements: Collection<Char>): CharArray {
    return fillFromCollection(this.copyOf(size + elements.size), size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun <T> Array<out T>.plus(elements: Array<out T>) {
    return this.asDynamic().concat(elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ByteArray.plus(elements: ByteArray) {
    return primitiveArrayConcat(this, elements)
}
Open Source Used In DNAC Platform Antelope

```kotlin
// @Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ShortArray.plus(elements: ShortArray): ShortArray {
    return primitiveArrayConcat(this, elements)
}

// @Suppress("NOTHING_TO_INLINE")
public actual inline operator fun IntArray.plus(elements: IntArray): IntArray {
    return primitiveArrayConcat(this, elements)
}

// @Suppress("NOTHING_TO_INLINE")
public actual inline operator fun LongArray.plus(elements: LongArray): LongArray {
    return primitiveArrayConcat(this, elements)
}

// @Suppress("NOTHING_TO_INLINE")
public actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {
    return primitiveArrayConcat(this, elements)
}

// @Suppress("NOTHING_TO_INLINE")
public actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {
    return primitiveArrayConcat(this, elements)
}

// @Suppress("NOTHING_TO_INLINE")
public actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {
    return primitiveArrayConcat(this, elements)
}

// @Suppress("NOTHING_TO_INLINE")
public actual inline operator fun CharArray.plus(elements: CharArray): CharArray {
    return primitiveArrayConcat(this, elements)
}

// @Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline fun <T> Array<out T>.plusElement(element: T): Array<T> {
    return this.asDynamic().concat(arrayOf(element))
}

// @library("primitiveArraySort")
public actual fun IntArray.sort() {
    definedExternally
}

// @library("primitiveArraySort")
public actual fun LongArray.sort() {
    if (size > 1) sort {
        a: Long, b: Long -> a.compareTo(b)
    }
}

// @library("primitiveArraySort")
public actual fun ByteArray.sort() {
    definedExternally
}

// @library("primitiveArraySort")
public actual fun ShortArray.sort() {
    definedExternally
}

// @library("primitiveArraySort")
public actual fun DoubleArray.sort() {
    definedExternally
}

// @library("primitiveArraySort")
public actual fun FloatArray.sort() {
    definedExternally
}

// @library("primitiveArraySort")
public actual fun CharArray.sort() {
    definedExternally
}

// @sample samples.collections.Arrays.Sorting.sortArrayOfComparable
public actual fun <T : Comparable<T>> Array<out T>.sort() {
    if (size > 1) sort {
        a: T, b: T -> a.compareTo(b)
    }
}
```

---

*An@kotlin.internal.InlineOnly/inline fun ByteArray.sort(noinline comparison: (a: Byte, b: Byte) -> Int):
Unit {
    asDynamic().sortWith(ByteArrayComparator(), comparison)
}*

*An@kotlin.internal.InlineOnly/inline fun ShortArray.sort(noinline comparison: (a: Short, b: Short) -> Int):
Unit {
    asDynamic().sortWith(ShortArrayComparator(), comparison)
}*

---

7864
comparison: (a: Short, b: Short) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
public inline fun IntArray.sort(noinline comparison: (a: Int, b: Int) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
public inline fun LongArray.sort(noinline comparison: (a: Long, b: Long) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparison] function.
 */
public inline fun CharArray.sort(noinline comparison: (a: Char, b: Char) -> Int): Unit {
    asDynamic().sort(comparison)
}

/**
 * Compares two values using the specified function [selector] to calculate the result of the comparison.
 * The function is applied to the given values [a] and [b] and return a [Comparable] instance.
 * The result of comparison of these [Comparable] instances is returned.
 */
@kotlin.internal.InlineOnly
public inline fun <T> compareValuesBy(a: T, b: T, selector: (T) -> *ComparisonKt*
Comparable<*?>: Int {\n    return compareValues(selector(a), selector(b)\n}\n
/**
 * Compares two values using the specified [selector] function to calculate the result of the comparison.\n * The function is applied to the given values [a] and [b] and return objects of type K which are then being\n * compared with the given [comparator].\n *\n * @fun sample samples.comparisons.Comparisons.compareValuesWithByWithComparator\n *\n *\n
// Not so useful without\n// type inference for receiver of expression\n// compareValuesWith(v1, v2, compareBy { it.prop1 }\nthenByDescending { it.prop2 }\n)\n
/**
 * Compares two nullable [Comparable] values. Null is considered less than any value.\n *\n * @fun samples.comparisons.Comparisons.compareValues\n */
fun <T : Comparable<*>> compareValues(a: T?, b: T?): Int {
    if (a === b) return 0
    if (a == null) return -1
    if (b == null) return 1

    @Suppress("UNCHECKED_CAST")
    return (a as Comparable<Any>).compareTo(b)
}

/**
 * Creates a comparator using the sequence of functions to calculate a result of comparison.\n * The functions are called sequentially, receive the given values `a` and `b` and return\n * objects of type K which are then being\n * compared with the given [comparator].\n *\n * @fun sample samples.comparisons.Comparisons.compareValuesWithByWithComparator\n */
fun <T> compareBy(vararg selectors: (T)\n-> Comparable<*?>): Comparator<T> {
    require(selectors.size > 0)
    return Comparator { a, b ->
        compareValuesByImpl(a, b, selectors) }
}

/**
 * Creates a comparator using the function to transform value to a [Comparable] instance for comparison.\n *\n * @fun samples.comparisons.Comparisons.compareByWithSingleSelector\n */
public inline fun <T> compareBy(crossinline selector: (T)\n-> Comparable<*?>): Comparator<T> =
    Comparator { a, b ->
        compareValuesBy(a, b, selector) }

/**
 * Creates a comparator using the [selector] function to transform values being compared and then applying\n * the specified [comparator] to compare transformed values.\n *\n * Note that an order of [comparator] is\n * reversed by this wrapper.\n */
public inline fun <T, K> Comparator<T>.thenBy(comparator: Comparator<in K>,\ncrossinline selector: (T)\n-> Comparable<*?>): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenBy.compare(a, b)\n        if (previousCompare != 0) previousCompare else\n            compareValuesBy(a, b, selector)\n    }

/**
 * Creates a descending comparator using the function to transform value to a [Comparable] instance for comparison.\n *\n * @fun samples.comparisons.Comparisons.compareByDescendingWithSingleSelector\n */
public inline fun <T> Comparator<T>.thenByDescending(crossinline selector: (T)\n-> Comparable<*?>): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenBy.compare(a, b)\n        if (previousCompare != 0) previousCompare else\n            compareValuesBy(a, b, selector)\n    }

/**
 * Creates a descending comparator using the [selector] function to transform values and then compares them with\n * the given [comparator].\n *\n * @fun samples.comparisons.Comparisons.compareByDescendingWithComparator\n */
public inline fun <T, K> Comparator<T>.thenByDescending(comparator: Comparator<in K>,\ncrossinline selector: (T)\n-> K): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenBy.compare(a, b)\n        if (previousCompare != 0) previousCompare else\n            compareValuesBy(a, b, selector)\n    }

// Not so useful without\n// type inference for receiver of expression\n// compareValuesWith(v1, v2, compareBy { it.prop1 }\nthenByDescending { it.prop2 }\n)\n
/**
 * Compares two values using the specified [selector] function to calculate the result of the comparison.\n * The function is applied to the given values [a] and [b] and return objects of type K which are then being\n * compared with the given [comparator].\n *\n * @fun sample samples.comparisons.Comparisons.compareValuesByWithComparator\n */
fun <T, K> compareValuesBy(a: T, b: T, comparator: Comparator<in K>,\nselector: (T)\n-> K): Int {\n    return compareValues(selector(a), selector(b))\n}
this@thenBy.compare(a, b)
        if (previousCompare != 0) previousCompare else compareValuesBy(a, b, comparator, selector)
    }

/**
 * Creates a descending comparator using the primary comparator and a function to transform value to a 
 * [Comparable] instance for comparison.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Comparator<T>.thenByDescending(comparator: Comparator<in T>, crossinline selector: (T) -> Comparable<*>?): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenByDescending.compare(a, b)
        if (previousCompare != 0) previousCompare else compareValuesBy(b, a, selector)
    }

/**
 * Creates a comparator using the primary comparator and function to calculate a result of comparison.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Comparator<T>.thenComparator(crossinline comparison: (a: T, b: T) -> Int): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenComparator.compare(a, b)
        if (previousCompare != 0) previousCompare else comparison(a, b)
    }

/**
 * Combines this comparator and the given comparator such that the latter is applied only when the former
 * considered values equal.
 */
public infix fun <T> Comparator<T>.then(comparator: Comparator<in T>): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@then.compare(a, b)
        if (previousCompare != 0) previousCompare else comparator.compare(a, b)
    }

/**
 * Extends the given comparator of non-nullable values to a comparator of nullable values considering `null` value less than any other value.
 */
public fun <T : Any> nullsFirst(comparator: Comparator<in T>): Comparator<T?> =
    Comparator { a, b ->
        when {
            a === b -> 0
            a == null -> -1
            b == null -> 1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Provides a comparator of nullable [Comparable] values considering `null` value less than any other value.
 */
@kotlin.internal.InlineOnly
public inline fun <T : Comparable<T>> nullsFirst(): Comparator<T?> =
    nullsFirst(naturalOrder())

/**
 * Extends the given comparator of non-nullable values to a comparator of nullable values considering `null` value greater than any other value.
 */
public fun <T : Any> nullsLast(comparator: Comparator<in T>): Comparator<T?> =
    Comparator { a, b ->
        when {
            a === b -> 0
            a == null -> 1
            b == null -> -1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Provides a comparator of nullable [Comparable] values considering `null` value greater than any other value.
 */
@kotlin.internal.InlineOnly
public inline fun <T : Comparable<T>> nullsLast(): Comparator<T?> =
    nullsLast(naturalOrder())

/**
 * Returns a comparator that compares [Comparable] objects in natural order.
 */
public fun <T : Comparable<T>> naturalOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as Comparator<T>)
Comparable<T> reverseOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST") (ReverseOrderComparator as Comparator<T>)

@Suppress("UNCHECKED_CAST")
Comparator<T>.reversed(): Comparator<T> = when (this) {
  is ReversedComparator -> this.comparator
  NaturalOrderComparator -> @Suppress("UNCHECKED_CAST") (ReverseOrderComparator as Comparator<T>)
  ReverseOrderComparator -> @Suppress("UNCHECKED_CAST")
}

Comparator<T>(public val comparator: Comparator<T>): Comparator<T> { override fun compare(a: T, b: T): Int = comparator.compare(b, a)
}

Comparator<Comparable<T>> reverseOrder(): Comparator<T> = @Suppress("UNCHECKED_CAST")

/**
 * Returns a comparator that imposes the reverse ordering of this comparator.
 * @sample samples.comparisons.Comparisons.reversed
 */
public fun <T> Comparator<T>.reversed(): Comparator<T> = when (this) {
  is ReversedComparator -> this.comparator
  NaturalOrderComparator -> @Suppress("UNCHECKED_CAST") (ReverseOrderComparator as Comparator<T>)
  ReverseOrderComparator -> @Suppress("UNCHECKED_CAST")
}

private class ReversedComparator<T>(public val comparator: Comparator<T>) : Comparator<T> {
    override fun compare(a: T, b: T): Int = comparator.compare(b, a)
    @Suppress("VIRTUAL_MEMBER_HIDDEN")
    fun reversed(): Comparator<T> = comparator
}

private object NaturalOrderComparator : Comparator<Comparable<Any>> {
    override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)
    @Suppress("VIRTUAL_MEMBER_HIDDEN")
    fun reversed(): Comparator<Comparable<Any>> = ReverseOrderComparator
}

private object ReverseOrderComparator : Comparator<Comparable<Any>> {
    override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)
    @Suppress("VIRTUAL_MEMBER_HIDDEN")
    fun reversed(): Comparator<Comparable<Any>> = NaturalOrderComparator
}

package kotlin

public actual interface Comparator<T> {
    @JsName("compare")
    actual fun compare(a: T, b: T): Int
}

public actual inline fun <T> Comparator(crossinline comparison: (a: T, b: T) -> Int): Comparator<T> = object : Comparator<T> {
    override fun compare(a: T, b: T): Int = comparison(a, b)
}

/**
 * An exception is thrown to indicate that a method body remains to be implemented.
 */
public class NotImplementedError(message: String = "An operation is not implemented.") : Error(message)

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(): Nothing = throw NotImplementedError()

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 * @param reason a string explaining why the implementation is missing.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(reason: String): Nothing = throw NotImplementedError("An operation is not implemented: $reason")

/**
 * Calls the specified function [block] and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <R> run(block: () -> R): R {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return block()
}

/**
 * Calls the specified function [block] with `this` value as its receiver and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R> T.run(block: T.() -> R): R {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return block()
}

/**
 * Calls the specified function [block] with the given [receiver] as its receiver and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R> with(receiver: T, block: T.() -> R): R {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return receiver.block()
}

/**
 * Calls the specified function [block] with `this` value as its argument and returns `this` value.
 */
@kotlin.internal.InlineOnly
public inline fun <T> T.apply(block: T.() -> Unit): T {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return block()
}

Open Source Used In DNAC Platform Antelope 7988
* For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#also).

```kotlin
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
public inline fun <T> T.also(block: (T) -> Unit): T {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    block(this)
    return this
}

/**
 * Calls the specified function [block] with `this` value as its argument and returns its result.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
public inline fun <T> T.let(block: (T) -> R): R {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return block(this)
}

/**
 * Returns `this` value if it satisfies the given [predicate] or `null`, if it doesn't.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
public inline fun <T> T.takeIf(predicate: (T) -> Boolean): T? {
    contract {
        callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)
    }
    return if (predicate(this)) this else null
}

/**
 * Returns `this` value if it does not satisfy the given [predicate] or `null`, if it does.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
public inline fun <T> T.takeUnless(predicate: (T) -> Boolean): T? {
    contract {
        callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)
    }
    return if (!predicate(this)) this else null
}

/**
 * Executes the given function [action] specified number of [times].
 *
 * A zero-based index of current iteration is passed as a parameter to [action].
 *
 * @sample samples.misc.ControlFlow.repeat
 */
@kotlin.internal.InlineOnly
public inline fun repeat(times: Int, action: (Int) -> Unit) {
    contract { callsInPlace(action) }

    for (index in 0 until times) {
        action(index)
    }
}
```

---

```kotlin
import kotlin.js.*

/**
 * Returns the greater of two values.
 * If values are equal, returns the first one.
 */
@SinceKotlin("1.1")
public actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Byte, b: Byte): Byte {
    return Math.max(a.toInt(), b.toInt()).unsafeCast<Byte>()
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.Inline.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Short, b: Short): Short {
    return Math.max(a.toInt(), b.toInt()).unsafeCast<Short>()
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.Inline.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Int, b: Int): Int {
    return Math.max(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.Inline.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Long, b: Long): Long {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.Inline.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Float, b: Float): Float {
    return Math.max(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.Inline.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Double, b: Double): Double {
    return Math.max(a, b)
}
```

inline fun maxOf(a: Short, b: Short, c: Short): Short {
    return Math.max(a.toInt(), b.toInt(),
    c.toInt()).unsafeCast<Short>() * Returns the greater of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun maxOf(a: Int, b: Int, c: Int): Int {
    return Math.max(a, b, c) * Returns the greater of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun maxOf(a: Long, b: Long, c: Long): Long {
    return maxOf(a, maxOf(b, c)) * Returns the greater of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun maxOf(a: Float, b: Float, c: Float): Float {
    return Math.max(a, b, c) * Returns the greater of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun maxOf(a: Double, b: Double, c: Double): Double {
    return Math.max(a, b, c) * Returns the greater of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Byte, b: Byte): Byte {
    return Math.min(a.toInt(), b.toInt()).unsafeCast<Byte>() * Returns the smaller of two values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Short, b: Short): Short {
    return Math.min(a.toInt(), b.toInt()).unsafeCast<Short>() * Returns the smaller of two values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Int, b: Int): Int {
    return Math.min(a, b) * Returns the smaller of two values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR"),\"NOTHING_TO_INLINE\"public actual inline fun minOf(a: Long, b: Long): Long {
    return a <= b a else b * Returns the smaller of two values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Byte, b: Byte, c: Byte): Byte {
    return Math.min(a.toInt(), b.toInt(),
    c.toInt()).unsafeCast<Byte>() * Returns the smaller of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Short, b: Short, c: Short): Short {
    return Math.min(a.toInt(), b.toInt(),
    c.toInt()).unsafeCast<Short>() * Returns the smaller of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Int, b: Int, c: Int): Int {
    return Math.min(a, b, c) * Returns the smaller of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Long, b: Long, c: Long): Long {
    return Math.min(a, b, c) * Returns the smaller of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Float, b: Float, c: Float): Float {
    return Math.min(a, b, c) * Returns the smaller of three values.
}@SinceKotlin("1.1")@kotlin.internal.InlineOnly@Suppress("DEPRECATION_ERROR")public actual inline fun minOf(a: Double, b: Double, c: Double): Double {
    return Math.min(a, b, c) * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors. * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("\"CollectionsKt\"\")

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns 1st *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 1.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component1(): T {
    return get(0)
}

/**
 * Returns 2nd *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 2.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component2(): T {
    return get(1)
}

/**
 * Returns 3rd *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 3.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component3(): T {
    return get(2)
}

/**
 * Returns 4th *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 4.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component4(): T {
    return get(3)
}

/**
 * Returns 5th *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 5.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component5(): T {
    return get(4)
}

/**
 * Returns `true` if [element] is found in the collection.
 * 
 */
public operator fun <T> Iterable<T>.contains(element: T): Boolean {
    if (this is Collection)
        return contains(element)
    return indexOf(element) >= 0
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this collection.
 */
public fun <T> Iterable<T>.elementAt(index: Int): T {
    if (this is List)
        return get(index)
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("Collection doesn't contain element at index $index.") }
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this collection.
 */
@kotlin.internal.InlineOnly
public inline fun <T> List<T>.elementAt(index: Int): T {
    return get(index)
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this list.
 */
public fun <T> Iterable<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {
    if (this is List)
        return this.getOrElse(index, defaultValue)
    if (index < 0)
        return defaultValue(index)
    val iterator = iterator()
    var count = 0
    while (iterator.hasNext()) {
        val element = iterator.next()
        if (index == count++)
            return element
    }
    return defaultValue(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this collection.
 */
@kotlin.internal.InlineOnly
public inline fun <T> List<T>.elementAtOrNull(index: Int): T? {
    return getOrNull(index)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.find(predicate: (T) -> Boolean): T? {
    return firstOrNull(predicate)
}

Open Source Used In DNAC Platform Antelope 7991
Open Source Used In DNAC Platform Antelope
throw NoSuchElementException("Collection is empty.")
var last = iterator.next()
while (iterator.hasNext())
    last = iterator.next()
return last

/**
 * Returns the last element matching the given [predicate].
 */
@Suppress("UNCHECKED_CAST")
public inline fun <T> List<T>.lastOrNull(predicate: (T) -> Boolean): T?
    val iterator = this.listIterator(size)
    while (iterator.hasPrevious())
        val element = iterator.previous()
        if (predicate(element)) return element
    return null

/**
 * Returns the last element matching the given [predicate].
 */
public fun <T> List<T>.lastOrNull(): T?
    return this.lastIndexOrNull()
Returns a list containing first elements satisfying the given [predicate].

```kotlin
sample.samples.collections.Collections.Transformations.take

// An public inline fun <T> Iterable<T>.takeWhile(predicate: (T) -> Boolean): List<T> { 
  // val array = ArrayList<T>() 
  // for (item in this) { 
  //   if (!predicate(item)) { 
  //     return array
  //   } 
  //   array.add(item)
  // } 

  // Returns a list with elements in reversed order.
  // An public fun <T> Iterable<T>.reversed(): List<T> { 
  //   return toList().reversed()
  // }

  // Sorts elements in the list in-place according to natural sort order of the value returned by specified [selector] function.
  // The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
  // An public inline fun <T> MutableList<T>.sortBy(selector: (T) -> R): Unit

  // Sorts elements in the list in-place according to their natural sort order.
  // The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
  // An public inline fun <T> MutableList<T>.sortDescending(): Unit

  // Returns a list of all elements sorted according to their natural sort order.
  // The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
  // An public inline fun <T>: Collection<T>.sorted(): List<T> {
  //   return toList().sorted()
  // }

  // Returns a list of all elements sorted according to the specified [comparator].
  // The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
  // An public inline fun <T>: Collection<T>.sortedWith(comparator: Comparator<in T>): List<T> {
  //   return toList().sortedWith(comparator)
  // }
```
all of the elements of this collection.

/**
 * Returns an array of Int containing all of the elements of this collection.
 */
public fun Collection<Int>.toIntArray(): IntArray {
    val result = IntArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Long containing all of the elements of this collection.
 */
public fun Collection<Long>.toLongArray(): LongArray {
    val result = LongArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Short containing all of the elements of this collection.
 */
public fun Collection<Short>.toShortArray(): ShortArray {
    val result = ShortArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns a [Map] containing key-value pairs provided by [transform] function
 * applied to elements of the given collection.
 * If any of two pairs would have the same key the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 * @sample samples.collections.Collections.Transformations.associate
 */
public inline fun <T, K, V> Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

/**
 * Returns a [Map] containing the elements from the given collection indexed
 * by the key * returned from [keySelector] function applied to each element.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 * @sample samples.collections.Collections.Transformations.associateBy
 */
public inline fun <T, K> Iterable<T>.associateBy(keySelector: (T) -> K): Map<K, T> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and
 * indexed by [keySelector] functions applied to elements of the given collection.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 * @sample samples.collections.Collections.Transformations.associateByWithValueTransform
 */
public inline fun <T, K, V> Iterable<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Populates and returns the [destination] mutable map with
 * key-value pairs, where key is provided by the [keySelector] function applied to each element of the given collection.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 * @sample samples.collections.Collections.Transformations.associateByTo
 */
public inline fun <T, K, M : MutableMap<in K, in T>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with
 * key-value pairs provided by [transform] function applied to each element of the given collection.
 * If any of two pairs would have the same key the last one gets added to the map.
 * Returns a [Map] where keys are
 * */

elements from the given collection and values are让您* produced by the [valueSelector] function applied to each element.您* *您* If any two elements are equal, the last one gets added to the map.您* *您* The returned map preserves the entry iteration order of the original collection.您* *您* @sample
samples.collections.Collections.Transformations.associateWith(您* @SinceKotlin("1.3.1").)public inline fun <K, V> Iterable<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {您* val result = LinkedHashMap<K, V>(mapCapacity(collectionSizeOrDefault(10))).coerceAtLeast(16))您* return associateWithTo(result, valueSelector)您* *您* }}
@SinceKotlin("1.3")
public inline fun <K, V> Iterable<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {您* for (element in this) {您* destination.put(element, valueSelector(element))您* }您* return destination��* *您* }}

public fun <T, C : MutableCollection<in T>> Iterable<T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination��* *您* }}

public fun <T> Iterable<T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>(mapCapacity(collectionSizeOrDefault(12))))��* *您* }

public fun <T> Iterable<T>.toList(): List<T> {
    if (this is Collection) {
        return when (size) {
            0 -> emptyList()
            1 -> listOf(if (this is List) get(0) else iterator().next())��* else -> this.toMutableList().Bạn* *
            0 -> emptyList()��* 1 -> setOf(if (this is List) get(0) else iterator().next())��* else -> this.toMutableList().Bạn* *
        }
    }
    return this.toMutableList().optimizeReadOnlyList��* *Bạn* }}

public inline fun <T, R> Iterable<T>.flatMap(transform: (T) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform��* *Bạn* }}

public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)��* return toCollection(ArrayList<T>(), iterator().next())).optimizeReadOnlyList()��* *
        destination.addAll(list��* *
    }
    return destination��* *Bạn* }}

public inline fun <T, K> Iterable<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector��* *
    * Vous* *
    * Vous* * Vous* *
    * Vous* * Vous* *
}
of the original collection by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n\n\n@sample samples.collections.Collections.Transformations.sortBy

Iterable<T>.sortBy(\n    destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n    val key = keySelector(element)\n    val list = destination.getOrPut(key) {\n        ArrayList<V>() }\n    list.add(element)\n    return destination\n}\n
\n/**
 * Groups values returned by the given [valueTransform] function applied to each element of the original collection
 * by the key returned by the given [keySelector] function applied to the element and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <T, K, V, M : MutableMap<in K, MutableList<V>>> Iterable<T>.groupByValues(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrPut(key) {\n            ArrayList<V>() }\n        list.add(valueTransform(element))\n    }\n    return destination
}

/**
 * Creates a [Grouping] source from a collection to be used later with one of group-and-fold operations
 * using the specified [keySelector] function to extract a key from each element.
 * @param [keySelector] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * @sample samples.collections.Groups.groupingByEachCount
 */
@SinceKotlin("1.1")
public inline fun <T, K> Iterable<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n    return object : Grouping<T, K> {\n        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n        override fun keyOf(element: T): K = keySelector(element)
    }
}

/**
 * Returns a list containing the results of applying the given [transform] function to each element in the original collection.
 * @sample samples.collections.Collections.Transformations.map
 */
public inline fun <T, R> Iterable<T>.map(transform: (T) -> R): List<R> {\n    return mapTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

/**
 * Returns a list containing the results of applying the given [transform] function to each element and its index in the original collection.
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * @sample samples.collections.Collections.Trans-formations.mapIndexed
 */
public inline fun <T, R> Iterable<T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {\n    return mapIndexedTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

/**
 * Applies the given [transform] function to each element and its index in the original collection
 * and appends only the non-null results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * @sample samples.collections.Collections.Transformations.mapIndexedNotNull
 */
public inline fun <T, R : Any> Iterable<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {\n    return mapIndexedNotNullTo(ArrayList<R>(), transform)
}

/**
 * Applies the given [transform] function to each element and its index in the original collection
 * and appends only the non-null results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * @sample samples.collections.Collections.Transformations.mapIndexedNotNull
 */
public inline fun <T, R : Any, C : MutableCollection<in R>> Iterable<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n    forEachIndexed { index, element -> destination.add(transform(index, element)) }\n    return destination
}

/**
 * Applies the given [transform] function to each element and its index in the original collection
 * and appends only the non-null results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * @sample samples.collections.Collections.Transformations.mapIndexedNotNull
 */
public inline fun <T, R : Any> Iterable<T>.mapIndexedNotNull(transform: (T) -> R?): List<R> {\n    return mapIndexedNotNullTo(ArrayList<R>(), transform)
}
element -> transform(element)?.let { destination.add(it) } | return destination

Applies the given [transform] function to each element of the original collection and appends the results to the given destination.

`*`Applies an inline function `fun <T, R, C : MutableCollection<R>> Iterable<T>.mapTo(destination: C, transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

`*`Applies a lazy [Iterable] that wraps each element of the original collection into an [IndexedValue] containing the index of that element and the element itself.

`*`Applies a function `fun <T> Iterable<T>.withIndex(): Iterable<IndexedValue<T>> { `fun return indexingIterable { iterator() }` |` return

Retains a list containing only distinct elements from the given collection.

`*`Returns a list containing only distinct elements from the given collection.

The elements in the resulting list are in the same order as they were in the source collection.

`*`Applies a function `fun <T> Iterable<T>.distinct(): List<T> { `fun return this.toMutableSet().toList()` |` return

Returns a list containing only elements from the given collection having distinct keys returned by the given [selector] function.

`*`The elements in the resulting list are in the same order as they were in the source collection.

`*`Applies a function `fun <T, K> Iterable<T>.distinctBy(selector: (T) -> K): List<T> { `fun set = HashSet<K>()
    `fun list = ArrayList<T>()
    for (e in this) {
        `fun key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

`*`Returns a set containing all elements that are contained by this collection and not contained by the specified collection.

`*`Returns a set containing all distinct elements from the given collection.

The returned set preserves the element iteration order of the original collection.

`*`To get a set containing all elements that are contained at least in one of these collections use [union].

`*`Applies a function `fun <T> Iterable<T>.intersect(other: Iterable<T>): Set<T> { `fun set = this.toMutableSet()
    `fun list = ArrayList<T>()
    for (e in this) {
        `fun key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

`*`Returns a set containing all elements that are contained in both collections.

`*`The returned set preserves the element iteration order of the original collection.

`*`To get a set containing all elements that are contained in both collections use [intersect].

`*`Returns the number of elements in this collection.

`*`Returns the number of elements in this collection.

`*`Returns the number of elements matching the given [predicate].

`*`Returns `true` if at least one element matches the given [predicate].

`*`Returns `true` if at least one element matches the given [predicate].

`*`Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator.
value and each element.

```kotlin
public inline fun <T, R> Iterable<T>.foldIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) {
        index++
        accumulator = operation(index, accumulator, element)
    }
    return accumulator
}
```

This method performs the given `[operation]` on each element of the collection, providing sequential index with the element. It takes the index of an element and the element itself as parameters and performs the desired action on the element. The method returns the final accumulator value.

Examples of using `foldIndexed()`:

- **Example 1:**
  ```kotlin
  val fruits = listOf("apple", "banana", "cherry")
  val orderedFruits = fruits
      .foldIndexed(0) { index, fruit, order -> order + 1 } // Increase order by 1 for each fruit
  println(orderedFruits) // Output: [1, 2, 3]
  ```

- **Example 2:**
  ```kotlin
  val numbers = listOf(1, 2, 3, 4, 5)
  val sum = numbers
      .foldIndexed(0) { index, number, sum -> sum + number } // Sum up all numbers
  println(sum) // Output: 15
  ```

These examples demonstrate how `foldIndexed()` can be used to accumulate values or perform computations on each element of a collection while preserving the index of each element.
= iterator().next() return null\n var min = iterator.next()\n if (min.isNaN()) return min\n while (iterator.hasNext()) \n { \n var e = iterator.next() \n if (e.isNaN()) return e \n if (min > e) min = e\n } \n return min\n }\n
Returns the smallest element or null if there are no elements

If any of elements is NaN returns NaN

\n
public fun Iterable<Float>.min(): Float? \n { \n val iterator = iterator() \n if (!iterator.hasNext()) return null\n var min = iterator.next() \n if (min.isNaN()) return min\n while (iterator.hasNext()) \n { \n var e = iterator.next() \n if (e.isNaN()) return e \n if (min > e) min = e\n } \n return min
}

/**
 * Returns the smallest element or null if there are no elements
 * (NaN) returns NaN.
 */
@SinceKotlin("1.1")
public fun Iterable<Float>.min(): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    if (min.isNaN()) return min
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (e.isNaN()) return e
        if (min > e) min = e
    }
    return min
}

/**
 * Returns the smallest element or null if there are no elements.
 */
public fun <T : Comparable<T>> Iterable<T>.min(): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minElem = iterator.next()
    if (!iterator.hasNext()) return minElem
    var minValue = selector(minElem)\n    do { \n        var e = iterator.next() \n        var v = selector(e) \n        if (minValue > v) { \n            minElem = e
            minValue = v
        }\n    } while (iterator.hasNext())\n    return minElem
}

/**
 * Returns `true` if the collection has no elements.
 */
public fun <T> Iterable<T>.none(): Boolean {
    if (this is Collection) return isEmpty()
    return !iterator().hasNext()\n}

/**
 * Performs the given action on each element and returns the collection itself afterwards.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <S, T> Iterable<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return null
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
    }
    return accumulator
}
starting with last element and applying [operation] from right to left to each element and current accumulator value.

* @param [operation] function that takes the index of an element, the element itself, and current accumulator value, and calculates the next accumulator value.

* @param [operation] function that takes the index of an element, current accumulator value, and the element itself, and calculates the next accumulator value.

* @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.

* @param [operation] function that takes the index of an element, current accumulator value, and the element itself, and calculates the next accumulator value.

* @param [operation] function that takes the index of an element, current accumulator value, and the element itself, and calculates the next accumulator value.
Iterable<T>.scanReduce(operation: (acc: S, T) -> S): List<S> {
    if (!iterator.hasNext()) return emptyList()
    var accumulator = operation(acc, iterator.next())
    result.add(accumulator)
    return result
}

Iterable<T>.scanReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    if (!iterator.hasNext()) return emptyList()
    var accumulator = operation(index, 0, iterator.next())
    result.add(accumulator)
    return result
}

Iterable<T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Iterable<T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Iterable<T>.requireNoNulls(): Iterable<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as Iterable<T>
}

Iterable<T>.chunked(size: Int): List<List<T>> {
    return windowed(size, size, partialWindows = true)
}

Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}

Iterable<T>.minus(element: T): List<T> {
    val result = ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}

Iterable<T>.scanReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    if (!iterator.hasNext()) return emptyList()
    var accumulator = operation(index, 0, iterator.next())
    result.add(accumulator)
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from
 * left to right to each element, its index in the original collection and current accumulator value
 * that starts with the first element of this collection.
 * @param [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the
 * element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scanReduce
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <S, T : S>
Iterable<T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are
 * any `null` elements.
 * @sample samples.collections.Collections.Transformations.chunked
 */
@SinceKotlin("1.2")
public fun <T> List<T?>.requireNoNulls(): List<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as List<T>
}

/**
 * Splits this collection into a list of
 * lists each not exceeding the given [size]
 * and applies the given [transform]
 * function to each list.
 * @param size the number of elements to take in each list, must be positive and can be greater
 * than the number of elements in this collection.
 * @sample samples.collections.Collections.Transformations.chunked
 */
@SinceKotlin("1.2")
public fun <T, R> List<T>.chunked(size: Int, transform: (List<T>) -> R): List<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}

/**
 * Returns a list
 * containing all elements of the original collection without the first occurrence of the given [element].
 * @param [operation] operator function
 * @sample samples.collections.Collections.Transformations.chunked
 */
@SinceKotlin("1.2")
public fun <T> ArrayList<T>(collectionSizeOrDefault(10)).apply { add(accumulator) }
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}

Iterable<T>.scanReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    if (!iterator.hasNext()) return emptyList()
    var accumulator = operation(index, 0, iterator.next())
    result.add(accumulator)
    return result
}

Iterable<T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Iterable<T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Iterable<T>.requireNoNulls(): Iterable<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as Iterable<T>
}

Iterable<T>.chunked(size: Int): List<List<T>> {
    return windowed(size, size, partialWindows = true)
}

Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}

Iterable<T>.minus(element: T): List<T> {
    val result = ArrayList<T>(collectionSizeOrDefault(10)).apply { add(accumulator) }
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}

Iterable<T>.scanReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    if (!iterator.hasNext()) return emptyList()
    var accumulator = operation(index, 0, iterator.next())
    result.add(accumulator)
    return result
}

Iterable<T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Iterable<T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Iterable<T>.requireNoNulls(): Iterable<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as Iterable<T>
}

Iterable<T>.chunked(size: Int): List<List<T>> {
    return windowed(size, size, partialWindows = true)
}

Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}

Iterable<T>.minus(element: T): List<T> {
    val result = ArrayList<T>(collectionSizeOrDefault(10)).apply { add(accumulator) }
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}

Iterable<T>.scanReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    if (!iterator.hasNext()) return emptyList()
    var accumulator = operation(index, 0, iterator.next())
    result.add(accumulator)
    return result
}

Iterable<T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Iterable<T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Iterable<T>.requireNoNulls(): Iterable<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as Iterable<T>
}

Iterable<T>.chunked(size: Int): List<List<T>> {
    return windowed(size, size, partialWindows = true)
}

Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}

Iterable<T>.minus(element: T): List<T> {
    val result = ArrayList<T>(collectionSizeOrDefault(10)).apply { add(accumulator) }
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}
implementation of `hashCode()` that doesn't change between successive invocations.

```kotlin
/**
 * Returns a list containing all elements of the original collection except the elements contained in the given [elements] collection.
 * The [elements] collection may be converted to a [HashSet] to speed up the operation, but the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 * @param elements The elements to remove.
 * @return The new list.
 */
public fun <T> List<T>.minus(elements: T): List<T> {
    return this.toSet().minus(elements)
}

/**
 * Returns a list containing all elements of the original collection except the elements contained in the given [elements] sequence.
 * The [elements] sequence may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 * @param elements The elements to remove.
 * @return The new list.
 */
public inline fun <T> Sequence<T>.minus(elements: Sequence<T>): List<T> {
    return this.toSet().minus(elements)
}

/**
 * Returns a list containing all elements of the original collection and then the given [element].
 * @param element The element to add.
 * @return The new list.
 */
public inline fun <T> List<T>.plus(element: T): List<T> {
    return this.toSet().plus(element)
}

/**
 * Returns a list containing all elements of the original collection and then the given [elements] array.
 * @param elements The elements to add.
 * @return The new list.
 */
public fun <T> List<T>.plus(elements: Array<T>): List<T> {
    return this.toSet().plus(elements)
}

/**
 * Returns a list containing all elements of the original collection and then the given [elements] collection.
 * @param elements The elements to add.
 * @return The new list.
 */
public fun <T> List<T>.plus(elements: Collection<T>): List<T> {
    return this.toSet().plus(elements)
}

/**
 * Returns a list containing all elements of the original collection and then the given [elements] sequence.
 * @param elements The elements to add.
 * @return The new list.
 */
public inline fun <T> Sequence<T>.plus(elements: Sequence<T>): List<T> {
    return this.toSet().plus(elements)
}
```

---

**Open Source Used In DNAC Platform Antelope 8005**
of the original collection and then the given [element].

```kotlin
Iterable<T>.plusElement(element: T): List<T> { 
    return plus(element)
} // *

Iterable<T>.plusElement(element: T): List<T> { 
    return plus(element)
} // **
```

```
@kotlin.internal.InlineOnly
public inline fun <T>
Iterable<T>.plusElement(element: T): List<T> {
    return plus(element)
}
```

```
/**
 * Returns a list containing all elements of the original collection and then the given 
 * [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> 
Collection<T>.plusElement(element: T): List<T> {
    return plus(element)
}
```

```
/**
 * Returns a list containing all elements of the original collection and then the given 
 * [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> 
Collection<T>.plusElement(element: T): List<T> {
    return plus(element)
}
```

```
/**
 * Returns a list of snapshots of the window of the given [size] sliding along this collection 
 * with the given [step], where each snapshot is a list.
 * Several last lists may have less elements than the given [size].
 * Both [size] and [step] must be positive and can be greater than the number of elements 
 * in this collection.
 */
@kotlin.internal.InlineOnly
public inline fun <T> 
    checkWindowSizeStep(size, step)
    if (this is RandomAccess && this is List) {
        val thisSize = this.size
        val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1
        val result = ArrayList<List<T>>(resultCapacity)
        var index = 0
        while (index in 0 until thisSize) {
            val windowSize = size.coerceAtMost(thisSize - index)
            if (windowSize < size && !partialWindows) break
            result.add(List(windowSize) { this[it + index] })
            index += step
        }
        return result
    }
    val result = ArrayList<List<T>>()
    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = false).forEach {
        result.add(it)
    }
    return result
}
```

```
/**
 * Returns a list of results of applying the given [transform] function to
 * an each list representing a view over the window of the given [size] sliding
 * along this collection with the given [step].
 * Several last lists may have less elements than the given [size].
 * Both [size] and [step] must be positive and can be greater than the number of elements 
 * in this collection.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R> 
Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): List<R> {
    checkWindowSizeStep(size, step)
    if (this is RandomAccess && this is List) {
        val thisSize = this.size
        val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1
        val result = ArrayList<R>(resultCapacity)
        val window = MovingSubList(this)
        var index = 0
        while (index in 0 until thisSize) {
            val windowSize = size.coerceAtMost(thisSize - index)
            if (!partialWindows && windowSize < size) break
            window.move(index, index + windowSize)
            result.add(transform(window))
            index += step
        }
        return result
    }
    val result = ArrayList<R>()
    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = true).forEach {
        result.add(transform(it))
    }
    return result
}
```

```
/**
 * Returns a list of pairs built from the elements of `this` collection and the 
 * [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R> 
Iterable<T>.zip(other: Array<out R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}
```

```
/**
 * Returns a list of values built from the elements of `this` collection and the [other] 
 * collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R, V>
Iterable<T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {
    val arraySize = other.size
    val result = ArrayList<V>(minOf(collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in this) {
        if (i >= arraySize) break
        result.add(transform(element, other[i++]))
        i += 1
    }
    return result
}
```

```
/**
 * Returns a list of pairs built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R, V> 
Iterable<T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {
    val arraySize = other.size
    val result = ArrayList<V>(minOf(collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in this) {
        if (i >= arraySize) break
        result.add(transform(element, other[i++]))
        i += 1
    }
    return result
}
```
public infix fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 ->
        t1 to t2 }
}

/**
 * Returns a list of values built from the elements of `this` collection and the [other] collection
 * with the same index * using the provided [transform] function applied to each pair of elements.
 */
void * The returned list has length of the shortest collection.

samples.collections.Iterations.zipIterableWithTransform
]

 Iterable<T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {
    val first = iterator()
    val second = other.iterator()
    val list = ArrayList<V>(minOf(collectionSizeOrDefault(10),
        other.collectionSizeOrDefault(10)))
    while (first.hasNext() && second.hasNext()) {
        list.add(transform(first.next(), second.next()))
    }
    return list
}

/**
 * Returns a list of pairs of each two
 * adjacent elements in this collection.
 */
void * The returned list is empty if this collection contains less than two elements.

samples.collections.CollectionsTransformations.zipWithNext

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] * elements will be appended, followed by the [truncated] string (which defaults to `\"...\"`).
 */
void * The returned list is empty if this collection contains less than two elements.

samples.collections.CollectionsTransformations.zipWithNextToFindDeltas

Iterable<T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1) buffer.append(separator)
        if (limit < 0 || count <= limit) {
            buffer.appendElement(element, transform)
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

Iterable<T>.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

Iterable<T>.asIterable(): Iterable<T> {
    return this
}

Iterable<T>.asSequence(): Sequence<T> {
    return Sequence { this.iterator() }
}

Iterable<Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return Double.NaN if count == 0 else sum / count
}

Iterable<Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return Double.NaN if count == 0 else sum / count
}
public fun Iterable<Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the collection.
 */
public fun Iterable<Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the collection.
 */
public fun Iterable<Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the collection.
 */
public fun Iterable<Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the collection.
 */
public fun Iterable<Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
public fun Iterable<Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
public fun Iterable<Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
public fun Iterable<Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
public fun Iterable<Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
public fun Iterable<Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the
 * Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

import kotlin.comparisons.naturalOrder
import kotlin.random.Random

/** Returns the array if it's not `null`, or an
 * empty array otherwise. */
@kotlin.internal.InlineOnly
public actual inline fun <T> Array<out T>? orEmpty(): Array<out T> = this ?: emptyArray<T>()

@kotlin.internal.InlineOnly
public actual inline fun <T> Collection<T>.toTypedArray(): Array<T> = copyToArray(this)

@JsName("copyToArray")
@PublishedApi
internal fun <T> copyToArray(collection: Collection<T>): Array<T> {
    return if (collection.asDynamic().toArray !== undefined)
        collection.asDynamic().toArray().unsafeCast<Array<T>>()
    else
        copyToArrayImpl(collection).unsafeCast<Array<T>>()
}

@JsName("copyToArrayImpl")
internal actual fun <T> copyToArrayImpl(collection: Collection<*>, array: Array<T>): Array<T> {
    if (array.size < collection.size)
        return copyToArrayImpl(collection).unsafeCast<Array<T>>()

    val iterator = collection.iterator()
    var index = 0
    while (iterator.hasNext()) {
        array[index++].unsafeCast<T>() = iterator.next()
    }
    if (index < array.size)
    {
        array[index].unsafeCast<T>() = null
    }
    return array
}

/**
 * Returns an immutable list
 * containing only the specified object 
 * [element].
 */
public fun <T> ListOf(element: T): List<T> = arrayListOf(element)

/**
 * Returns an immutable set containing only the specified object 
 * [element].
 */
public fun <T> SetOf(element: T): Set<T> = hashSetOf(element)

open fun <T> mapOf(pair: Pair<K, V>):
Map<K, V> = hashMapOf(pair)

/**
 * Fills the list with the provided [value].
 * Each element in the list
 * gets replaced with the [value].
 */
public actual fun <T> MutableList<T>.fill(value: T): Unit

/**
 * Randomly shuffles elements in
 * this list randomly shuffled.
 * The sort is _stable_. It means that equal elements preserve their
 * order relative to each other after sorting.
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.shuffle(): Unit = this.shuffle(Random)

/**
 * Returns a new list with the elements of this list
 * randomly shuffled.
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.sort(): Unit

/**
 * Sorts elements in the list in-place
 * according to their natural sort order.
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.sortWith(comparator: Comparator<in T>): Unit

private fun <T> collectionsSort(list: MutableList<T>, comparator: Comparator<in T>) {
    if (list.size <= 1) return

    val array = copyToArray(list)
    sortArrayWith(array, comparator)
    for (i in 0 until array.size) list[i] = array[i]
}

internal actual fun <T> arrayOfNulls(reference: Array<T>, size: Int): Array<T> {
    return arrayOfNulls<Any>(size).unsafeCast<Array<T>>()
}

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@PublishedApi
internal actual fun <T> arrayCopy(source: Array<out T>, destination: Array<in T>, destinationOffset: Int, startIndex: Int, endIndex: Int) {
    AbstractList.checkRangeIndexes(startIndex, endIndex, source.size)
    val rangeSize = endIndex - startIndex
    AbstractList.checkRangeIndexes(destinationOffset, destinationOffset + rangeSize, destination.size)
    if (js("ArrayBuffer").isView(destination) && js("ArrayBuffer").isView(source)) {
        destination.subarray(destinationOffset, destinationOffset + rangeSize) = source.subarray(startIndex, endIndex)
    } else {
        if (source !== destination || destinationOffset <= startIndex) {
            for (index in 0 until rangeSize) destination[destinationOffset + index] = source[startIndex + index]
        } else {
            for (index in rangeSize - 1 downTo 0) destination[destinationOffset + index] = source[startIndex + index]
        }
    }
}

@Suppress("NOTHING_TO_INLINE")
internal actual inline fun <K, V> Map<K, V>.toSingletonMapOrSelf(): Map<K, V> = this

@Suppress("NOTHING_TO_INLINE")
internal actual inline fun <K, V> Map<K, V>.toSingletonMap(): Map<K, V> = this.toMutableMap()

@PublishedApi
internal actual fun checkIndexOverflow(index: Int): Int {
    if (index < 0) throwIndexOverflow()
    return index
}

@PublishedApi
internal actual fun checkCountOverflow(count: Int): Int {
    if (count < 0) throwCountOverflow()
    return count
}

/**
 * Checks a collection builder function capacity argument.
 * In JS no validation is made in
 * Map/Set constructor yet.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@PublishedApi
internal actual fun checkBuilderCapacity(capacity: Int) {
    require(capacity >= 0) { "capacity must be non-negative." }
}

/**
 * Returns the given iterator itself.
 * Allows to use an instance of iterator in a
 * `for` loop.
 */
@sample samples.collections.Iterators.iterator

Iterator<IndexedValue<T>> = IndexingIterator(this)

/**
 * Performs the given [operation] on each element of
 * this [Iterator].
 */
Iterator<T>.forEach(operation: (T) -> Unit): Unit {
    for (element in this) operation(element)
}

/**
 * Iterator transforming original `iterator` into iterator of [IndexedValue],
 * counting index from zero.
 */
IndexingIterator<T>(private val iterator: Iterator<T>): Iterator<IndexedValue<T>> {
    private var index = 0
    final override fun hasNext(): Boolean = iterator.hasNext()
    final override fun next(): IndexedValue<T> =
        IndexedValue(checkIndexOverflow(index++), iterator.next())
}

/*
 * Copyright 2010-2019 JetBrains s.r.o.
 * Use of this source code is governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("ComparisonsKt")

package kotlin.comparisons

/*
 * NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
 * See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
 */

import kotlin.random.*

/**
 * Returns the greater of two values.
 * If values are equal, returns the first one.
 */
maxOf(a: T, b: T): T

/**
 * Returns the greater of two values.
 */
maxOf(a: Byte, b: Byte): Byte

/**
 * Returns the greater of two values.
 */
maxOf(a: Short, b: Short): Short

/**
 * Returns the greater of two values.
 */
maxOf(a: Int, b: Int): Int

/**
 * Returns the greater of two values.
 */
maxOf(a: Long, b: Long): Long

/**
 * Returns the greater of two values.
 */
maxOf(a: Float, b: Float): Float

/**
 * Returns the greater of three values.
 */
maxOf(a: T, b: T, c: T): T {
    return maxOf(a, maxOf(b, c, comparator), comparator)
}

/**
 * Returns the greater of two values.
 * If values are equal, returns the first one.
 */
minOf(a: T, b: T, comparator: Comparator<T>): T

/*
 * Open Source Used In DNAC Platform Antelope 8010
 */
fun minOf(a: Long, b: Long): Long

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Float, b: Float):
Float

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Double, b: Double):
Double

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Byte, b: Byte, c: Byte):
Byte

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Short, b: Short, c: Short):
Short

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Int, b: Int, c: Int):
Int

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Long, b: Long, c: Long):
Long

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Float, b: Float, c: Float):
Float

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Double, b: Double, c: Double):
Double

/**
 * Returns the smaller of three values according to the order specified by the given [comparator].
 */
@SinceKotlin("1.1")
public fun <T> minOf(a: T, b: T, c: T, comparator: Comparator<in T>): T

/**
 * Returns the smaller of two values
according to the order specified by the given 
[comparator]. If values are equal, returns the first one.
 */
@SinceKotlin("1.1")
public fun <T> minOf(a: T, b: T, comparator: Comparator<in T>): T

/**
 * Returns a [List] containing all key-value pairs.
 */
public fun <K, V> Map<out K, V>.toList(): List<Pair<K, V>> {
    if (size == 0)
        return emptyList()
    val iterator = entries.iterator()
    if (!iterator.hasNext())
        return emptyList()
    val first = iterator.next()
    if (!iterator.hasNext())
        return listOf(first.toPair())
    val result = ArrayList<Pair<K, V>>(size)
    result.add(first.toPair())
    do {
        result.add(iterator.next().toPair())
    } while (iterator.hasNext())
    return result
}

/**
 * Returns a single list of all elements yielded from results of 
[transform] function being invoked on each entry of original map.
 */
public inline fun <K, V, R> Map<out K, V>.map(transform: (Map.Entry<K, V>) -> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}

/**
 * Returns a list containing only the non-null results of applying the given 
[transform] function to each entry in the original map.
 */
public inline fun <K, V, R : Any> Map<out K, V>.mapNotNull(transform: (Map.Entry<K, V>) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(), transform)
}

/**
 * Applies the given 
[transform] function being invoked on each entry of original map, to the given [destination].
 */
public inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) -> R): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Returns a list containing the results of applying the given [transform] function to each entry in the original map.
 */
public inline fun <K, V, R><Map<out K, V>>.flatMap(transform: (Map.Entry<K, V>) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Returns a list of all elements yielded from results of 
[transform] function being invoked on each entry of original map, to the given destination.
 */
public inline fun <K, V, R> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Returns a list containing all elements yielded from results of 
[transform] function being invoked on each entry of original map.
 */
public inline fun <K, V, R> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> R): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}
R>> Map{out K, V}.mapNotNullTo(destination: C, transform: (Map.Entry<K, V>) -> R?): C {
    for (item in this)
        if (predicate(item))
            return false
    return true
}

R>> Map{out K, V}.mapTo(destination: C, transform: (Map.Entry<K, V>) -> R?): C {
    foreach (element) {
        transform(element)?.
            let (destination, add(it))
            return destination
    }
    return destination
}

R>> Map{out K, V}.count(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {
    if (isEmpty())
        return true
    for (element in this)
        if (predicate(element))
            return false
    return true
}

R>> Map{out K, V}.forEach(action: (Map.Entry<K, V>) -> Unit): Unit {
    for (element in this)
        action(element)
}

R>> Map{out K, V}.onEach(action: (Map.Entry<K, V>) -> Unit): M {
    return apply {
        for (element in this)
            action(element)
    }
}

R>> Map{out K, V}.asIterable(): Iterable<Map.Entry<K, V>> {
    return entries
}

R>> Map{out K, V}.asSequence(): Sequence<Map.Entry<K, V>> {
    return entries.asSequence()
}

R>> Map{out K, V}.mapTo(destination: C, transform: (Map.Entry<K, V>) -> R?): C {
    for (item in this)
        if (predicate(item))
            return false
    return true
}

R>> Map{out K, V}.mapTo(destination: C, transform: (Map.Entry<K, V>) -> R?): C {
    foreach (element) {
        transform(element)?.
            let (destination, add(it))
            return destination
    }
    return destination
}

R>> Map{out K, V}.count(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {
    if (isEmpty())
        return true
    for (element in this)
        if (predicate(element))
            return false
    return true
}

R>> Map{out K, V}.forEach(action: (Map.Entry<K, V>) -> Unit): Unit {
    for (element in this)
        action(element)
}

R>> Map{out K, V}.onEach(action: (Map.Entry<K, V>) -> Unit): M {
    return apply {
        for (element in this)
            action(element)
    }
}

R>> Map{out K, V}.asIterable(): Iterable<Map.Entry<K, V>> {
    return entries
}

R>> Map{out K, V}.asSequence(): Sequence<Map.Entry<K, V>> {
    return entries.asSequence()
}
```kotlin
//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib/

import kotlin.random.*

/**
 * Returns `true` if [element] is found in the sequence.
 */
public operator fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.contains(element: T): Boolean {
    return indexOf(element) != -1
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the
 * [index] is out of bounds of this sequence.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.elementAt(index: Int): T {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("Sequence doesn't contain element at index $index.") }
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the
 * [index] is out of bounds of this sequence.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.elementAtOrNull(index: Int): T? {
    if (index < 0)
        return null
    val iterator = iterator()
    var count = 0
    while (iterator.hasNext()) {
        val element = iterator.next()
        if (index == count++)
            return element
    }
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 */
@kotlin.internal.InlineOnly
public inline fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.find(predicate: (T) -> Boolean): T? {
    return firstOrNull(predicate)
}

/**
 * Returns the first index of [element], or -1 if the sequence does not contain
 * [element].
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (element == item)
            return index
        index++
    }
    return -1
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if no such element was found.
 */
public inline fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOfFirst(predicate: (T) -> Boolean): Int {
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (predicate(item))
            return index
        index++
    }
    return -1
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the sequence does not contain
 * [element].
 */
public inline fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOfLast(predicate: (T) -> Boolean): Int {
    var lastIndex = -1
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (predicate(item))
            lastIndex = index
        index++
    }
    return lastIndex
}
```

public fun <T> Sequence<T>.last(): T {
    val iterator = iterator()
    if (!iterator.hasNext())
        throw NoSuchElementException("Sequence is empty.")
    var last = iterator.next()
    while (iterator.hasNext())
        last = iterator.next()
    return last
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such 
 * element was found.
 */
public inline fun <T> Sequence<T>.lastOrNull(predicate: (T) -> Boolean): T? {
    var last: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            last = element
            found = true
        }
    }
    return if (!found) throw NoSuchElementException("Sequence contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return last as T
}

/**
 * Returns the single element, or throws an exception if the sequence is empty or has more than one element.
 */
public fun <T> Sequence<T>.single(): T {
    val iterator = iterator()
    if (!iterator.hasNext())
        throw NoSuchElementException("Sequence is empty.")
    val single = iterator.next()
    if (iterator.hasNext())
        throw IllegalArgumentException("Sequence has more than one element.")
    return single
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one 
 * matching element.
 */
public inline fun <T> Sequence<T>.singleOrNull(predicate: (T) -> Boolean): T? {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Sequence contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) return null
    @Suppress("UNCHECKED_CAST")
    return single as T
}

/**
 * Returns a sequence containing all elements except first [n] elements.
 */
public fun <T> Sequence<T>.drop(n: Int): Sequence<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return when {
        n == 0 -> this
        this is DropTakeSequence -> this.drop(n)
        else -> DropSequence(this, n)
    }
}

/**
 * Returns a sequence containing all elements except first elements that satisfy the given [predicate].
 */
public fun <T> Sequence<T>.dropWhile(predicate: (T) -> Boolean): Sequence<T> {
    return when when (predicate) {
        true -> this
        else -> DropWhileSequence(this, predicate)
    }
}

/**
 * Returns a sequence containing only elements matching the given [predicate].
 */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    return when when (predicate) {
        true -> this
        else -> FilterSequence(this, predicate)
    }
}

/**
 * Returns a sequence containing all elements except first [n] elements that satisfy the given [predicate].
 */
public fun <T> Sequence<T>.dropTake(n: Int): Sequence<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return when {
        n == 0 -> this
        this is DropTakeSequence -> this.dropTake(n)
        else -> DropTakeSequence(this, n)
    }
}

/**
 * Returns a sequence containing all elements except first elements that satisfy the given [predicate].
 */
public fun <T> Sequence<T>.dropWhileTake(n: Int): Sequence<T> {
    return when when (predicate) {
        true -> this
        else -> DropWhileTakeSequence(this, predicate, n)
    }
}

/**
 * Returns a sequence containing only elements matching the given [predicate].
 */
public fun <T> Sequence<T>.filterTake(n: Int): Sequence<T> {
    return when when (predicate) {
        true -> this
        else -> FilterTakeSequence(this, predicate, n)
    }
}
* \n * @sample samples.collections.Collections.Filtering.filter\n */
public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {
    return FilteringSequence(this, true, predicate)\n}

/**
 * Returns a sequence containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself.\n * Returns the result of predicate evaluation on the element\n * The operation is _intermediate_ and _stateless_.\n */
public fun <T> Sequence<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): Sequence<T> {
    return FilteringSequence(IndexingSequence(this), true, predicate)\n}

/**
 * Returns a sequence containing elements that are instances of specified type parameter R.\n * The operation is _intermediate_ and _stateless_.\n */
public fun <reified R> Sequence<@kotlin.internal.NoInfer R>.filterIsInstance(): Sequence<R> {
    @Suppress("UNCHECKED_CAST")
    return filter { it is R } as Sequence<R>\n}

/**
 * Appends all elements matching the given [predicate] to the given [destination].\n * The operation is _intermediate_ and _stateless_.\n * @sample samples.collections.Collections.Filtering.filter\n */
public fun <T> Sequence<T>.filterTo(destination: MutableList<T>, predicate: (T) -> Boolean): MutableList<T> {
    for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n}

/**
 * Returns a sequence containing first \[n\] elements.\n * @throws IllegalArgumentException if \[n\] is negative.\n * @sample samples.collections.Collections.Transformations.take\n */
public fun <T> Sequence<T>.take(n: Int): Sequence<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }\n    return when {
        n == 0 -> emptySequence()\n        this is DropTakeSequence -> this.take(n)\n        else -> TakeSequence(this, n)\n    }\n}

/**
 * Returns a sequence containing first elements satisfying the given [predicate].\n * @sample samples.collections.Collections.Transformations.take\n */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)\n}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].\n * The operation is _intermediate_ and _stateless_.\n * @sample samples.collections.Collections.Transformations.take\n */
public fun <T> Sequence<T>.filterNotTo(destination: MutableList<T>, predicate: (T) -> Boolean): MutableList<T> {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}

/**
 * Returns a sequence containing all elements not matching the given [predicate].\n * The operation is _intermediate_ and _stateless_.\n * @sample samples.collections.Collections.Filtering.filter\n */
public fun <T> Sequence<T>.filterNot(predicate: (T) -> Boolean): Sequence<T> {
    return FilteringSequence(this, false, predicate)\n}

/**
 * Returns a sequence containing all elements not `null`.\n * The operation is _intermediate_ and _stateless_.\n * @sample samples.collections.Collections.Filtering.filterNot\n */
public fun <T : Any> Sequence<T?>.filterNotNull(): Sequence<T> {
    @Suppress("UNCHECKED_CAST")
    return filterNot { it == null } as Sequence<T>\n}

/**
 * Appends all elements not `null` to the given [destination].\n * The operation is _intermediate_ and _stateless_.\n * @sample samples.collections.Collections.Filtering.filterNot\n */
public fun <C : MutableList<T>, T : Any> Sequence<T?>.filterNotNullTo(destination: C): MutableList<T> {
    for (element in this) if (element != null) destination.add(element)\n    return destination\n}

/**
 * Returns a sequence that yields elements of this sequence sorted according to their natural sort order.\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * The operation is _intermediate_ and _stateful_.\n */
public fun <T : Comparable<T>> Sequence<T>.sorted(): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sorted.toMutableList()\n            sortedList.sort()\n            return sortedList.iterator()\n        }\n    }\n}

Open Source Used In DNAC Platform Antelope 8015
return sortedList.iterator()\n\nSequence<T>.sortedBy(crossinline selector: (T) -> R?): Sequence<T> \n{\n    return sortedWith(compareBy(selector))\n}\n\n/**
 * Returns a sequence that yields elements of this sequence sorted
descending according to natural sort order of the value returned by specified [selector] function.\n* \n* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n* \n* The operation is _intermediate_ and _stateful_.\n* \n* The operation\n\nSequence<T>.sortedByDescending(crossinline selector: (T) -> R?): Sequence<T> \n{\n    return sortedWith(compareByDescending(selector))\n}\n\n/**
 * Returns a sequence that yields elements of this sequence sorted
descending according to the specified [comparator].\n* \n* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n* \n* The operation is _intermediate_ and _stateful_.\n* \n* The operation\n\nSequence<T>.sortedDescending(): Sequence<T> \n{\n    return sortedWith(reverseOrder())\n}\n\n/**
 * Returns a [Map] containing key-value pairs provided by [transform] function\n* applied to elements of the given sequence.\n* \n* The operation is _terminal_.\n* \n* @sample\n\nsamples.collections.Collections.Transformations.associate\n*/
public inline fun <T, K, V>\n    Sequence<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    return associateTo(LinkedHashMap<K, V>(), transform)\n}\n\n/**
 * Returns a [Map] containing the elements from the given sequence indexed by the\n* key returned from [keySelector] function applied to each element.\n* \n* The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n* \n* The operation is _intermediate_ and _stateful_.\n* \n* The operation\n\nSequence<T>.associateBy(keySelector: (T) -> K): Map<K, T> \n{\n    return associateByTo(LinkedHashMap<K, T>(), keySelector)\n}\n\n/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n* where key is provided by the [keySelector] function applied to\n* each element of the given sequence\n* and value is the element itself.\n* \n* The operation is _terminal_.\n* \n* @sample\n\nsamples.collections.Collections.Transformations.associateByTo\n*/
public inline fun <T, K, M : MutableMap<in K, in T>>\n    Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K)\n{\n    for (element in this) {\n        destination.put(keySelector(element), element)\n    }\n    return destination\n}\n\n/**
 * Populates and returns the [destination] mutable map with\n* key-value pairs,\n* where key is provided by the [keySelector] function and\n* value is provided by the [valueTransform] function applied to elements of the\n* given sequence.\n* \n* The operation is _terminal_.\n* \n* @sample\n\nsamples.collections.Collections.Transformations.associateByTo\n*/
public inline fun <T, K, M : MutableMap<in K, in T>>\n    Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K): M \n{\n    for (element in this) {\n        destination.put(keySelector(element), element)\n    }\n    return destination\n}\n\n/**
 * Populates and returns the [destination] mutable map with\n* key-value pairs,\n* where key is provided by the [keySelector] function and\n* value is provided by the [valueTransform] function applied to elements of the\n* given sequence.\n* \n* The operation is _terminal_.\n* \n* @sample\n\nsamples.collections.Collections.Transformations.associateByTo\n*/
public inline fun <T, K, M : MutableMap<in K, in T>>\n    Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> \n{\n    for (element in this) {\n        destination.put(keySelector(element), valueTransform(element))\n    }\n    return destination\n}\n
samples.collections.Collections.Transformations.associateByToWithValueTransform<
VALUE_TRANSFORM: (T) -> V>: M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
}

This method returns a [Map] where keys are elements from the given sequence and values are produced by the
[valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the
map. The operation is _terminal_.

The operation is _terminal_.

If you store elements by a key function and need to perform a lookup operation,
you can use the following method:

```kotlin
public fun <T, K, M : MutableMap<in K, in V>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}
```

This method returns a [Map] where keys are elements from the given sequence and values are produced by the
[valueSelector] function applied to each element.

If any two elements are equal, the last one gets added to the
map. The operation is _terminal_.

### List transformations

#### map

The `map` method applies a transformation function to each element of a sequence and returns a new sequence containing the transformed elements.

```kotlin
public fun <T, R> Sequence<T>.map(transform: (T) -> R): Sequence<R> {
    return FlatteningSequence(this, transform, { it.iterator() })
}
```

This method returns a new sequence where each element is the result of applying the `transform` function to the corresponding element of the original sequence.

#### flatMap

The `flatMap` method applies a transformation function to each element of a sequence and returns a new sequence containing all the elements yielded by the transformation function.

```kotlin
public fun <T, R> Sequence<T>.flatMap(transform: (T) -> Sequence<R>): Sequence<R> {
    return FlatteningSequence(this, transform, { it.iterator() })
}
```

This method returns a new sequence where each element is the result of applying the `transform` function to the corresponding element of the original sequence.

### Collection transformations

#### toCollection

The `toCollection` method returns a collection containing all elements of the sequence.

```kotlin
public fun <T, C : Collection<in T>> Sequence<T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}
```

This method returns a new collection where each element is one of the elements of the original sequence.

### Set transformations

#### toSet

The `toSet` method returns a set containing all unique elements of the sequence.

```kotlin
public fun <T> Sequence<T>.toSet(): Set<T> {
    return toCollection(LinkedHashSet<T>().also { it.addAll(this) }).also { it.optimizeReadOnlySet() }
}
```

This method returns a new set where each element is one of the unique elements of the original sequence.

### Advanced transformations

#### associatedBy

The `associatedBy` method associates elements of a sequence with a key function and returns a map where each key is associated with a list of corresponding elements.

```kotlin
public fun <T, K, V> Sequence<T>.associatedBy(keySelector: (T) -> K): Map<K, MutableList<T>> {
    val map = mutableMapOf<K, MutableList<T>>()
    for (element in this) {
        val key = keySelector(element)
        val list = map.getOrPut(key) { mutableListOf<T>() }
        list.add(element)
    }
    return map
}
```

This method returns a map where each key is associated with a list of corresponding elements. Each list contains all elements for which the key function returns the same key.

### Utility functions

#### optimizedListAndSet

The `optimizedListAndSet` functions optimize the creation of a list and a set by using an already existing list or set.

```kotlin
public fun <T> Sequence<T>.optimizedListAndSet(): MutableList<T> {
    return toMutableList().also { it.optimizeReadOnlyList() }
}
```

This method returns a new list where each element is one of the elements of the original sequence.

```kotlin
public fun <T> Sequence<T>.optimizedListAndSet(): Set<T> {
    return toSet().also { it.optimizeReadOnlySet() }
}
```

This method returns a new set where each element is one of the elements of the original sequence.

### Transformation strategies

The `associateByToWithValueTransform` and `associateByToWithValueTransform` methods apply a key function and a value transformation function to each element of a sequence and return a map where each key is associated with a list of corresponding elements.

```kotlin
public inline fun <T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateByToWithValueTransform<
    valueTransform: (T) -> V>: M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}
```

This method returns a map where each key is associated with a list of corresponding elements. Each list contains all elements for which the key function returns the same key.

---

**Open Source Used In DNAC Platform Antelope 8017**
each element of the original sequence by the key returned by the given [keySelector] function applied to the elementn * and returns a map where each group key is associated with a list of corresponding values.n *n * The returned map preserves the entry iteration order of the keys produced from the original sequence.n *n * The operation is _terminal_.n *n @sample

samples.Collections.Collections.Transformations.groupByKeysAndValues

*public fun <T, K, V> Sequence<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)n *n * Groups elements of the original sequence by the key returned by the given [keySelector] functionn * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.n *n * @return The [destination] map.n *n * The operation is _terminal_.n *n @sample

samples.Collections.Collections.Transformations.groupBy

*public fun <T, K, M : MutableMap<in K, MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this)
    {
        val key = keySelector(element)n val list = destination.getOrPut(key) { ArrayList<T>() }n
        list.add(element)n
    }
    return destination
}

*public fun <T, K, V, M : MutableMap<in K, MutableList<V>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)n
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

*public inline fun <T, K> Sequence<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {
    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}

*public fun <T, R> Sequence<T>.map(transform: (T) -> R): Sequence<R> {
    return TransformingSequence(this, transform)n
}

*public fun <T, R> Sequence<T>.mapIndexed(transform: (index: Int, T) -> R): Sequence<R> {
    return TransformingIndexedSequence(this, transform)n
}

*public fun <T, R: Any> Sequence<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): Sequence<R> {
    return TransformingIndexedSequence(this, transform).filterNotNull()n
}

*public inline fun <T, R> Sequence<T>.mapIndexedNotNullTo(destination: R): R {
    return TransformingIndexedSequence(this, transform).filterNotNull().to(destination)n
}

*public fun <T, R> Sequence<T>.mapIndexedNotContains(transform: (index: Int, T) -> R): Sequence<R> {
    return TransformingIndexedSequence(this, transform).filterNotContains(n
}

*public fun <T, R> Sequence<T>.mapIndexedNotNullTo(destination: R): Sequence<R> {
    return TransformingIndexedSequence(this, transform).filterNotNullTo(destination)
The operation is _terminal_.

```java
public inline fun <T, R, C : MutableList<R>>
Sequence<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(checkIndexOverflow(index++), item))
    return destination
}
```

The operation is _terminal_.

```java
/**
 * Returns a sequence containing only the non-null results of applying the given [transform] function to each element in the original sequence.
 * The operation is _intermediate_ and _stateless_.
 */
public fun <T, R : Any> Sequence<T>.mapNotNull(transform: (T) -> R?): Sequence<R> {
    return TransformingSequence(this, transform).filterNotNull()}
```

The operation is _terminal_.

```java
/**
 * Applies the given [transform] function to each element of the original sequence
 * and appends the results to the given [destination].
 * The operation is _terminal_.
 */
public inline fun <T, R, C : MutableList<R>> Sequence<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
    forEach { element -> transform(element)?.let { destination.add(it) } }
    return destination
}
```

The operation is _terminal_.

```java
/**
 * Applies the given [transform] function to each element in the original sequence
 * and returns the result of the transform applied to the element.
 * The operation is _terminal_.
 */
public inline fun <T, R, C : MutableList<R>>
Sequence<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(checkIndexOverflow(index++), item))
    return destination
}
```

The operation is _terminal_.

```java
/**
 * Returns a sequence that wraps each element of the original sequence into an [IndexedValue] containing the index of that element and the element itself.
 * The operation is _intermediate_ and _stateless_.
 */
public fun <T> Sequence<T>.withIndex():
Sequence<IndexedValue<T>> {
    return IndexingSequence(this)
}
```

The operation is _terminal_.

```java
/**
 * Returns a mutable set containing all distinct elements from the given sequence.
 * The returned set preserves the element iteration order of the original sequence.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.toMutableSet():
MutableSet<T> {
    val set = LinkedHashSet<T>()
    for (item in this)
        set.add(item)
    return set
}
```

The operation is _terminal_.

```java
/**
 * Returns `true` if all elements match the given [predicate].
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.all(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}
```

The operation is _terminal_.

```java
/**
 * Returns the number of elements in this sequence.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.count(): Int {
    var count = 0
    for (item in this)
        checkCountOverflow(++count)
    return count
}
```

The operation is _terminal_.

```java
/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * The operation is _terminal_.
 */
public inline fun <T, R, C : MutableList<R>>
Sequence<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
    var accumulator = initial
    for (element in this)
        accumulator = operation(accumulator, element)
    return accumulator
}
```

The operation is _terminal_.

```java
/**
 * Accumulates value starting
```
with [initial] value and applying [operation] from left to right to current accumulator value and each element with its index in the original sequence. The operation is _terminal_ in _**/public inline fun `<T, R>` Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)\n    return accumulator\n}\n\n/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * @param [operation] function that takes the index of an element and the element itself and performs the desired action on the element.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n    for (element in this) action(checkIndexOverflow(index++), element)\n}

/**
 * Returns the smallest element or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.forEach(action: (T) -> Unit): Unit {\n    for (element in this) action(element)\n}

/**
 * Performs the given [action] on each element.\n * @param [operation] function that takes the index of an element and the element itself and performs the desired action on the element.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n    for (element in this) action(checkIndexOverflow(index++), element)\n    return element\n}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.maxBy(selector: (T) -> R): T {\n    var maxElem = iterator.next()\n    var maxValue = selector(maxElem)\n    do {\n        var max = iterator.next()\n        if (max > maxValue) {\n            maxElem = max\n            maxValue = selector(max)\n        }\n    } while (iterator.hasNext())\n    return maxElem\n}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.max(): T {\n    var max = iterator.next()\n    do {\n        var max = iterator.next()\n        if (max > iterator.next()) max = iterator.next()\n    } while (iterator.hasNext())\n    return max\n}

/**
 * Returns the smallest element or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.min(): T {\n    var min = iterator.next()\n    do {\n        var min = iterator.next()\n        if (min < iterator.next()) min = iterator.next()\n    } while (iterator.hasNext())\n    return min\n}

/**
 * Returns the smallest element or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.minBy(selector: (T) -> R): T {\n    var min = iterator.next()\n    var minValue = selector(min)\n    do {\n        var min = iterator.next()\n        if (min < iterator.next()) min = iterator.next()\n    } while (iterator.hasNext())\n    return min\n}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.min(): T {\n    var min = iterator.next()\n    do {\n        var min = iterator.next()\n        if (min > iterator.next()) min = iterator.next()\n    } while (iterator.hasNext())\n    return min\n}

/**
 * Performs the given [operation] from left to right to current accumulator value and each element with its index in the original sequence.\n * The operation is _terminal_ in _*/public inline fun `<T, R>` Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)\n    return accumulator\n}\n\n/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.maxBy(selector: (T) -> R): T {\n    var maxElem = iterator.next()\n    var maxValue = selector(maxElem)\n    do {\n        var max = iterator.next()\n        if (max > maxValue) {\n            maxElem = max\n            maxValue = selector(max)\n        }\n    } while (iterator.hasNext())\n    return maxElem\n}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.max(): T {\n    var max = iterator.next()\n    do {\n        var max = iterator.next()\n        if (max > iterator.next()) max = iterator.next()\n    } while (iterator.hasNext())\n    return max\n}

/**
 * Returns the smallest element or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.minBy(selector: (T) -> R): T {\n    var min = iterator.next()\n    var minValue = selector(min)\n    do {\n        var min = iterator.next()\n        if (min < iterator.next()) min = iterator.next()\n    } while (iterator.hasNext())\n    return min\n}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * The operation is _terminal_ in _*/public inline fun `<T>` Sequence<T>.min(): T {\n    var min = iterator.next()\n    do {\n        var min = iterator.next()\n        if (min > iterator.next()) min = iterator.next()\n    } while (iterator.hasNext())\n    return min\n}

/**
 * Performs the given [operation] from left to right to current accumulator value and each element with its index in the original sequence.\n * The operation is _terminal_ in _*/public inline fun `<T, R>` Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)\n    return accumulator\n}
The operation is _terminal_.

* The operation is _terminal_.

```kotlin
public inline fun <T, R : Comparable<R>>
Sequence<T>.minBy(selector: (T) -> R): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minElem = iterator.next()
    var minValue = selector(minElem)
    do {
        val e = iterator.next()
        val v = selector(e)
        if (minValue > v) {
            if (minValue == e
                minValue = v
                } while (iterator.hasNext())
            return minElem
        } * Retruns the first element
    }
```

* The operation is _terminal_.

```kotlin
public fun <T> Sequence<T>.minWith(comparator: Comparator<in T>): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
```

* The operation is _terminal_.

```kotlin
public fun <T> Sequence<T>.none(): Boolean {
    return !iterator().hasNext()
}
```

* The operation is _terminal_.

```kotlin
public fun <T> Sequence<T>.onEach(action: (T) -> Unit): Sequence<T> {
    return map {
        action(it)
        it
    }
}
```

* The operation is _terminal_.

```kotlin
public fun <S, T : S>
Sequence<T>.reduce(operation: (acc: S, T) -> S): S {
    val iterator = this.iterator()
    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.")
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
    }
    return accumulator
}
```

* The operation is _terminal_.

```kotlin
public fun <T, R> Sequence<T>.scan(initial: R, operation:
An@SinceKotlin("1.3")\@ExperimentalStdlibApi\public fun <T, R> Sequence<T>.scan(initial: R, operation:
```
(acc: R, T) -> R): Sequence<R> {
    return sequence {
        yield(initial)
        var accumulator = initial
        for 
        element in this@scan) {
            accumulator = operation(accumulator, element)
            yield(accumulator)
    }
}

/**
 * Returns a sequence containing successive accumulation values generated by applying
 * operation from left to right to each element, its index in the original sequence and current accumulator value
 * that starts with [initial] value.
 * Note that `acc' value passed to [operation] function should not be mutated; it
 * otherwise it would affect the previous value in resulting sequence.
 * The [initial] value should also be immutable (or should not be mutated) as it may be passed to [operation] function later because of sequence's lazy nature.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * The operation is _intermediate_ and _stateless_.
 * @param [initial] initial value.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun <R, T> Sequence<T>.scanIndexed(initial: R,
    operation: (Int, acc: R, T) -> R): Sequence<R> {
    return sequence {
        yield(initial)
        var index = 0
        var accumulator = initial
        for (element in this@scanIndexed) {
            accumulator = operation(checkIndexOverflow(index++), accumulator, element)
            yield(accumulator)
        }
    }
}

/**
 * Returns a sequence containing successive accumulation values generated by applying [operation]
 * from left to right to each element and current accumulator value that starts with the first element of this sequence.
 * Note that `acc' value passed to [operation] function should not be mutated; it
 * otherwise it would affect the previous value in resulting sequence.
 * The operation is _intermediate_ and _stateless_.
 * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.
 * The operation is _intermediate_ and _stateless_.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun <S, T : S> Sequence<T>.scanReduce(operation: (acc: S, T) -> S): Sequence<S> {
    return sequence {
        val iterator = iterator()
        if (iterator.hasNext()) {
            var accumulator: S = iterator.next()
            yield(accumulator)
            while (iterator.hasNext()) {
                accumulator = operation(accumulator, iterator.next())
                yield(accumulator)
            }
        }
    }
}

/**
 * Returns a sequence containing successive accumulation values generated by applying [operation]
 * from left to right to each element, its index in the original sequence and current accumulator value that starts
 * with the first element of this sequence.
 * Note that `acc' value passed to [operation] function should not be mutated; it
 * otherwise it would affect the previous value in resulting sequence.
 * The operation is _intermediate_ and _stateless_.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * The operation is _intermediate_ and _stateless_.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun <S, T : S> Sequence<T>.scanReduceIndexed(operation: (index: Int, acc: S, T) -> S): Sequence<S> {
    return sequence {
        val iterator = iterator()
        if (iterator.hasNext()) {
            var accumulator: S = iterator.next()
            yield(accumulator)
            var index = 1
            while (iterator.hasNext()) {
                accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
                yield(accumulator)
            }
        }
    }
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the sequence.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the sequence.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Splits this sequence into a sequence of lists each not exceeding the given [size].
 * The last list in the resulting sequence may have less elements than the given [size].
 * @param size the number of elements to take in each list, must be positive
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun <T> Sequence<T>.requireNoNulls(): Sequence<T?> {
    return map { it ?:
        throw IllegalArgumentException("null element found in $this.") }
}

/**
 * Splits this sequence into a sequence of lists each not exceeding the given [size].
 * The last list in the resulting sequence may have less elements than the given [size].
 * @param size the number of elements to take in each list, must be positive
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun <T> Sequence<T>.splits(): Sequence<Sequence<T>> {
    return map { it ?:
        throw IllegalArgumentException("null element found in $this.") }
}

/**
 * Returns an original collection containing all the non-null elements, throwing an
 * IllegalArgumentException if there are any `null` elements.
 * The operation is _intermediate_ and _stateless_.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun <T : Any> Sequence<T?>.requireNoNulls(): Sequence<T> {
    return map { it ?:
        throw IllegalArgumentException("null element found in $this.") }
}
and can be greater than the number of elements in this sequence. The operation is _intermediate_ and _stateful_.

@SinceKotlin("1.2")
public fun <T> Sequence<T>.chunked(size: Int): Sequence<List<T>> {
    return windowed(size, size, partialWindows = true)
}

/**
 * Splits this sequence into several lists each not exceeding the given [size] and applies the given [transform] function to each.
 * The operation is _intermediate_ and _stateful_.
 */
@SinceKotlin("1.2")
public fun <T> Sequence<T>.chunked(size: Int, transform: (List<T>) -> R): Sequence<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}

/**
 * Returns a sequence containing all elements of the original sequence without the first occurrence of the given element.
 * The operation is _intermediate_ and _stateless_.
 */
public operator fun <T> Sequence<T>.minus(element: T): Sequence<T> {
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            var removed = false
            return this@minus.filter { if (!removed && it == element) { removed = true; false } else true }.iterator()
        }
    }
}

/**
 * Returns a sequence containing all elements of original sequence except the elements contained in the given elements array.
 * Note that the source sequence and the array being subtracted are iterated only when an `iterator` is requested from the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.
 * The elements array may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 * The operation is _intermediate_ and _stateful_.
 */
public operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty()) return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}

/**
 * Returns a sequence containing all elements of original sequence except the elements contained in the given elements collection.
 * Note that the source sequence and the collection being subtracted are iterated only when an `iterator` is requested from the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.
 * The elements collection may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.
 * The operation is _intermediate_ and _stateful_.
 */
public operator fun <T> Sequence<T>.minus(elements: Iterable<T>): Sequence<T> {
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.convertToSetForSetOperation()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}

/**
 * Returns a sequence containing all elements of original sequence except the elements contained in the given elements sequence.
 * Note that the source sequence and the sequence being subtracted are iterated only when an `iterator` is requested from the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.
 * The operation is _intermediate_ and _stateful_.
 */
public operator fun <T> Sequence<T>.minus(elements: Sequence<T>): Sequence<T> {
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}

/**
 * Returns a sequence containing all elements of original sequence without the first occurrence of the given element.
 * The operation is _intermediate_ and _stateless_.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minusElement(element: T): Sequence<T> {
    return minus(element)
}

/**
 * Splits the
original sequence into pair of lists, *where* first list contains elements for which [predicate] yielded `true`, while second list contains elements for which [predicate] yielded `false`. The operation is _terminal_.

```kotlin
public inline fun <T>
    Sequence<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {
    val first = ArrayList<T>()
    val second = ArrayList<T>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

```kotlin
/**
 * Returns a sequence containing all elements of the original sequence and then the given [element].
 */
public operator fun <T> Sequence<T>.plus(element: T): Sequence<T> {
    return sequenceOf(this, sequenceOf(element)).flatten()
}
```

```kotlin
/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] array.
 */
public operator fun <T> Sequence<T>.plus(elements: Array<out T>): Sequence<T> {
    return this.plus(elements.asList())
}
```

```kotlin
/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] collection.
 */
public operator fun <T> Sequence<T>.plus(elements: Iterable<T>): Sequence<T> {
    return sequenceOf(this, elements.asSequence()).flatten()
}
```

```kotlin
/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] sequence.
 */
public operator fun <T> Sequence<T>.plus(elements: Sequence<T>): Sequence<T> {
    return sequenceOf(this, elements).flatten()
}
```

```kotlin
/**
 * Returns a sequence containing all elements of original sequence and then the given [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.plusElement(element: T): Sequence<T> {
    return plus(element)
}
```

```kotlin
/**
 * Returns a sequence of snapshots of the window of the given [size] sliding along this sequence with the given [step], where each snapshot is a list.
 */
    return windowedSequence(size, step, partialWindows, reuseBuffer = false)
}
```

```kotlin
/**
 * Returns a sequence of results of applying the given [transform] function to an each list representing a view over the window of the given [size] sliding along this sequence with the given [step].
 */
public fun <T, R>
    Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): Sequence<R> {
    return windowedSequence(size, step, partialWindows, reuseBuffer = false).map(transform)
}
```
The `zip` function combines elements from two sequences, applying a transformation function to each pair. It ends as soon as the shortest input sequence ends. The operation is intermediate and stateless.

```kotlin
/**
 * Returns a sequence of pairs of each two adjacent elements in
 * this sequence.
 * The returned sequence is empty if this sequence contains less than two elements.
 * The operation is intermediate and stateless.
 */
@SinceKotlin("1.2")
public fun <T> Sequence<T>.zipWithNext(): Sequence<Pair<T, T>> {
    return sequence {
        var current = iterator.next()
        while (iterator.hasNext()) {
            val next = iterator.next()
            yield(current to next)
            current = next
        }
    }
}
```

The `zipWithNext` function is used when you need the previous element for each new element in the sequence.

```kotlin
/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * The operation is terminal.
 */
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}
```

The `joinToString` function is useful for joining elements of a sequence into a single string.

```kotlin
/**
 * Creates an [Iterable] instance that wraps the original sequence returning its elements when being iterated.
 */
public fun <T> Sequence<T>.asIterable(): Iterable<T> {
    return Iterable { this.iterator() }
}
```

The `asIterable` function allows you to use a sequence as an iterable.

```kotlin
/**
 * Returns this sequence as a [Sequence].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.asSequence(): Sequence<T> {
    return this
}
```

The `asSequence` function returns the sequence as a sequence itself.

```kotlin
/**
 * Returns an average value of elements in the sequence.
 * The operation is terminal.
 */
@kotlin.jvm.JvmName("averageOfByte")
public fun Sequence<Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}
```

The `average` function calculates the average of Byte elements in a sequence.
Double = 0.0
var count: Int = 0
for (element in this) {
    sum += element
}
return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("averageOfInt")
public fun Sequence<Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Sequence<Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Sequence<Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("averageOfDouble")
public fun Sequence<Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Sequence<Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Sequence<Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Sequence<Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Sequence<Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Sequence<Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal._
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Sequence<Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
 * contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SetsKt")

package kotlin.collections

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns a set containing all elements of the original
 * set except the given [element].
 * The returned set preserves the element iteration order of the original set.
 */
@kotlin.jvm.JvmName("minus")
public fun <T> Set<T>.minus(element: T): Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(size))
    var removed = false
    return this.filterTo(result) { if (!removed && it == element) { removed = true; false } else true }
}

/**
 * Returns a set containing all elements of the original
 * set except the elements contained in the given [elements] array.
 * The returned set preserves the element iteration order of the original set.
 */
@kotlin.jvm.JvmName("minus")
public fun <T> Set<T>.minus(elements: Array<out T>): Set<T> {
    val result = LinkedHashSet<T>(this)
    return result.removeAll(elements)
}

/*
 * Open Source Used In DNAC Platform Antelope 8026
 */
* Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.

* The returned set preserves the element iteration order of the original set.

* The [elements] collection may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.

```kotlin
public operator fun <T> Set<T>.minus(elements: Iterable<T>): Set<T> {
    val other = elements.convertToSetForSetOperationWith(this)
    if (other.isEmpty())
        return this.toSet()
    if (other is Set)
        return this.filterNotTo(LinkedHashSet<T>()) { it in other }
    val result = LinkedHashSet<T>(this)
    result.removeAll(other)
    return result
}

/**
 * Returns a set containing all elements of the original set except the elements contained in the given [elements] sequence.

 * The returned set preserves the element iteration order of the original set.

 * The [elements] sequence may be converted to a [HashSet] to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn't change between successive invocations.

 * @sample samples.collections.Collections.Elements.minusElements
 */
public operator fun <T> Set<T>.minus(elements: Sequence<T>): Set<T> {
    val result = LinkedHashSet<T>(this)
    result.removeAll(elements)
    return result
}

/**
 * Returns a set containing all elements of the original set except the given [element].

 * The returned set preserves the element iteration order of the original set.

 * @sample samples.collections.Collections.Elements.minusElement
 */
@kotlin.internal.InlineOnly
public inline fun <T> Set<T>.minusElement(element: T): Set<T> {
    return minus(element)
}

/**
 * Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.

 * The returned set preserves the element iteration order of the original set.

 * @sample samples.collections.Collections.Elements.plusElement
 */
@kotlin.internal.InlineOnly
public inline fun <T> Set<T>.plusElement(element: T): Set<T> {
    return plus(element)
}

/**
 * Returns a set containing all elements of the original set and the given [elements] array,

 * which aren't already in this set.

 * The returned set preserves the element iteration order of the original set.

 * @sample samples.collections.Collections.Elements.plusElements
 */
public operator fun <T> Set<T>.plus(elements: Array<out T>): Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(this.size + elements.size))
    result.addAll(this)
    result.addAll(elements)
    return result
}

/**
 * Returns a set containing all elements of the original set and the given [elements] collection,

 * which aren't already in this set.

 * The returned set preserves the element iteration order of the original set.

 * @sample samples.collections.Collections.Elements.plusElements
 */
public operator fun <T> Set<T>.plus(elements: Iterable<T>): Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(elements.collectionSizeOrNull()?.let { this.size + it } ?: this.size * 2))
    result.addAll(this)
    result.addAll(elements)
    return result
}
```

---

```
/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.
 * 
 * @file:kotlin.jvm.JvmMultifileClass
 * @file:kotlin.jvm.JvmName("StringsKt")
 * @file:package
 * @file:kotlin.text
 * @file:Note: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
 * See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
 * Import kotlin.io.*
 * Import kotlin.random.*

 * '
 */

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import java.util.*

/**
 * Returns a character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char sequence.

 * @sample sample.samples.collections.Collections.CharSequence
 */
public expect fun CharSequence.elementAt(index: Int): Char

/**
 * Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.

 * @sample sample.samples.collections.Collections.Character
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.

 * @sample sample.samples.collections.Collections.Character
 */
public operator fun <Char> CharSequence.plus(index: Int): Char {
    return get(index)
}

/**
 * Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.

 * @sample sample.samples.collections.Collections.Character
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.plus(elseValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else elseValue(index)
}
```
CharSequence.elementAtOrNull(index: Int): Char? {
    return this.getOrNull(index)
}

/**
 * Returns the first character matching the given [predicate], or `null` if no such character was found.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.elementAtOrNull(index: Int): Char? {
    return this.getOrNull(index)
}

/**
 * Returns the last character matching the given [predicate], or `null` if no such character was found.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.last(): Char {
    if (isEmpty())
        throw NoSuchElementException("Char sequence is empty.")
    return this[length - 1]
}

/**
 * Returns the last character matching the given [predicate], or `null` if no such character was found.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.lastOrNull(): Char? {
    return if (isEmpty()) null else this[length - 1]
}

/**
 * Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.
 */
public fun CharSequence.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.
 */
public fun CharSequence.getOrNull(index: Int): Char? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}

/**
 * Returns index of the first character matching the given [predicate], or -1 if the char sequence does not contain such character.
 */
public inline fun CharSequence.indexOfFirst(predicate: (Char) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

/**
 * Returns index of the last character matching the given [predicate], or -1 if the char sequence does not contain such character.
 */
public inline fun CharSequence.indexOfLast(predicate: (Char) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

/**
 * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.
 */
public fun CharSequence.getOrNull(index: Int): Char? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}

/**
 * Returns a random character from this char sequence.
 * 
 * @throws NoSuchElementException if this char sequence is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun CharSequence.random(): Char {
    return random(Random)
}

/**
 * Returns a random character from this char sequence using the specified source of randomness.
 * 
 * @throws NoSuchElementException if this char sequence is empty.
 */
@SinceKotlin("1.3")
public fun CharSequence.random(random: Random): Char {
    return random.nextInt(length)
}

/**
 * Returns a random character from this char sequence, or `null` if this char sequence is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun

CharSequence.randomOrNull(): Char? {
    return randomOrNull(Random)
}

/**
 * Returns a random character from this char sequence using the specified source of randomness, or `null` if this char sequence is empty.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun CharSequence.randomOrNull(random: Random): Char? {
    return randomOrNull(Random)
}

/**
 * Returns the single character, or throws an exception if the char sequence is empty or has more than one character.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun CharSequence.single(): Char {
    return when (length) {
        0 -> throw NoSuchElementException("Char sequence is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Char sequence has more than one character.")
    }
}

/**
 * Returns the single character matching the given [predicate], or throws exception if there is no or more than one matching character.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun CharSequence.single(predicate: (Char) -> Boolean): Char {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Char sequence contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Char sequence contains no character matching the predicate.")
    return single
}

/**
 * Returns the first [n] characters removed.
 * @throws IllegalArgumentException if [n] is negative.
 */
public fun String.drop(n: Int): String {
    require(n >= 0) { "Requested character count $n is less than zero." }
    return substring(n.coerceAtMost(length))
}

/**
 * Returns a subsequence of this char sequence containing all characters except first characters that satisfy the given [predicate].
 * @throws IllegalArgumentException if [n] is negative.
 */
public fun String.dropWhile(predicate: (Char) -> Boolean): String {
    for (index in this.indices)
        if (!predicate(this[index]))
            return substring(index)
    return ""
}

/**
 * Returns a subsequence of this char sequence containing all characters except last characters that satisfy the given [predicate].
 * @throws IllegalArgumentException if [n] is negative.
 */
public fun String.dropLastWhile(predicate: (Char) -> Boolean): String {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index]))
            return subSequence(index)
    return ""
}
match the given [predicate].

```kotlin
CharSequence.filter(predicate: (Char) -> Boolean): CharSequence {
    return filterTo(StringBuilder(), predicate)
} /* Returns a string containing only those characters from the original string that match the given [predicate].

```
```java
/**
 * Returns a subsequence of this char sequence containing the last [n] characters from this string, or the entire string if this string is shorter.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.text.Strings.take
 */
public fun CharSequence.takeLast(n: Int): CharSequence {
    require(n >= 0) { "Requested character count $n is less than zero." }
    val length = length
    return subSequence(length - n.coerceAtMost(length), length)
}

/**
 * Returns a string containing the last [n] characters from this string.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.text.Strings.take
 */
public fun String.takeLast(n: Int): String {
    require(n >= 0) { "Requested character count $n is less than zero." }
    val length = length
    return substring(length - n.coerceAtMost(length))
}

/**
 * Returns a subsequence of this char sequence containing last characters that satisfy the given [predicate].
 * @sample samples.text.Strings.take
 */
public inline fun CharSequence.takeLastWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return subSequence(index + 1, length)
        }
    }
    return subSequence(0, length)
}

/**
 * Returns a string containing last characters that satisfy the given [predicate].
 * @sample samples.text.Strings.take
 */
public inline fun String.takeLastWhile(predicate: (Char) -> Boolean): String {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return substring(index + 1)
        }
    }
    return this
}

/**
 * Returns a subsequence of this char sequence containing the first characters that satisfy the given [predicate].
 * @sample samples.text.Strings.take
 */
public inline fun CharSequence.takeWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in 0 until length) {
        if (!predicate(get(index))) {
            return subSequence(0, index)
        }
    }
    return subSequence(0, length)
}

/**
 * Returns a string containing the first characters that satisfy the given [predicate].
 * @sample samples.text.Strings.take
 */
public inline fun String.takeWhile(predicate: (Char) -> Boolean): String {
    for (index in 0 until length) {
        if (!predicate(get(index))) {
            return substring(0, index)
        }
    }
    return this
}

/**
 * Returns a char sequence with characters in reversed order.
 */
public fun CharSequence.reversed(): CharSequence {
    return StringBuilder(this).reverse()
}

/**
 * Returns a string with characters in reversed order.
 */
@kotlin.internal.InlineOnly
public inline fun String.reversed(): String {
    return (this as CharSequence).reversed().toString()
}

/**
 * Returns a Map containing key-value pairs provided by [transform] function applied to characters of the given char sequence.
 * If any of two pairs would have the same key the last one gets added to the map. The returned map preserves the entry iteration order of the original char sequence.
 * @sample samples.text.Strings.associate
 */
public inline fun <K, V> CharSequence.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(length).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

/**
 * Returns a Map containing the characters from the given char sequence indexed by the key
 * returned from [keySelector] function applied to each character.
 * @sample samples.text.Strings.associateBy
 */
public inline fun <K> CharSequence.associateBy(keySelector: (Char) -> K): Map<K, Char> {
    val capacity = mapCapacity(length).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)
}

/**
 * Returns a Map containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to characters of the given char sequence.
 * @sample samples.text.Strings.associateByWithValueTransform
 */
public inline fun <K, V> CharSequence.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {
    val capacity = mapCapacity(length).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}
```

Open Source Used In DNAC Platform Antelope 8031
* and value is the character itself.

If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

Populates and returns the [destination] mutable map with key-value pairs.

If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

Populates and returns the [destination] mutable map with key-value pairs.

If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

If any two characters would have the same key returned by [keySelector] the last one gets added to the map.

If any two characters would have the same key returned by [keySelector] the last one gets added to the map.
returned map preserves the entry iteration order of the keys produced from the original char sequence.

```kotlin
CharSequence.groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector, valueTransform)\n\n* Groups values returned by the [valueTransform] function applied to each character of the original char sequence \n* by the key returned by the given [keySelector] function applied to the character \n* and returns a map where each group key is associated with a list of corresponding values.
```

```kotlin
CharSequence.groupByTo(destination: M, keySelector: (Char) -> K): M \n\n* @return The [destination] map.
```

```kotlin
CharSequence.groupingBy(crossinline keySelector: (Char) -> K): Grouping<Char, K> \n\n* override fun sourceIterator(): Iterator<Char> = this@groupingBy.iterator()
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M \n\n* @param [transform] function that takes the index of a character and the character itself \n* and returns the result of the transform applied to the character.
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K): Grouping<Char, K> \n\n* override fun keyOf(element: Char): K = keySelector(element) \n* @return A function that returns the index of a character and the character itself
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K): Grouping<Char, K> \n\n* override fun sourceIterator(): Iterator<Char> = this@groupingBy.iterator()
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M \n\n* @param [transform] function that takes the index of a character and the character itself \n* and returns the result of the transform applied to the character.
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K): Grouping<Char, K> \n\n* override fun keyOf(element: Char): K = keySelector(element) \n* @return A function that returns the index of a character and the character itself
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K): Grouping<Char, K> \n\n* override fun sourceIterator(): Iterator<Char> = this@groupingBy.iterator()
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M \n\n* @param [transform] function that takes the index of a character and the character itself
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K): Grouping<Char, K> \n\n* override fun keyOf(element: Char): K = keySelector(element) \n* @return A function that returns the index of a character and the character itself
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M \n\n* @param [transform] function that takes the index of a character and the character itself
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K): Grouping<Char, K> \n\n* override fun keyOf(element: Char): K = keySelector(element) \n* @return A function that returns the index of a character and the character itself
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M \n\n* @param [transform] function that takes the index of a character and the character itself
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K): Grouping<Char, K> \n\n* override fun keyOf(element: Char): K = keySelector(element) \n* @return A function that returns the index of a character and the character itself
```

```kotlin
CharSequence.groupingByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M \n\n* @param [transform] function that takes the index of a character and the character itself
```
sequence
* and appends the results to the given [destination].

* @param [transform] function that takes the index of a character and the character itself
* and returns the result of the transform applied to the character.

* @public inline fun <R, C : MutableCollection<in R>> CharSequence.mapIndexedTo(destination: C, transform:
    (index: Int, Char) -> R?): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

* @public inline fun <R : Any> CharSequence.mapNotNullTo(transform: (Char) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(), transform)
}

* Applies the given [transform] function to each character in the original char sequence
* and appends only the non-null results to the given [destination].

* @public inline fun <R : Any, C :MutableCollection<in R>> CharSequence.mapIndexedTo(destination: C, transform:
    (Char) -> R?): C {
    var index = 0
    for (element in this)
        destination.add(transform(element)!!)
    return destination
}

* Returns `true` if all characters match the given [predicate].

* @sample samples.collections.Collections.Aggregates.all

* Returns `true` if char sequence has at least one character.

* @sample samples.collections.Collections.Aggregates.any

* Returns `true` if at least one character matches the given [predicate].

* @sample samples.collections.Collections.Aggregates.anyWithPredicate

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each character.

* @param [operation] function that takes the index of a character, the character itself and current accumulator value, and calculates the next accumulator value.

* @public inline fun <R> CharSequence.fold(initial: R, operation: (Char, R) -> R): R {
    var accumulator = initial
    for (element in this)
        accumulator = operation(accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from right to left to each character and current accumulator value.

* @param [operation] function that takes the index of a character, the character itself and current accumulator value, and calculates the next accumulator value.

* @public inline fun <R> CharSequence.foldRight(initial: R, operation: (R, Char) -> R): R {
    var accumulator = initial
    return accumulator
}

* Performs the given [action] on each character.
function that takes current accumulator value and a character, and calculates the next accumulator value.

```kotlin
@sample samples.collections.Collections.Aggregates.scanReduce

*\@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic inline fun CharSequence.scanReduce(operation: (acc: Char, Char) -> Char): List<Char> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Char>(length).apply { add(accumulator) }
    for (index in 1 until length) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

```kotlin
/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right:\n * to each character, its index in the original char sequence and current accumulator value that starts with the first character of this char sequence:\n * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.\n * @param [operation] function that takes the index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.scanReduce
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun CharSequence.scanReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Char>(length).apply {
        add(accumulator)
    }
    for (index in 1 until length) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

```kotlin
/**
 * Splits this char sequence into a sequence of strings each not exceeding the given [size].
 * The last string in the resulting sequence may have less characters than the given [size].\n * @param [size] the number of characters to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence.\n * @sample samples.text.Strings.chunkedSequence
 */
@SinceKotlin("1.2")
public fun CharSequence.chunkedSequence(size: Int): Sequence<String> {
    return chunkedSequence(size) { it.toString() }
}
```

```kotlin
/**
 * Returns the sum of all values produced by \[selector\] function applied to each character in the char sequence.\n */
public inline fun CharSequence.sumBy(selector: (Char) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
/**
 * Returns the sum of all values produced by \[selector\] function applied to each character in the char sequence.\n */
public inline fun CharSequence.sumByDouble(selector: (Char) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
/**
 * Splits this char sequence into several char sequences each not exceeding the given [size].
 * The last char sequence may have less characters than the given [size].\n * @param [size] the number of characters to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence.\n * @sample samples.collections.Collections.Transformations.chunked
 */
@SinceKotlin("1.2")
public fun <R> CharSequence.chunked(size: Int, transform: (CharSequence) -> R): List<R> {
    return windowed(size, size, partialWindows = true) { it.toString() }
}
```

```kotlin
/**
 * Splits this char sequence into several char sequences each not exceeding the given [size].\n * The last string in the resulting sequence may have less characters than the given [size].\n * @param [size] the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence.\n * @sample samples.collections.Collections.Transformations.chunkedToSequence
 */
@SinceKotlin("1.2")
public fun <R> CharSequence.chunkedToSequence(size: Int): Sequence<String> {
    return chunkedToSequence(size) { it.toString() }
}
```
public fun <R> CharSequence.chunkedSequence(size: Int, transform: (CharSequence) -> R): Sequence<R> {
    return windowedSequence(size, step, partialWindows = true, transform = transform) { it.toString() } as Sequence<R>
}

/**
 * Splits the original string into pair of strings,
 * where *first* string contains characters for which [predicate] yielded `true`, and *second* string contains characters for which [predicate] yielded `false`. Both pair elements have the same length.
 * 
 * Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.
 * @param size the number of elements to take in each window
 * @param step the number of elements to move the window forward by on each step, by default 1
 * @param partialWindows controls whether or not to keep partial windows in the end if any, by default `false` which means partial windows won't be preserved
 * @sample samples.collections.Sequences.Transformations.averageWindows
 */
@SinceKotlin("1.2")
public fun <R> CharSequence.partition(predicate: (Char) -> Boolean): Pair<CharSequence, CharSequence> {
    val first = StringBuilder()
    val second = StringBuilder()
    for (element in this) {
        if (predicate(element)) {
            first.append(element)
        } else {
            second.append(element)
        }
    }
    return Pair(first.toString(), second.toString())
}

/**
 * Returns a list of snapshots of the window of the given [size] sliding along this char sequence with the given [step], where each snapshot is a string.
 * Several last strings may have less characters than the given [size].
 * Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.
 * @param size the number of elements to take in each window
 * @param step the number of elements to move the window forward by on each step, by default 1
 * @param partialWindows controls whether or not to keep partial windows in the end if any, by default `false` which means partial windows won't be preserved
 * @sample samples.collections.Sequences.Transformations.takeWindows
 */
@SinceKotlin("1.2")
public fun CharSequence windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<String> {
    return windowed(size, step, partialWindows) { it.toString() }
}
sequence representing a view over the window of the given [size] sliding along this char sequence with the given [step]. Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function. You should not store it or allow it to escape in some way, unless you made a snapshot of it. Several last char sequences may have less characters than the given [size]. Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence. * @param step the number of elements to take in each window * @param transform function applied to each pair of characters.
The returned list has length of the shortest char sequence. * @param size the number of elements to take in each window * @param step the number of elements to move the window forward by on each step, by default 1
 * @param partialWindows controls whether or not to keep partial windows in the end if any, by default `false` which means partial windows won't be preserved
 * @param transform function applied to each pair of characters. The returned list has length of the shortest char sequence.

```kotlin
@SinceKotlin("1.2")
public fun <R> CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): Sequence<R> {
    checkWindowSizeStep(size, step)
    val windows = (if (partialWindows) indices else 0 until length - size + 1) step step
    return windows.asSequence().map { index ->
        val end = index + size
        val coercedEnd = if (end < 0 || end > length) length else end
        transform(subSequence(index, coercedEnd))
    }
}
```

```kotlin
/**
 * Returns a sub sequence of this char sequence having leading and trailing characters matching the [predicate] removed.
 */
public inline fun CharSequence.trim(predicate: (Char) -> Boolean): CharSequence {
    var startIndex = 0
    var endIndex = length
    var startFound = false

    while (startIndex <= endIndex) {
        val index = if (!startFound) startIndex else endIndex
        val match = predicate(this[index])

        if (!startFound) {
            if (!match)
                startFound = true
            else
                startIndex += 1
        } else {
            if (!match)
                break
            else
                endIndex -= 1
        }
    }

    return subSequence(startIndex, endIndex + 1)
}
```

```kotlin
/**
 * Returns a string having leading and trailing characters matching the [predicate] removed.
 */
public fun CharSequence.toString(): String {
    return trim { it }()
}
```
String.trim(predicate: (Char) -> Boolean): String =
    (this as CharSequence).trim(predicate).toString()

/**
 * Returns a sub sequence of this char sequence having leading characters matching the [predicate] removed.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.trimStart(predicate: (Char) -> Boolean): CharSequence {
    var index = 0
    for (i in this.indices)
        if (!predicate(this[i]))
            return subSequence(index, length)
    return subSequence(0, index)
}

/**
 * Returns a string having trailing characters matching the [predicate] removed.
 */
@kotlin.internal.InlineOnly
public inline fun String.trimEnd(predicate: (Char) -> Boolean): String =
    trimEnd(predicate).toString()

/**
 * Returns a sub sequence of this char sequence having leading and trailing characters from the [chars] array removed.
 */
@kotlin.internal.InlineOnly
public inline fun String.trim(files: CharArray): CharSequence =
    trimStart(files).trimEnd(files)

/**
 * Returns a string having trailing whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun String.trim(): String =
    trim(inline = true)

/**
 * Returns a char sequence with content of this char sequence padded at the beginning
 * to the specified [length] with the specified character or space.
 * @param length the desired string length.
 * @param padChar the character to pad string with, if it has length less than the [length] specified.
 * @return Returns a string having leading whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.padStart(length: Int, padChar: Char = ' '): CharSequence {
    val sb = StringBuilder()
    for (i in 0..(length - this.length))
        sb.append(padChar)
    sb.append(this)
    return sb
}

/**
 * Returns a string having trailing whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun String.padEnd(length: Int, padChar: Char = ' '): String =
    padEnd(length, padChar).toString()

/**
 * Returns a sub sequence of this char sequence having leading whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.padStart(): CharSequence =
    padStart(inline = true)

/**
 * Returns a string having trailing whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun String.padEnd(): String =
    padEnd(inline = true)

/**
 * Returns a char sequence with content of this char sequence padded at the end
 * to the specified [length] with the specified character or space.
 * @param length the desired string length.
 * @param padChar the character to pad string with, if it has length less than the [length] specified.
 * @return Returns a string having trailing whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.padEnd(length: Int, padChar: Char = ' '): CharSequence {
    val sb = StringBuilder()
    for (i in 0..(length - this.length))
        sb.append(padChar)
    sb.append(this)
    return sb
}

/**
 * Returns a string having trailing whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun String.padEnd(length: Int, padChar: Char = ' '): String =
    padEnd(length, padChar).toString()
length the desired string length.

* @param padChar the character to pad string with, if it has length less than the
  length specified. Space is used by default.

* @return Returns a char sequence of length at least \[length\] consisting of `this` char sequence appended with \[padChar\] as many times as are necessary to reach that length.

* @param length the desired string length.

* @param padChar the character to pad string with, if it has length less than the
  length specified. Space is used by default.

* @return Returns a string of length at least \[length\] consisting of `this` string appended with \[padChar\] as many times as are necessary to reach that length.

* @sample samples.text.Strings.padEnd

```kotlin
public fun CharSequence.padEnd(length: Int, padChar: Char = ' '): CharSequence {
    if (length < 0)
        throw IllegalArgumentException("Desired length $length is less
than zero.")
    if (length <= this.length)
        return this.subSequence(0, this.length)

    val sb = StringBuilder(length)
    sb.append(this)
    for (i in 1..(length - this.length))
        sb.append(padChar)
    return
```

```kotlin
/**
 * Pads the string to the specified \[length\] at the end with the specified character or space.
 * @param length the desired string length.
 * @param padChar the character to pad string with, if it has length less than the
 *     length specified. Space is used by default.
 * @return Returns a string of length at least \[length\] consisting of `this` string appended with \[padChar\] as many times as are necessary to reach that length.
 * @sample samples.text.Strings.padEnd
 */
public fun String.padEnd(length: Int, padChar: Char = ' '): String =
    (this as CharSequence).padEnd(length, padChar).toString()

/**
 * Returns `true` if this nullable char sequence is either `null` or empty.
 * @sample samples.text.Strings.stringIsNullOrEmpty
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence?.isNullOrEmpty(): Boolean {
    contract {
        returns(false) implies (this@isNullOrEmpty != null)
    }

    return this == null || this.length == 0
}

/**
 * Returns `true` if this char sequence is empty (contains no characters).
 * @sample samples.text.Strings.stringIsEmpty
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.isEmpty(): Boolean = length == 0

/**
 * Returns `true` if this char sequence is not empty.
 * @sample samples.text.Strings.stringIsNotEmpty
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.isNotEmpty(): Boolean = length > 0

// implemented differently in JVM and JS
public fun String.isBlank(): Boolean = length() == 0 || all { it.isWhitespace() }

/**
 * Returns `true` if this char sequence is not empty and contains some characters except of whitespace characters.
 * @sample samples.text.Strings.stringIsNotBlank
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.isNotBlank(): Boolean = !isBlank()

/**
 * Returns `true` if this nullable char sequence is either `null` or empty or consists solely of whitespace characters,\n * @sample samples.text.Strings.stringIsNullOrBlank
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence?.isNullOrBlank(): Boolean {
    contract {
        returns(false) implies (this@isNullOrBlank != null)
    }

    return this == null || this.isBlank()
}

/**
 * Iterator for characters of the given char sequence.
 */
public operator fun CharSequence.iterator(): CharIterator = object : CharIterator() {
    private var index = 0

    public override fun nextChar(): Char = get(index++)

    public override fun hasNext(): Boolean = index < length
}

/** Returns the string if it is not `null`, or the empty string otherwise.
 * @sample samples.text.Strings.stringIfEmpty
 */
@kotlin.internal.InlineOnly
public inline fun String?.orEmpty(): String = this ?: ""

/**
 * Returns this char sequence if it is not empty
 * or the result of calling \[defaultValue\] function if the char sequence is empty.
 * @sample samples.text.Strings.stringIfEmpty
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : CharSequence, C : R =
    if (isEmpty()) defaultValue() else this

/**
 * Returns this char sequence if it is not empty and doesn't consist solely of whitespace characters,
 * @sample samples.text.Strings.stringIfBlank
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <C, R> C.ifBlank(defaultValue: () -> R): R where C : CharSequence, C : R =
    if (isBlank()) defaultValue() else this

/**
 * Returns the range of valid character indices for this char sequence.
 */
public val CharSequence.indices: IntRange
    get() = 0..length - 1

/**
 * Returns the index of the last character in the char sequence or -1 if it is empty.
 */
public val CharSequence.lastIndex: Int
    get() = if (isBlank()) defaultVal() else this.length - 1

/**
 * Returns the index of the last character in the char sequence or -1 if it is empty.
 * @sample samples.text.Strings.stringIfEmpty
 */
@kotlin.internal.InlineOnly
public inline fun String?.lastIndex: Int = -1

/**
 * Returns a substring specified by the given \[range\] of indices.
 * @sample samples.text.Strings.stringIfEmpty
 */
@kotlin.internal.InlineOnly
public inline fun String.substring(range: IntRange): String = substring(range.start, range.endInclusive + 1)
subsequence of this char sequence specified by the given [range] of indices.

```kotlin
public fun CharSequence.subSequence(range: IntRange): CharSequence = subSequence(range.start, range.endInclusive + 1)

/**
 * Returns a subsequence of this char sequence.
 * @param range A range of indices specifying which portion of this sequence should be included.
 * @return A new object containing the specified range of characters.
 */
@UseExperimental(ExperimentalStdlibApi::class)
public fun String.substring(range: IntRange): String = subSequence(range.start, range.endInclusive + 1).toString()
```

```kotlin
/**
 * Returns a substring of chars from a range of this char sequence starting at the [startIndex] and ending right before the [endIndex].
 * @param startIndex the start index (inclusive).
 * @param endIndex the end index (exclusive). If not specified, the length of the char sequence is used.
 * @return a substring from [startIndex] to the end index (exclusive).
 */
public fun CharSequence.substring(startIndex: Int, endIndex: Int = length): String = subSequence(startIndex, endIndex).toString()
```

```kotlin
/**
 * Returns a substring of chars at indices from the specified [range] of this char sequence.
 * @param range A range of indices specifying which portion of this sequence should be included.
 * @return a substring from [range.start] to [range.endInclusive].
 */
public fun CharSequence.substring(range: IntRange): String = subSequence(range.start, range.endInclusive + 1).toString()
```

```kotlin
/**
 * Returns a substring before the first occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
 * @param delimiter the character to search for.
 * @param missingDelimiterValue the value to return if the delimiter is not found.
 * @return a substring from the start of the string to the index of the first occurrence of [delimiter], or [missingDelimiterValue] if not found.
 */
public fun String.substringBefore(delimiter: Char, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(0, index)
}
```

```kotlin
/**
 * Returns a substring before the last occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
 * @param delimiter the character to search for.
 * @param missingDelimiterValue the value to return if the delimiter is not found.
 * @return a substring from the start of the string to the index of the last occurrence of [delimiter], or [missingDelimiterValue] if not found.
 */
public fun String.substringBeforeLast(delimiter: Char, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(0, index)
}
```

```kotlin
/**
 * Returns a substring after the first occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
 * @param delimiter the character to search for.
 * @param missingDelimiterValue the value to return if the delimiter is not found.
 * @return a substring from the index after the last occurrence of [delimiter], or [missingDelimiterValue] if not found.
 */
public fun String.substringAfter(delimiter: Char, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(index + 1, length)
}
```

```kotlin
/**
 * Returns a substring after the last occurrence of [delimiter].
 * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.
 * @param delimiter the character to search for.
 * @param missingDelimiterValue the value to return if the delimiter is not found.
 * @return a substring from the index after the first occurrence of [delimiter], or [missingDelimiterValue] if not found.
 */
public fun String.substringAfterLast(delimiter: Char, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(index + delimiter.length, length)
}
```

```kotlin
/**
 * Returns a char sequence with content of this char sequence where its part at the given range
 * is replaced with the [replacement] char sequence.
 * @param startIndex the index of the first character to be replaced.
 * @param endIndex the index after the replacement to keep in the string.
 * @param replacement the replacement string.
 * @return a new object containing the specified range of characters.
 */
public fun CharSequence.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence) =
```
IndexOutOfBoundsException("End index ($endIndex) is less than start index ($startIndex).")
    return sb
}

/**
 * Replaces the part of the string at the given range with the [replacement] char sequence.
 * @param startIndex the index of the first character to be replaced.
 * @param endIndex the index of the first character after the replacement to keep in the string.
 */
@kotlin.internal.InlineOnly
public inline fun String.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence): String =
    (this as CharSequence).replaceRange(startIndex, endIndex, replacement).toString()

/**
 * Returns a char sequence with content of the char sequence where its part at the given [range] is replaced with the [replacement] char sequence.
 * The end index of the [range] is included in the part to be replaced.
 * @param sb
 * @param startIndex
 * @param endIndex
 */
@UseExperimental(ExperimentalStdlibApi::class)
public fun String.removeRange(startIndex: Int, endIndex: Int): String =
    (this as CharSequence).removeRange(startIndex, endIndex).toString()

/**
 * Removes the part of string at the given [range].
 * The end index of the [range] is included in the removed part.
 */
@kotlin.internal.InlineOnly
public inline fun String.removeRange(range: IntRange): String =
    (this as CharSequence).removeRange(range).toString()

/**
 * If this char sequence starts with the given [prefix] and ends with the given [suffix], returns a new char sequence with both the [prefix] and [suffix] removed.
 * Otherwise returns a new char sequence with the same characters.
 * @param prefix
 * @param suffix
 */
public fun CharSequence.removeSurrounding(prefix: CharSequence, suffix: CharSequence): CharSequence {
    if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {
        return subSequence(0, length - (prefix.length + suffix.length))
    }
    return subSequence(0, length)
}

/**
 * Opens the DNAC Platform Antelope 8043
 */
Open Source Used In DNAC Platform Antelope 8043
removes from a string both the given [prefix] and [suffix] if and only if
it starts with the [prefix] and ends with the [suffix].
Otherwise returns this string unchanged.

return subSequence(prefix.length, length - suffix.length)
return subSequence(0, length)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(prefix: CharSequence, suffix: CharSequence): String {
    return subSequence(prefix.length, length - suffix.length)
}

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: CharSequence): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: String): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: Char): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: CharSequence): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: String): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: Char): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: String): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: Char): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: String): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: Char): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: String): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: Char): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: String): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: Char): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: String): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
 */
public fun String.removeSurrounding(delimiter: Char): String = removeSurrounding(delimiter, delimiter)

/**
 * Removes from a string both the given [prefix] and [suffix] if and only if
 * it starts with the [prefix] and ends with the [suffix].
 * Otherwise returns this string unchanged.
newValue: String, ignoreCase: Boolean): String // JVM- and JS-specific

/**
 * Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the given replacement.
 * @param replacement A replacement expression that can include substitutions. See [kotlin.text.Regex.Companion.escapeReplacement] for details.
 * @return a string to be used as a replacement for that match.
 */
public inline fun CharSequence.replace(regex: Regex, replacement: String): String =
    regex.replace(this, replacement)

/**
 * Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the result of the given function [transform] that takes [MatchResult] and returns a string to be used as a replacement for that match.
 * @param transform A function that takes a match result and returns a string to be used as a replacement for that match.
 * @return a new string.
 */
public inline fun CharSequence.replace(regex: Regex, noinline transform: (MatchResult) -> String): String =
    regex.replace(this, transform)

/**
 * Replaces the first occurrence of the given regular expression [regex] in this char sequence with specified [replacement] expression.
 * @param replacement A replacement expression that can include substitutions. See [kotlin.text.Regex.Companion.escapeReplacement] for details.
 * @return a new string.
 */
public inline fun CharSequence.replaceFirst(regex: Regex, replacement: String): String =
    regex.replaceFirst(this, replacement)

/**
 * Returns `true` if this char sequence matches the given regular expression.
 * @return `true` if this char sequence matches the given regular expression.
 */
public inline infix fun CharSequence.matches(regex: Regex): Boolean = regex.matches(this)

/**
 * Implementation of [regionMatches] for CharSequences.
 * Invoked when it's already known that arguments are not Strings, so that no additional type checks are performed.
 * @param thisOffset the offset to start searching in this
 * @return regionMatches results.
 */
internal fun CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean): Boolean {
    if ((otherOffset < 0) || (thisOffset < 0) || (thisOffset > this.length - length) ||
        (otherOffset > other.length - length)) {
        return false
    }

    for (index in 0 until length) {
        if (!this[thisOffset + index].equals(other[otherOffset + index], ignoreCase))
            return false
    }
    return true
}

/**
 * Returns `true` if this char sequence starts with the specified character.
 * @return `true` if this char sequence starts with the specified character.
 */
public fun CharSequence.startsWith(char: Char, ignoreCase: Boolean = false): Boolean =
    this.length > 0 &&
    this[0].equals(char, ignoreCase)

/**
 * Returns `true` if this char sequence ends with the specified character.
 * @return `true` if this char sequence ends with the specified character.
 */
public fun CharSequence.endsWith(char: Char, ignoreCase: Boolean = false): Boolean =
    this.length > 0 &&
    this[lastIndex].equals(char, ignoreCase)

/**
 * Returns `true` if this char sequence starts with the specified prefix.
 * @return `true` if this char sequence starts with the specified prefix.
 */
public fun CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && prefix is String)
        return this.startsWith(prefix)
    else
        return regionMatchesImpl(0, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if this char sequence starts with the specified prefix starting at the specified offset.
 * @return `true` if this char sequence starts with the specified prefix.
 */
public fun CharSequence.startsWith(prefix: CharSequence, startIndex: Int, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && prefix is String)
        return this.startsWith(prefix, startIndex)
    else
        return regionMatchesImpl(startIndex, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if this char sequence ends with the specified suffix.
 * @return `true` if this char sequence ends with the specified suffix.
 */
public fun CharSequence.endsWith(suffix: CharSequence, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && suffix is String)
        return this.endsWith(suffix)
    else
        return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
}

/**
 * Returns the longest string `prefix` such that this char sequence and [other] char sequence both start with this prefix,
 * taking care not to split surrogate pairs.
 * If this and [other] have no common prefix, removes the empty string.
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return the longest string `prefix`.
 */
public fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {
    val shortestLength = minOf(this.length, other.length)
    var i = 0
    while (i < shortestLength &&
        this[i].equals(other[i], ignoreCase)) {
        i++
    }
    if (this.hasSurrogatePairAt(i - 1) ||
        other.hasSurrogatePairAt(i - 1)) {
        i--
    }
    return subSequence(0, i).toString()
}

/**
 * Returns the longest string `suffix` such that this char sequence and [other] char sequence both end with this suffix,
 * taking care not to split surrogate pairs.
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return the longest string `suffix`.
 */
public fun CharSequence.commonSuffixWith(other: CharSequence, ignoreCase: Boolean = false): String {
    val shortestLength = minOf(this.length, other.length)
    var i = 0
    while (i < shortestLength &&
        this[i].equals(other[i], ignoreCase)) {
        i++
    }
    if (this.hasSurrogatePairAt(i - 1) ||
        other.hasSurrogatePairAt(i - 1)) {
        i--
    }
    return subSequence(i, this.length).toString()
}

// common prefix and suffix

/**
 * Returns the longest string `prefix` such that this char sequence and [other] char sequence both start with this prefix,
 * taking care not to split surrogate pairs.
 * If this and [other] have no common prefix, removes the empty string.
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return the longest string `prefix`.
 */
public fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {
    val shortestLength = minOf(this.length, other.length)
    var i = 0
    while (i < shortestLength &&
        this[i].equals(other[i], ignoreCase)) {
        i++
    }
    if (this.hasSurrogatePairAt(i - 1) ||
        other.hasSurrogatePairAt(i - 1)) {
        i--
    }
    return subSequence(0, i).toString()
}

/**
 * Returns the longest string `suffix` such that this char sequence and [other] char sequence both end with this suffix,
 * taking care not to split surrogate pairs.
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return the longest string `suffix`.
 */
public fun CharSequence.commonSuffixWith(other: CharSequence, ignoreCase: Boolean = false): String {
    val shortestLength = minOf(this.length, other.length)
    var i = 0
    while (i < shortestLength &&
        this[i].equals(other[i], ignoreCase)) {
        i++
    }
    if (this.hasSurrogatePairAt(i - 1) ||
        other.hasSurrogatePairAt(i - 1)) {
        i--
    }
    return subSequence(i, this.length).toString()
}
ignoreCase: Boolean = false): String {\n    val thisLength = this.length
    val otherLength = other.length
    val shortestLength = minOf(thisLength, otherLength)
    val var i = 0\n    while (i < shortestLength && & this[i] == that[i]) {
        i++
    }
    return subSequence(thisLength - i, thisLength).toString()
}

// indexOfAny()

/**
 * Finds the index of the first occurrence of any of the specified [chars] in this char sequence, \n * starting from the specified [startIndex] and optionally ignoring the case. \n * @param ignoreCase `true` to ignore character case when matching a character. \n * @return An index of the first occurrence of matched character from [chars] \n * or -1 if none of [chars] are found. \n */
public fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int = 0, ignoreCase: Boolean = false): Int {
    if (!ignoreCase && chars.size == 1) {
        val char = chars.single()
        return charIndex // smart cast\n    }
    val thisLength = this.length
    val otherLength = other.length
    val shortestLength = minOf(thisLength, otherLength)
    @\n    val var i = 0\n    while (i < shortestLength && & this[i] == that[i]) {
        i++
    }
    if (i == shortestLength) {
        return -1
    }
    return i
}

private fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int, last: Boolean = false): Int {
    if (!ignoreCase && chars.size == 1) {
        val char = chars.single()
        return charIndex // smart cast\n    }
    val thisLength = this.length
    val otherLength = other.length
    val shortestLength = minOf(thisLength, otherLength)
    val var i = 0\n    while (i < shortestLength && & this[i] == that[i]) {
        i++
    }
    if (i == shortestLength) {
        return -1
    }
    return i
}

private fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {
    if (!ignoreCase && chars.size == 1) {
        val char = chars.single()
        return charIndex // smart cast\n    }
    val indices = if (last) {\n        Collections.emptyList()\n    } else {\n        for (index in startIndex.coerceAtMost(lastIndex)) {\n            if (other.regionMatches(0, this, index, otherLength, ignoreCase)) {\n                return index
            }
        }
    }
    if (indices.isEmpty()) {\n        return -1
    }
    return indices.get(indices.indexOfLast { index -> other.regionMatches(0, this, index, otherLength, ignoreCase) })
}

private fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int, endIndex: Int, last: Boolean = false): Int {
    if (!ignoreCase && chars.size == 1) {
        val char = chars.single()
        return charIndex // smart cast\n    }
    val indices = if (last) {\n        Collections.emptyList()\n    } else {\n        for (index in startIndex.coerceAtMost(lastIndex).downTo(endIndex.coerceAtMost(lastIndex))) {\n            if (other.regionMatches(0, this, index, otherLength, ignoreCase)) {\n                return index
            }
        }
    }
    if (indices.isEmpty()) {\n        return -1
    }
    return indices.get(indices.indexOfLast { index -> other.regionMatches(0, this, index, otherLength, ignoreCase) })
}

private fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int, endIndex: Int, last: Boolean = false): String {
    val indices = if (last) {\n        Collections.emptyList()\n    } else {\n        for (index in startIndex.coerceAtMost(lastIndex).downTo(endIndex.coerceAtMost(lastIndex))) {\n            if (other.regionMatches(0, this, index, otherLength, ignoreCase)) {\n                return index to string\n            }
        }
    }
    if (indices.isEmpty()) {\n        return -1
    }
    return indices.get(indices.indexOfLast { index -> other.regionMatches(0, this, index, otherLength, ignoreCase) }) to string
}

private fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int, endIndex: Int, last: Boolean = false): CharSequence {
    val indices = if (last) {\n        Collections.emptyList()\n    } else {\n        for (index in startIndex.coerceAtMost(lastIndex).downTo(endIndex.coerceAtMost(lastIndex))) {\n            if (other.regionMatches(0, this, index, otherLength, ignoreCase)) {\n                return index to string\n            }
        }
    }
    if (indices.isEmpty()) {\n        return -1
    }
    return indices.get(indices.indexOfLast { index -> other.regionMatches(0, this, index, otherLength, ignoreCase) }) to string
}

private fun CharSequence.indexOfAny(chars: CharArray, startIndex: Int, endIndex: Int, last: Boolean = false): CharSequence {
    val indices = if (last) {\n        Collections.emptyList()\n    } else {\n        for (index in startIndex.coerceAtMost(lastIndex).downTo(endIndex.coerceAtMost(lastIndex))) {\n            if (other.regionMatches(0, this, index, otherLength, ignoreCase)) {\n                return index to string\n            }
        }
    }
    if (indices.isEmpty()) {\n        return -1
    }
    return indices.get(indices.indexOfLast { index -> other.regionMatches(0, this, index, otherLength, ignoreCase) }) to string
}
occurrence of matched string from [strings] and the string matched or 'null' if none of [strings] are found.

To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from *n* to the end toward the beginning of this string, and finds at each position the first element in [strings] *n* that matches this string at that position.

```kotlin
public fun CharSequence.findLastAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Returns the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
public fun CharSequence.indexOfAny(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Int =
    findAnyOf(strings, startIndex, ignoreCase, last = false) *n* Returns the index of the first occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
public fun CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Returns the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
public fun CharSequence.indexOf(char: Char, startIndex: Int = 0, ignoreCase: Boolean = false): Int {
    return if (ignoreCase
        || this !is String)
            indexOfAny(charArrayOf(char), startIndex, ignoreCase)
    else
        nativeIndexOf(char, startIndex)
}
```

```kotlin
public fun CharSequence.indexOf(string: String, startIndex: Int = 0, ignoreCase: Boolean = false): Int {
    return if (ignoreCase
        || this !is String)
            indexOf(string, startIndex, length, ignoreCase)
    else
        nativeIndexOf(string, startIndex)
}
```

```kotlin
public fun CharSequence.lastIndexOf(char: Char, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {
    return if (ignoreCase
        || this !is String)
            lastIndexOfAny(charArrayOf(char), startIndex, ignoreCase)
    else
        nativeLastIndexOf(char, startIndex)
}
```

```kotlin
public fun CharSequence.lastIndexOf(string: String, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {
    return if (ignoreCase
        || this !is String)
            indexOf(string, startIndex, 0, ignoreCase, last = true)
    else
        nativeLastIndexOf(string, startIndex)
}
```

```kotlin
// indexOf
```

```kotlin
/**
 * Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].
 *
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return An index of the first occurrence of matched character from [strings] or -1 if none of [strings] are found.
 *
 * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from *n* to the end toward the beginning of this string, and finds at each position the first element in [strings] *n* that matches this string at that position.
 *n* Returns the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
 */
public fun findAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the first occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// findLastAnyOf

/**
 * Finds the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
 */
public fun findLastAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// indexOfAny

/**
 * Finds the index of the first occurrence of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally ignoring the case.
 */
public fun indexOfAny(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Int =
    findAnyOf(strings, startIndex, ignoreCase, last = false) *n* Returns the index of the first occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// lastIndexOfAny

/**
 * Finds the index of the last occurrence of the specified character, starting from the specified [startIndex] and optionally ignoring the case.
 */
public fun lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Returns the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// indexOf

/**
 * Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].
 *
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return An index of the first occurrence of matched character from [strings] or -1 if none of [strings] are found.
 *
 * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from *n* to the end toward the beginning of this string, and finds at each position the first element in [strings] *n* that matches this string at that position.
 */
public fun findAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the first occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// indexOf

/**
 * Finds the index of the last occurrence of any of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally ignoring the case.
 */
public fun lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// findAnyOf

/**
 * Finds the index of the last occurrence of the specified character, starting from the specified [startIndex] and optionally ignoring the case.
 */
public fun lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// indexOf

/**
 * Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].
 *
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return An index of the first occurrence of matched character from [strings] or -1 if none of [strings] are found.
 *
 * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from *n* to the end toward the beginning of this string, and finds at each position the first element in [strings] *n* that matches this string at that position.
 */
public fun findAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the first occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// findLastAnyOf

/**
 * Finds the index of the last occurrence of any of the specified [strings] in this char sequence, starting from the specified [startIndex] and optionally ignoring the case.
 */
public fun findLastAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// indexOf

/**
 * Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].
 *
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return An index of the first occurrence of matched character from [strings] or -1 if none of [strings] are found.
 *
 * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from *n* to the end toward the beginning of this string, and finds at each position the first element in [strings] *n* that matches this string at that position.
 */
public fun findAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the first occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// lastIndexOfAny

/**
 * Finds the index of the last occurrence of the specified character, starting from the specified [startIndex] and optionally ignoring the case.
 */
public fun lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the last occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```

```kotlin
// indexOf

/**
 * Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].
 *
 * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
 * @return An index of the first occurrence of matched character from [strings] or -1 if none of [strings] are found.
 *
 * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from *n* to the end toward the beginning of this string, and finds at each position the first element in [strings] *n* that matches this string at that position.
 */
public fun findAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =
    findAnyOf(strings, startIndex, ignoreCase, last = true) *n* Finds the index of the first occurrence of any of the specified [strings] in this char sequence, *n* starting from the specified [startIndex] and optionally ignoring the case.
```
ignoreCase = ignoreCase, last = false)?.let { it.first to it.second.length } }).unwrapOr("") * @param limit The maximum number of substrings to return. Zero by default means no limit is set.

```kotlin
rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it) }
```

Splits this char sequence to a list of strings around occurrences of the specified delimiters. The method proceeds from the beginning to the end of this string, and finds at each position the first element in [delimiters] that matches this string at that position.

```kotlin
splitToSequence(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): Sequence<String> =
```

```kotlin
rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it) }
```

Splits this char sequence to a list of strings around occurrences of the specified delimiters. The method proceeds from the beginning to the end of this string, and finds at each position the first element in [delimiters] that matches this string at that position.

```kotlin
split(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): List<String> {
    if (delimiters.size == 1) {
        val delimiter = delimiters[0]
        if (!delimiter.isEmpty()) {
            return split(delimiter, ignoreCase, limit)
        }
    }

    return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }
}
```

Splits this char sequence to a list of strings around occurrences of the specified delimiters. The method proceeds from the beginning to the end of this string, and finds at each position the first element in [delimiters] that is equal to a delimiter in this instance at that position.

```kotlin
splitToSequence(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0): Sequence<String> =
```

```kotlin
split(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0): List<String> {
    if (delimiters.size == 1) {
        return split(delimiters[0].toString(), ignoreCase, limit)
    }

    return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }
}
```

Splits this char sequence to a list of strings around occurrences of the specified delimiters. The method proceeds from the beginning to the end of this string, and finds at each position the first element in [delimiters] that is equal to a delimiter in this instance at that position.

```kotlin
split(delimiter: String, ignoreCase: Boolean, limit: Int): List<String> {
    require(limit >= 0, {
        "Limit must be non-negative, but was $limit." })

    var currentOffset = 0
    var nextIndex = indexOf(delimiter, currentOffset, ignoreCase)
    if (nextIndex == -1 || limit == 1) {
        return listOf(this.toString())
    }

    val isLimited = limit > 0
    val result = ArrayList<String>(if (isLimited) limit.coerceAtMost(10) else 10)
    do {
        result.add(substring(currentOffset, nextIndex))
        currentOffset = nextIndex + delimiter.length
        // Do not search for next occurrence if we're reaching limit
        if (isLimited && & result.size == limit - 1) break
        nextIndex = indexOf(delimiter, currentOffset, ignoreCase)
    } while (nextIndex != -1)

    result.add(substring(currentOffset, length))
    return result
}
```

Splits this char sequence to a list of strings around matches of the given regular expression. The method proceeds from the beginning to the end of this string, and finds at each position the first element in [delimiters] that matches this string at that position.

```kotlin
split(regex: Regex, limit: Int = 0): List<String> =
regex.split(this, limit).unwrapOr("") * @param limit The maximum number of substrings to return. Zero by default means no limit is set.

```
Splits this char sequence to a list of lines delimited by any of the following character sequences: CRLF, LF or CR. The lines returned do not include terminating line separators.

```kotlin
/**
 * Splits this char sequence to a list of lines delimited by any of the following character sequences:
 * CRLF, LF or CR.
 * The lines returned do not include terminating line separators.
 */
public fun CharSequence.lines(): List<String> = lineSequence().toList()
```

*Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.*

```kotlin
package kotlin.text

import kotlin.js.RegExp

/**
 * Converts the characters in the specified array to a string.
 */
public actual fun String(chars: CharArray): String {
    var result = ""
    for (char in chars) {
        result += char
    }
    return result
}

/**
 * Converts the characters from a portion of the specified array to a string.
 */
public actual fun String(chars: CharArray, offset: Int, length: Int): String {
    if (offset < 0 || length < 0 || chars.size - offset < length)
        throw IndexOutOfBoundsException("size: ${chars.size}; offset: $offset; length: $length")
    var result = ""
    for (index in offset until offset + length) {
        result += chars[index]
    }
    return result
}

/**
 * Concatenates characters in this [CharArray] into a string.
 */
public actual fun CharArray.concatToString(): String {
    var result = ""
    for (char in this) {
        result += char
    }
    return result
}

/**
 * Concatenates characters in this [CharArray] or its subrange into a string.
 *
 * @param startIndex the beginning (inclusive) of the subrange of characters, 0 by default.
 * @param endIndex the end (exclusive) of the subrange of characters, size of this array by default.
 *
 * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].
 */
public actual fun CharArray.concatToString(startIndex: Int = 0, endIndex: Int = this.size): String {
    AbstractList.checkBoundsIndexes(startIndex, endIndex, size)
    var result = ""
    for (index in startIndex until endIndex) {
        result += this[index]
    }
    return result
}

/**
 * Decodes a string from the bytes in UTF-8 encoding in this array.
 *
 * Malformed byte sequences are replaced by the replacement char `\uFFFD`.
 */
public actual fun ByteArray.decodeToString(): String {
    return decodeUtf8(this, 0, size, false)
}

/**
 * Decodes a string from the bytes in UTF-8 encoding in this array or its subrange.
 *
 * @param startIndex the beginning (inclusive) of the subrange to decode, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to decode, size of this array by default.
 * @param throwOnInvalidSequence specifies whether to throw an exception on malformed byte sequence or replace it by the replacement char `\uFFFD`.
 */
public actual fun ByteArray.decodeToString(
    startIndex: Int = 0,
    endIndex: Int = this.size,
    throwOnInvalidSequence: Boolean = false
): String {
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UByteArray
@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@PublishedApi
internal constructor(@PublishedApi internal val storage: ByteArray) : Collection<UByte>
{

    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    public constructor(size: Int) : this(ByteArray(size))

    /** Returns the array element at the given [index]. This method can be called using
     * the index operator. */
    *\n    * If the [index] is out of bounds of this array, throws an
     *\n     * [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.\n    */
    public inline fun get(index: Int): UByte = storage[index].toUByte()

    /** Sets the element at the given [index] to the given [value]. This method can be called using
     * the index operator. */
    public inline fun set(index: Int, value: UByte)
    {
        storage[index] = value.toByte()
    }

    /** Returns the number of elements in the array. */
    public override val size: Int get() = storage.size

    /** Creates an iterator over the elements of the array. */
    public override operator fun iterator(): UByteIterator = Iterator(storage)

    private class Iterator(private val array: ByteArray) : UByteIterator()
    {
        private var index = 0
        override fun hasNext() = index < array.size
        override fun nextUByte() = if (index < array.size)
        array[index++].toUByte() else throw NoSuchElementException(index.toString())
    }

    override fun contains(element: UByte): Boolean
    {
        if ((element as Any?) is UByte)
        return storage.contains(element.toByte())
    }

    override fun containsAll(elements: Collection<UByte>): Boolean
    {
        return (elements as Collection<*>).all { it is UByte &&
        storage.contains(it.toByte()) }
    }

    override fun isEmpty(): Boolean = this.storage.size == 0
}

/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray(size: Int, init: (Int) -> UByte): UByteArray
    {
    return UByteArray(ByteArray(size) { index ->
    init(index).toByte() })
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UIntArray
@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@PublishedApi
internal constructor(@PublishedApi internal val storage: IntArray) : Collection<UInt>
{

    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    public constructor(size: Int) : this(IntArray(size))

    /** Returns the array element at the given [index]. This method can be called using
     * the index operator. */
    *\n    * If the [index] is out of bounds of this array, throws an
     *\n     * [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.\n    */
    public inline fun get(index: Int): UInt = storage[index].toUInt()

    /** Sets the element at the given [index] to the given [value]. This method can be called using
     * the index operator. */
    public inline fun set(index: Int, value: UInt)
    {
        storage[index] = value.toInt()
    }

    /** Returns the number of elements in the array. */
    public override val size: Int get() = storage.size

    /** Creates an iterator over the elements of the array. */
    public override operator fun iterator(): UIntIterator = Iterator(storage)

    private class Iterator(private val array: IntArray) : UIntIterator()
    {
        private var index = 0
        override fun hasNext() = index < array.size
        override fun nextUInt() = if (index < array.size)
        array[index++].toInt() else throw NoSuchElementException(index.toString())
    }

    override fun contains(element: UInt): Boolean
    {
        // TODO: Eliminate this check after KT-30016 gets fixed.\n        // Currently JS BE does not generate special bridge method for this method.\n        if ((element as Any?) is UByte)
        return false
    }

    override fun containsAll(elements: Collection<UInt>): Boolean
    {
        return (elements as Collection<*>).all { it is UInt &&
        storage.contains(it.toUInt()) }
    }

    override fun isEmpty(): Boolean = this.storage.size == 0
}
Currently JS BE does not generate special bridge method for this method.

```kotlin
if ((element as Any?) is UInt)
    return false

return storage.contains(element.toInt())
```

```kotlin
override fun isEmpty(): Boolean = this.size == 0
```

```kotlin
/**
 * Creates a new array of the specified [size], with all elements initialized to zero.
 * @param size the size of the array
 * @return a new array of size [size]
 */
@PublishedApi
public inline fun UIntArray(size: Int): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates an array of the specified [size], where each element is calculated by calling the specified
 * [init] function. *\n * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates an array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```

```kotlin
/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}
```
constructor(@PublishedApi internal val data: Long) : Comparable<ULong> {

    companion object {
        /**
         * A constant holding the minimum value an instance of ULong can have.
         */
        public const val MIN_VALUE: ULong = ULong(0)

        /**
         * A constant holding the maximum value an instance of ULong can have.
         */
        public const val MAX_VALUE: ULong = ULong(Long.MAX_VALUE)

        /**
         * The number of bytes used to represent an instance of ULong in a binary form.
         */
        public const val SIZE_BYTES: Int = 8

        /**
         * The number of bits used to represent an instance of ULong in a binary form.
         */
        public const val SIZE_BITS: Int = 64
    }

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UByte): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UShort): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UInt): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = ulongCompare(this.data, other.data)

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UByte): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UShort): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UInt): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: ULong): ULong = ULong(this.data.plus(other.data))

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UByte): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UShort): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UInt): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: ULong): ULong = ULong(this.data.minus(other.data))

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UByte): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UShort): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UInt): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UByte): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UShort): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UInt): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: ULong): ULong = ulongDivide(this, other)

    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UByte): ULong = this.rem(other.toULong())

    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UShort): ULong = this.rem(other.toULong())

    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())

    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: ULong): ULong = this.rem(other.toULong())

    /**
     * A constant holding the minimum value an instance of ULong can have.
     */
    *
    public const val MIN_VALUE: ULong = ULong(0)

    /**
     * A constant holding the maximum value an instance of ULong can have.
     */
    *
    public const val MAX_VALUE: ULong = ULong(Long.MAX_VALUE)

    /**
     * The number of bytes used to represent an instance of ULong in a binary form.
     */
    *
    public const val SIZE_BYTES: Int = 8

    /**
     * The number of bits used to represent an instance of ULong in a binary form.
     */
    *
    public const val SIZE_BITS: Int = 64
}
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = ulongRemainder(this, other)
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun inc(): ULong = ULong(data.inc())
/** Increments this value. */

@kotlin.internal.InlineOnly
public inline operator fun dec(): ULong = ULong(data.dec())
/** Decrements this value. */

@kotlin.internal.InlineOnly
public inline operator fun rangeTo(other: ULong): ULongRange = ULongRange(this, other)
/** Ranges this value from this value to the specified [other] value. */

@kotlin.internal.InlineOnly
public inline operator fun shr(bitCount: Int): ULong = ULong(data.shr(bitCount))
/** Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros. */

@kotlin.internal.InlineOnly
public inline operator fun shl(bitCount: Int): ULong = ULong(data shl bitCount)
/** Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros. */

@kotlin.internal.InlineOnly
public inline operator fun xor(other: ULong): ULong = ULong(data xor other.data)
/** Performs a bitwise XOR operation between the two values. */

@kotlin.internal.InlineOnly
public inline operator fun or(other: ULong): ULong = ULong(data or other.data)
/** Performs a bitwise OR operation between the two values. */

@kotlin.internal.InlineOnly
public inline operator fun and(other: ULong): ULong = ULong(data and other.data)
/** Performs a bitwise AND operation between the two values. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = ulongRemainder(this, other)
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun inc(): ULong = ULong(data.inc())
/** Increments this value. */

@kotlin.internal.InlineOnly
public inline operator fun dec(): ULong = ULong(data.dec())
/** Decrements this value. */

@kotlin.internal.InlineOnly
public inline operator fun rangeTo(other: ULong): ULongRange = ULongRange(this, other)
/** Ranges this value from this value to the specified [other] value. */

@kotlin.internal.InlineOnly
public inline operator fun shr(bitCount: Int): ULong = ULong(data.shr(bitCount))
/** Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros. */

@kotlin.internal.InlineOnly
public inline operator fun shl(bitCount: Int): ULong = ULong(data shl bitCount)
/** Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros. */

@kotlin.internal.InlineOnly
public inline operator fun xor(other: ULong): ULong = ULong(data xor other.data)
/** Performs a bitwise XOR operation between the two values. */

@kotlin.internal.InlineOnly
public inline operator fun or(other: ULong): ULong = ULong(data or other.data)
/** Performs a bitwise OR operation between the two values. */

@kotlin.internal.InlineOnly
public inline operator fun and(other: ULong): ULong = ULong(data and other.data)
/** Performs a bitwise AND operation between the two values. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = ulongRemainder(this, other)
/** Calculates the remainder of dividing this value by the other value. */
'ULong' value is exactly between two 'Float's, * the one with zero at least significant bit of mantissa is selected.
 */
 @kotlin.internal.InlineOnly
 public inline fun toFloat(): Float = this.toDouble().toFloat()

/**
 * Converts this [ULong] value to [Double].
 * The resulting value is the closest 'Double' to this 'ULong' value.
 * In case when this 'ULong' value is exactly between two 'Double's, * the one with zero at least significant bit of mantissa is selected.
 */
 @kotlin.internal.InlineOnly
 public inline fun toDouble(): Double = ulongToDouble(data)

    public override fun toString(): String = ulongToString(data)

}

/**
 * Converts this [Byte] value to [ULong].
 *
 * If this value is positive, the resulting 'ULong' value represents the same numerical value as this 'Byte'. * The least significant 8 bits of the resulting 'ULong' value are the same as the bits of this 'Byte' value, whereas the most significant 56 bits are filled with the sign bit of this value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Byte.toULong(): ULong = ULong(this.toLong())

/**
 * Converts this [Short] value to [ULong].
 *
 * If this value is positive, the resulting 'ULong' value represents the same numerical value as this 'Short'. * The least significant 32 bits of the resulting 'ULong' value are the same as the bits of this 'Short' value, whereas the most significant 48 bits are filled with the sign bit of this value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Short.toULong(): ULong = ULong(this.toLong())

/**
 * Converts this [Int] value to [ULong].
 *
 * If this value is positive, the resulting 'ULong' value represents the same numerical value as this 'Int'. * The least significant 32 bits of the resulting 'ULong' value are the same as the bits of this 'Int' value, whereas the most significant 32 bits are filled with the sign bit of this value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Int.toULong(): ULong = ULong(this.toLong())

/**
 * Converts this [Long] value to [ULong].
 *
 * If this value is positive, the resulting 'ULong' value represents the same numerical value as this 'Long'. * The least significant 32 bits of the resulting 'ULong' value are the same as the bits of this 'Long' value, whereas the most significant 32 bits are filled with the sign bit of this value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Long.toULong(): ULong = ULong(this)

/**
 * Converts this [Float] value to [ULong].
 *
 * The fractional part, if any, is rounded down towards zero. * Returns zero if this 'Float' value is negative or 'NaN', [ULong.MAX_VALUE] if it's bigger than 'ULong.MAX_VALUE'.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Float.toULong(): ULong = doubleToULong(this.toDouble())

/**
 * Converts this [Double] value to [ULong].
 *
 * The fractional part, if any, is rounded down towards zero. * Returns zero if this 'Double' value is negative or 'NaN', [ULong.MAX_VALUE] if it's bigger than 'ULong.MAX_VALUE'.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Double.toULong(): ULong = doubleToULong(this)

package kotlin

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UShortArray
@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@PublishedApi
internal constructor(@PublishedApi internal val storage: ShortArray) : Collection<UShort> {

    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    public constructor(size: Int) : this(ShortArray(size))

    /**
     * Returns the array element at the given [index]. This method can be called using the index operator.
     */
    public override fun get(index: Int): UShort = storage[index].toUShort()

    /**
     * Sets the element at the given [index] to the given [value].
     */
    public override fun set(index: Int, value: UShort) {
        storage[index] = value.toShort()
    }

    /** Returns the number of elements in the array. */
    public override val size: Int get() = storage.size

    /** Creates an iterator over the elements of the array. */
    public override fun iterator(): UShortIterator =
Iterator(storage)\n
private class Iterator(private val array: ShortArray) : UShortIterator() {
    private var index = 0
    override fun hasNext() = index < array.size
    override fun nextUShort() = if (index < array.size) array[index++].toUShort() else throw NoSuchElementException()

    override fun contains(element: UShort): Boolean {
        // TODO: Eliminate this check after KT-30016 gets fixed.
        return storage.contains(element.toString())
    }

    override fun containsAll(elements: Collection<UShort>): Boolean {
        return storage.containsAll(elements)
    }

    override fun isEmpty(): Boolean = this.storage.size == 0
}

/**
 * Creates a new array of the specified size, where each element is calculated by calling the specified function
 * sequentially starting from the first one.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray(size: Int, init: (Int) -> UShort): UShortArray {
    return UShortArray(ShortArray(size) { index -> init(index).toShort() })
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ushortArrayOf(vararg elements: UShort): UShortArray = elements

package kotlin.collections

import kotlin.random.*
import kotlin.ranges.*

/**
 * Returns 1st element from the array.
 * If the size of this array is less than 1, throws an IndexOutOfBoundsException except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.component1(): UInt {
    return get(0)
}

/**
 * Returns 1st element from the array.
 * If the size of this array is less than 1, throws an IndexOutOfBoundsException except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.component1(): ULong {
    return get(0)
}

/**
 * Returns 1st element from the array.
 * If the size of this array is less than 1, throws an IndexOutOfBoundsException except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component1(): UByte {
    return get(0)
}

/**
 * Returns 2nd element from the array.
 * If the size of this array is less than 2, throws an IndexOutOfBoundsException except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.component2(): UInt {
    return get(1)
}

/**
 * Returns 2nd element from the array.
 * If the size of this array is less than 2, throws an IndexOutOfBoundsException except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.component2(): ULong {
    return get(1)
}

/**
 * Returns 2nd element from the array.
 * If the size of this array is less than 2, throws an IndexOutOfBoundsException except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component2(): UByte {
    return get(1)
}
/**
 * Returns 2nd *element* from the array.
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */

/**
 * Returns 3rd *element* from the array.
 * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */

/**
 * Returns 3rd *element* from the array.
 * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */

/**
 * Returns 4th *element* from the array.
 * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */

/**
 * Returns 5th *element* from the array.
 * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
If the size of this array is less than 5, throws an `[IndexOutOfBoundsException]` except in Kotlin/JS

```kotlin
public inline operator fun UShortArray.component5(): UShort {
    return get(4)
}
```

```kotlin
public expect fun UIntArray.elementAt(index: Int): UInt
```

```kotlin
public expect fun ULongArray.elementAt(index: Int): ULong
```

```kotlin
public expect fun UByteArray.elementAt(index: Int): UByte
```

```kotlin
public expect fun UShortArray.elementAt(index: Int): UShort
```

```kotlin
public inline fun UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
public inline fun ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
public inline fun UByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
public inline fun UShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
public inline fun UIntArray.elementAtOrNull(index: Int): UInt? {
    return this.getOrNull(index)
}
```

```kotlin
public inline fun ULongArray.elementAtOrNull(index: Int): ULong? {
    return this.getOrNull(index)
}
```

```kotlin
public inline fun UByteArray.elementAtOrNull(index: Int): UByte? {
    return this.getOrNull(index)
}
```

```kotlin
public inline fun UShortArray.elementAtOrNull(index: Int): UShort? {
    return this.getOrNull(index)
}
```
Open Source Used In DNAC Platform Antelope

```kotlin
ULongArray.first(predicate: (ULong) -> Boolean): ULong {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate], or `null` if
 * no such element is found.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun ULongArray.firstOrNull(): ULong? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element matching the given [predicate], or `null`
 * if element was not found.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun UByteArray.firstOrNull(predicate: (UByte) -> Boolean): UByte? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null`
 * if element was not found.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns an element at the given [index] or the result of calling
 * the [defaultValue] function if the [index] is out of bounds of this array.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun UIntArray.getOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

<ulineOpen Source Used In DNAC Platform Antelope 8061</uline>
calling the [defaultValue] function if the [index] is out of bounds of this array.

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UShortArray.getOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.getOrElse(index: Int): UShort? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UIntArray.indexOf(element: UInt): Int {
    return storage.indexOf(element.toInt())
}

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
ULongArray.indexOf(element: ULong): Int {
    return storage.indexOf(element.toLong())
}

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UByteArray.indexOf(element: UByte): Int {
    return storage.indexOf(element.toByte())
}

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UShortArray.indexOf(element: UShort): Int {
    return storage.indexOf(element.toShort())
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUInt()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toULong()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUByte()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUInt()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toULong()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUByte()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUShort()) }
}
**Predicate**

Returns index of the last element matching the given `predicate`. If the array does not contain such element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@inline
val @Kotlin("1.3") nullable @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfLast {
        predicate(it.toUByte())
    }
}
```

**Predicate**

Returns index of the last element matching the given `predicate`. If the array does not contain such element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@inline
val @Kotlin("1.3") nullable @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {
    return storage.indexOfLast {
        predicate(it.toUShort())
    }
}
```

**Predicate**

Returns last index of `element`, or -1 if the array does not contain such element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@inline
val @Kotlin("1.3") nullable @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.lastIndexOf(element: UInt): Int {
    return storage.lastIndexOf(element.toInt())
}
```

**Predicate**

Returns last index of `element`, or -1 if the array does not contain such element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@inline
val @Kotlin("1.3") nullable @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.lastIndexOf(element: ULong): Int {
    return storage.lastIndexOf(element.toLong())
}
```

**Predicate**

Returns last index of `element`, or -1 if the array does not contain such element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@inline
val @Kotlin("1.3") nullable @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.lastIndexOf(element: UByte): Int {
    return storage.lastIndexOf(element.toUByte())
}
```

**Predicate**

Returns last index of `element`, or -1 if the array does not contain such element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@inline
val @Kotlin("1.3") nullable @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.lastIndexOf(element: UShort): Int {
    return storage.lastIndexOf(element.toUShort())
}
```
Open Source Used In DNAC Platform Antelope 8064

`UByteArray.lastIndexOf(element: UByte): Int { return storage.lastIndexOf(element.toByte())
}

/**
 * Returns last index of [element], or -1 if the array does not contain element.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public inline fun UByteArray.lastIndexOf(element: UByte): Int { return storage.lastIndexOf(element.toByte())
}

/**
 * Returns last index of [element], or -1 if the array does not contain element.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UShortArray.lastIndexOf(element: UShort): Int { return storage.lastIndexOf(element.toShort())
}

/**
 * Returns last index of [element], or -1 if the array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UIntArray.lastIndexOf(element: UInt): Int { return if (isEmpty())
    null else this[size - 1]..isNull
}

/**
 * Returns last index of [element], or -1 if the array does not contain element.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun ULongArray.lastIndexOf(element: ULong): Int { return if (isEmpty())
    null else this[size - 1].isNull
}

/**
 * Returns last null index of [element] that is not -1 if the array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UByteArray.lastIndexOf(): UByte? { return if (isEmpty())
    null else this[size - 1].isNull
}

/**
 * Returns last null index of [element] that is not -1 if the array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UShortArray.lastIndexOf(): UShort? { return if (isEmpty())
    null else this[size - 1].isNull
}

/**
 * Returns last null index of [element] that is not -1 if the array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UIntArray.lastOrNull(): UInt? { return if (isEmpty())
    null else this[size - 1]
}

/**
 * Returns last null index of [element] that is not -1 if the array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun ULongArray.lastOrNull(): ULong? { return if (isEmpty())
    null else this[size - 1]
}

/**
 * Returns last null index of [element] that is not -1 if the array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UByteArray.lastOrNull(): UByte? { return if (isEmpty())
    null else this[size - 1]
}

/**
 * Returns last null index of [element] that is not -1 if the array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UShortArray.lastOrNull(): UShort? { return if (isEmpty())
    null else this[size - 1]
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UIntArray.random(): UInt { if (isEmpty())
    throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun ULongArray.random(random: Random): ULong { if (isEmpty())
    throw NoSuchElementException("Array is empty.")
    return get(random.nextLong(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UByteArray.random(random: Random): UByte { if (isEmpty())
    throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UShortArray.random(random: Random): UShort { if (isEmpty())
    throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
get(random.nextInt(size))\n\n* Returns a random element from this array using the specified source of randomness.\n* \n* @throws NoSuchElementException if this array is empty.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.random(random: Random): UByte {\n  \n  \n* Returns a random element from this array using the specified source of randomness.\n* \n* @throws NoSuchElementException if this array is empty.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.random(random: Random): UShort {\n  \n  \n* Returns a random element from this array, or `null` if this array is empty.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UIntArray.randomOrNull(): UInt? {\n  \n  \n* Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun ULongArray.randomOrNull(): ULong? {\n  \n  \n* Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UByteArray.randomOrNull(): UByte? {\n  \n  \n* Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UShortArray.randomOrNull(): UShort? {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UIntArray.single(): UInt {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun ULongArray.single(): ULong {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UByteArray.single(): UByte {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UShortArray.single(): UShort {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UIntArray.single(): UInt {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun ULongArray.single(): ULong {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UByteArray.single(): UByte {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*\n@ServiceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UShortArray.single(): UShort {\n  \n  \n* Returns the single element, or throws an exception if the array is empty or has more than one element.\n*
matching the given [predicate], or throws exception if there is no or more than one matching element.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun UIntArray.single(predicate: (UInt) -> Boolean): UInt
    {\n        var single: UInt? = null
        var found = false
        for (element in this) {\n            if (predicate(element)) \n            {\n                if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n                single = element\n                found = true\n            } \n            if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n        }\n        return single as UInt
    }

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun ULongArray.single(predicate: (ULong) -> Boolean): ULong
    {\n        var single: ULong? = null
        var found = false
        for (element in this) {\n            if (predicate(element)) \n            {\n                if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n                single = element\n                found = true\n            } \n            if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n        }\n        return single as ULong
    }

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun UByteArray.single(predicate: (UByte) -> Boolean): UByte
    {\n        var single: UByte? = null
        var found = false
        for (element in this) {\n            if (predicate(element)) \n            {\n                if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n                single = element\n                found = true\n            } \n            if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n        }\n        return single as UByte
    }

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun UShortArray.single(predicate: (UShort) -> Boolean): UShort
    {\n        var single: UShort? = null
        var found = false
        for (element in this) {\n            if (predicate(element)) \n            {\n                if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n                single = element\n                found = true\n            } \n            if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n        }\n        return single as UShort
    }

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UIntArray.singleOrNull(): UInt? {\n    return if (size == 1) this[0] else null
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun ULongArray.singleOrNull(): ULong? {\n    return if (size == 1) this[0] else null
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UByteArray.singleOrNull(): UByte? {\n    return if (size == 1) this[0] else null
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UShortArray.singleOrNull(): UShort? {\n    return if (size == 1) this[0] else null
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {\n    var single: UInt? = null
    var found = false
    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        } \n    }\n    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    return single as UInt
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun ULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {\n    var single: ULong? = null
    var found = false
    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        } \n    }\n    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    return single as ULong
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {\n    var single: UByte? = null
    var found = false
    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        } \n    }\n    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    return single as UByte
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
    var found = false
    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        } \n    }\n    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    return single as UShort
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {\n    var single: UInt? = null
    var found = false
    for (element in this) {\n        if (predicate(element)) \n        {\n            if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n            single = element\n            found = true\n        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    }\n    return single as UInt
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun ULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {\n    var single: ULong? = null
    var found = false
    for (element in this) {\n        if (predicate(element)) \n        {\n            if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n            single = element\n            found = true\n        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    }\n    return single as ULong
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {\n    var single: UByte? = null
    var found = false
    for (element in this) {\n        if (predicate(element)) \n        {\n            if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n            single = element\n            found = true\n        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    }\n    return single as UByte
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes
    var found = false
    for (element in this) {\n        if (predicate(element)) \n        {\n            if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n            single = element\n            found = true\n        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    }\n    return single as UShort
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UIntArray.singleOrNull(): UInt? {\n    return if (size == 1) this[0] else null
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun ULongArray.singleOrNull(): ULong? {\n    return if (size == 1) this[0] else null
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UByteArray.singleOrNull(): UByte? {\n    return if (size == 1) this[0] else null
}

/**
 * Returns single element, or `null` if the array is empty or has more than one element.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UShortArray.singleOrNull(): UShort? {\n    return if (size == 1) this[0] else null
}
single element matching the given `predicate`, or `null` if element was not found or more than one element was
found.

```kotlin
fun UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {
    var single: UByte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

fun UShortArray.singleOrNull(predicate: (UShort) -> Boolean): UShort? {
    var single: UShort? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

fun UIntArray.drop(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." } return takeLast((size - n).coerceAtLeast(0))
}

fun ULongArray.drop(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." } return takeLast((size - n).coerceAtLeast(0))
}

fun UByteArray.drop(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count $n is less than zero." } return takeLast((size - n).coerceAtLeast(0))
}

fun UShortArray.drop(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count $n is less than zero." } return takeLast((size - n).coerceAtLeast(0))
}

fun UIntArray.dropLast(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." } return take((size - n).coerceAtLeast(0))
}

fun ULongArray.dropLast(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." } return take((size - n).coerceAtLeast(0))
}

fun UByteArray.dropLast(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count $n is less than zero." } return take((size - n).coerceAtLeast(0))
}

fun UShortArray.dropLast(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count $n is less than zero." } return take((size - n).coerceAtLeast(0))
}
```

---

**Open Source Used In DNAC Platform Antelope 8067**

- Antelope
- DNAC Platform

---
List<UShort> { 
    require(n >= 0) { "$Requested element count Sn is less than zero." } 
    return take((size - n).coerceAtLeast(0))
} 

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> { 
    for (index in lastIndex downTo 0) { 
        if (!predicate(this[index])) { 
            return take(index + 1)
        }
    }
    return emptyList()
} 

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> { 
    for (index in lastIndex downTo 0) { 
        if (!predicate(this[index])) { 
            return take(index + 1)
        }
    }
    return emptyList()
} 

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> { 
    for (index in lastIndex downTo 0) { 
        if (!predicate(this[index])) { 
            return take(index + 1)
        }
    }
    return emptyList()
} 

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> { 
    for (index in lastIndex downTo 0) { 
        if (!predicate(this[index])) { 
            return take(index + 1)
        }
    }
    return emptyList()
} 

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> { 
    var yielding = false
    val list = ArrayList<UInt>()
    for (item in this) { 
        if (yielding) { 
            list.add(item)
        } else if (!predicate(item)) { 
            list.add(item)
            yielding = true
        }
    }
    return list
} 

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.dropWhile(predicate: (ULong) -> Boolean): List<ULong> { 
    var yielding = false
    val list = ArrayList<ULong>()
    for (item in this) { 
        if (yielding) { 
            list.add(item)
        } else if (!predicate(item)) { 
            list.add(item)
            yielding = true
        }
    }
    return list
} 

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> { 
    var yielding = false
    val list = ArrayList<UByte>()
    for (item in this) { 
        if (yielding) { 
            list.add(item)
        } else if (!predicate(item)) { 
            list.add(item)
            yielding = true
        }
    }
    return list
} 

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> { 
    var yielding = false
    val list = ArrayList<UShort>()
    for (item in this) { 
        if (yielding) { 
            list.add(item)
        } else if (!predicate(item)) { 
            list.add(item)
            yielding = true
        }
    }
    return list
}
.predicate)*

/**
 * Returns a list containing only elements matching the given [predicate].
 * 
 * @sample
 * samples.collections.Collections.Filtering.filter
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {
    return filterTo(ArrayList<ULong>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * 
 * @sample
 * samples.collections.Collections.Filtering.filter
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {
    return filterTo(ArrayList<UByte>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * 
 * @sample
 * samples.collections.Collections.Filtering.filter
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {
    return filterTo(ArrayList<UShort>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * @param 
 * [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.filterIndexed(predicate: (index: Int, UInt) -> Boolean): List<UInt> {
    return filterIndexedTo(ArrayList<UInt>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * @param 
 * [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.filterIndexed(predicate: (index: Int, ULong) -> Boolean): List<ULong> {
    return filterIndexedTo(ArrayList<ULong>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * @param 
 * [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.filterIndexed(predicate: (index: Int, UByte) -> Boolean): List<UByte> {
    return filterIndexedTo(ArrayList<UByte>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * @param 
 * [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.filterIndexed(predicate: (index: Int, UShort) -> Boolean): List<UShort> {
    return filterIndexedTo(ArrayList<UShort>(), predicate)
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param 
 * [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : 
MutableCollection<in UInt>> UIntArray.filterIndexedTo(destination: C, predicate: (index: Int, UInt) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param 
 * [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : 
MutableCollection<in ULong>> ULongArray.filterIndexedTo(destination: C, predicate: (index: Int, ULong) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param 
 * [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : 
MutableCollection<in UByte>> UByteArray.filterIndexedTo(destination: C, predicate: (index: Int, UByte) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param 
 * [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : 
MutableCollection<in UShort>> UShortArray.filterIndexedTo(destination: C, predicate: (index: Int, UShort) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}
MutableCollection<in UByte>> UByteArray.filterIndexedTo(destination: C, predicate: (index: Int, UByte) -> Boolean): C { forEachIndexed { index, element ->
    if (predicate(index, element))
        destination.add(element)
} return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element
 * @returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UShort>> UShortArray.filterIndexedTo(destination: C, predicate: (index: Int, UShort) -> Boolean): C { forEachIndexed { index, element ->
    if (predicate(index, element))
        destination.add(element)
} return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Filtering.filterNot
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UIntArray.filterNot(predicate: (UInt) -> Boolean): List<UInt> {
    return filterNotTo(ArrayList<UInt>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Filtering.filterNot
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun ULongArray.filterNot(predicate: (ULong) -> Boolean): List<ULong> {
    return filterNotTo(ArrayList<ULong>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Filtering.filterNot
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UByteArray.filterNot(predicate: (UByte) -> Boolean): List<UByte> {
    return filterNotTo(ArrayList<UByte>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Filtering.filterNot
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UShortArray.filterNot(predicate: (UShort) -> Boolean): List<UShort> {
    return filterNotTo(ArrayList<UShort>(), predicate)
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UInt>> UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C {
    for (element in this) if (predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in ULong>> ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C {
    for (element in this) if (predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {
    for (element in this) if (predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UShort>> UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C {
    for (element in this) if (predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UInt>> UIntArray.filterNotTo(destination: C, predicate: (UInt) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in ULong>> ULongArray.filterNotTo(destination: C, predicate: (ULong) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UByte>> UByteArray.filterNotTo(destination: C, predicate: (UByte) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UShort>> UShortArray.filterNotTo(destination: C, predicate: (UShort) -> Boolean): C {
    for (element in this) if (!predicate(element))
        destination.add(element)
    return destination
}
(element in this) if (predicate(element)) destination.add(element)
return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {
for (element in this) if (predicate(element)) destination.add(element)
return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <C : MutableCollection<in UShort>> UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C {
for (element in this) if (predicate(element)) destination.add(element)
return destination
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.slice(indices: IntRange): List<UByte> {
if (indices.isEmpty()) return listOf()
return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.slice(indices: IntRange): List<UShort> {
if (indices.isEmpty()) return listOf()
return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.slice(indices: Iterable<Int>): List<UByte> {
val size = indices.toListDefault(10)
if (size == 0) return emptyList()
val list = ArrayList<UByte>(size)
for (index in indices) {
list.add(get(index))
}
return list
}

/**
 * Returns a list containing elements at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.slice(indices: Iterable<Int>): List<UShort> {
val size = indices.toListDefault(10)
if (size == 0) return emptyList()
val list = ArrayList<UShort>(size)
for (index in indices) {
list.add(get(index))
}
return list
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: Collection<Int>): UByteArray {
return UByteArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: Collection<Int>): UShortArray {
return UShortArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: Collection<Int>): UIntArray {
return UIntArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: Collection<Int>): ULongArray {
return ULongArray(storage.sliceArray(indices))
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.slice(indices: IntRange): List<UInt> {
if (indices.isEmpty()) return listOf()
return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.slice(indices: IntRange): List<ULong> {
if (indices.isEmpty()) return listOf()
return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

/**
 * Returns a list containing elements at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.slice(indices: Iterable<Int>): List<UInt> {
val size = indices.toListDefault(10)
if (size == 0) return emptyList()
val list = ArrayList<UInt>(size)
for (index in indices) {
list.add(get(index))
}
return list
}

/**
 * Returns a list containing elements at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.slice(indices: Iterable<Int>): List<ULong> {
val size = indices.toListDefault(10)
if (size == 0) return emptyList()
val list = ArrayList<ULong>(size)
for (index in indices) {
list.add(get(index))
}
return list
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: Collection<Int>): UIntArray {
return UIntArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: Collection<Int>): ULongArray {
return ULongArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: Collection<Int>): UByteArray {
return UByteArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: Collection<Int>): UShortArray {
return UShortArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: Collection<Int>): UIntArray {
return UIntArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: Collection<Int>): ULongArray {
return ULongArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: Collection<Int>): UByteArray {
return UByteArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements of this array at specified [indices].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: Collection<Int>): UShortArray {
return UShortArray(storage.sliceArray(indices))
}
Collection<Int>): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: IntRange): UIntArray {
    return UIntArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: IntRange): ULongArray {
    return ULongArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: IntRange): UByteArray {
    return UByteArray(storage.sliceArray(indices))
}

/**
 * Returns an array containing elements at indices in the specified [indices] range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: IntRange): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}

/**
 * Returns a list containing first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.take(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UInt>(n)
    for (item in this) {
        list.add(item)
        if (++count == n)
            break
    }
    return list
}

/**
 * Returns a list containing first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.take(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<ULong>(n)
    for (item in this) {
        list.add(item)
        if (++count == n)
            break
    }
    return list
}

/**
 * Returns a list containing first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.take(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UByte>(n)
    for (item in this) {
        list.add(item)
        if (++count == n)
            break
    }
    return list
}

/**
 * Returns a list containing first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.take(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UShort>(n)
    for (item in this) {
        list.add(item)
        if (++count == n)
            break
    }
    return list
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.takeLast(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UInt>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.takeLast(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<ULong>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.takeLast(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UByte>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.takeLast(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[0])
    var count = 0
    val list = ArrayList<UShort>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
ArrayList<ULong>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.takeLast(n: Int): List<UByte>
{
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = this.size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<UByte>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.takeLast(n: Int): List<UByte>
{
    require(n >= 0) { "Requested element count $n is less than zero." }
    if (n == 0) return emptyList()
    val size = this.size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<UByte>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.takeWhile(predicate: (UInt) -> Boolean): List<UInt> {
    val list = ArrayList<UInt>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.takeWhile(predicate: (ULong) -> Boolean): List<ULong> {
    val list = ArrayList<ULong>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.takeWhile(predicate: (UByte) -> Boolean): List<UByte> {
    val list = ArrayList<UByte>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {
    val list = ArrayList<UShort>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}
Returns a list containing first elements satisfying the given [predicate].

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {
    val list = ArrayList<UShort>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}
```

Reverses elements in the array in-place.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reverse(): Unit {
    storage.reverse()
}
```

Reverses elements in the array in-place.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reverse(): Unit {
    storage.reverse()
}
```

Reverses elements in the array in-place.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reverse(): Unit {
    storage.reverse()
}
```

Reverses elements in the array in-place.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.reversed(): List<UInt> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}
```

Reverses elements in the array in-place.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.reversed(): List<ULong> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}
```

Reverses elements in the array in-place.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.reversed(): List<UByte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}
```

Reverses elements in the array in-place.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.reversed(): List<UShort> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}
```

Returns a list with elements in reversed order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reversedArray(): UIntArray {
    return UIntArray(storage.reversedArray())
}
```

Returns a list with elements in reversed order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reversedArray(): ULongArray {
    return ULongArray(storage.reversedArray())
}
```

Returns a list with elements in reversed order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reversedArray(): UByteArray {
    return UByteArray(storage.reversedArray())
}
```

Returns a list with elements in reversed order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reversedArray(): UShortArray {
    return UShortArray(storage.reversedArray())
}
```

Returns a list with elements in reversed order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sorted(): List<UInt> {
    return copyOf().apply { sort() }.asList()
}
```

Returns a list with elements in reversed order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sorted(): List<ULong> {
    return copyOf().apply { sort() }.asList()
}
```

Returns a list with elements in reversed order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sorted(): List<UByte> {
    return copyOf().apply { sort() }.asList()
}
```

Returns a list with elements in reversed order.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sorted(): List<UShort> {
    return copyOf().apply { sort() }.asList()
}
```
order.

@ExperimentalUnsignedTypes
public fun ULongArray.sorted(): List<ULong> {
    return copyOf().apply { sort() }.asList()
}

@ExperimentalUnsignedTypes
public fun UByteArray.sorted(): List<UByte> {
    return copyOf().apply { sort() }.asList()
}

@ExperimentalUnsignedTypes
public fun UShortArray.sorted(): List<UShort> {
    return copyOf().apply { sort() }.asList()
}

@ExperimentalUnsignedTypes
public fun UIntArray.sortedArray(): UIntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

@ExperimentalUnsignedTypes
public fun ULongArray.sortedArray(): ULongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

@ExperimentalUnsignedTypes
public fun UByteArray.sortedArray(): UByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

@ExperimentalUnsignedTypes
public fun UShortArray.sortedArray(): UShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

@ExperimentalUnsignedTypes
public fun UIntArray.sortedArrayDescending(): UIntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

@ExperimentalUnsignedTypes
public fun ULongArray.sortedArrayDescending(): ULongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

@ExperimentalUnsignedTypes
public fun UByteArray.sortedArrayDescending(): UByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

@ExperimentalUnsignedTypes
public fun UShortArray.sortedArrayDescending(): UShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

@ExperimentalUnsignedTypes
public fun UIntArray.sortedDescending(): List<UInt> {
    return copyOf().apply { sort() }.reversed()
}

@ExperimentalUnsignedTypes
public fun ULongArray.sortedDescending(): List<ULong> {
    return copyOf().apply { sort() }.reversed()
}

@ExperimentalUnsignedTypes
public fun UByteArray.sortedDescending(): List<UByte> {
    return copyOf().apply { sort() }.reversed()
}

@ExperimentalUnsignedTypes
public fun UShortArray.sortedDescending(): List<UShort> {
    return copyOf().apply { sort() }.reversed()
}

@kotlin.internal.InlineOnly
public inline fun UByteArray.asByteArray(): ByteArray {
    return storage
}

@FunctionalInterface
@kotlin.internal.InlineOnly
is a view of this array where each element is a signed reinterpretation\n* of the corresponding element of this
array.\n*/

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.asIntArray(): IntArray {\n    return storage\n}

/**
 * Returns a [List] that wraps the original array.\n */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.asList(): List<ULong> {\n    return storage\n}

/**
 * Returns a [List] that wraps the original array.\n */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.asList(): List<UByte> {\n    return storage\n}

/**
 * Returns a [List] that wraps the original array.\n */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.asList(): List<UShort> {\n    return storage\n}

/**
 * Returns an array of type [LongArray], which is a view of this array where each element is
a signed reinterpretation\n* of the corresponding element of this array.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.asLongArray(): LongArray {\n    return storage\n}

/**
 * Returns an array of type [ShortArray], which is a view of this array where each element is a signed
reinterpretation\n* of the corresponding element of this array.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.asShortArray(): ShortArray {\n    return storage\n}

/**
 * Returns a [List] that wraps the original array.\n */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.asUByteArray(): UByteArray {\n    return UByteArray(this)\n}

/**
 * Returns an array of type [UIntArray], which is a view of this array where each element is an unsigned
reinterpretation\n* of the corresponding element of this array.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.asUIntArray(): UIntArray {\n    return UIntArray(this)\n}

/**
 * Returns an array of type [ULongArray], which is a view of this array where each element is an unsigned
reinterpretation\n* of the corresponding element of this array.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.asULongArray(): ULongArray {\n    return LongArray(this)\n}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,\n* i.e. contain the same number of the same elements in the
same order.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic infix fun UIntArray.contentEquals(other: UIntArray): Boolean {\n    return storage.contentEquals(other.storage)\n}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,\n* i.e. contain the same number of the same elements in the
same order.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic infix fun ULongArray.contentEquals(other: ULongArray): Boolean {\n    return storage.contentEquals(other.storage)\n}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,\n* i.e. contain the same number of the same elements in the
same order.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic infix fun UByteArray.contentEquals(other: UByteArray): Boolean {\n    return storage.contentEquals(other.storage)\n}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,\n* i.e. contain the same number of the same elements in the
same order.\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic infix fun UShortArray.contentEquals(other: UShortArray): Boolean {\n    return storage.contentEquals(other.storage)\n}

/**
 * Returns a hash code based on the contents of this array as if it is
[List].\n*/
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic fun UIntArray.contentHashCode(): Int {\n    return storage.contentHashCode()\n}
is [List].

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.contentHashCode(): Int { return storage.contentHashCode() }

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.contentHashCode(): Int {
    return storage.contentHashCode()
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.contentHashCode(): Int {
    return storage.contentHashCode()
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.contentToString(): String {
    return joinToString(", ", "[", "]")
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.contentToString(): String {
    return joinToString(", ", "[", "]")
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UIntArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
}
```

`Open Source Used In DNAC Platform Antelope 8077`
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`. 

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset]. 

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range. 

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`. 

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline inline fun
UByteArray.copyOf(newSize: Int): UByteArray { return UByteArray(storage.copyOf(newSize))}
* Returns new array which is a copy of the original array, resized to the given [newSize].
* The copy is either truncated or padded at the end with zero values if necessary.
* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline inline fun
UShortArray.copyOf(newSize: Int): UShortArray { return UShortArray(storage.copyOf(newSize))}
* Returns a new array which is a copy of the specified range of the original array.
* - * param fromIndex the start of the range (inclusive), must be in \`0..array.size\`.
* - * @param toIndex the end of the range (exclusive), must be in \`fromIndex..array.size\`.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline inline fun
UIntArray.copyOfRange(fromIndex: Int, toIndex: Int): UIntArray {
    return UIntArray(storage.copyOfRange(fromIndex, toIndex))
* Returns a new array which is a copy of the specified range of the original array.
* - * param fromIndex the start of the range (inclusive), must be in \`0..array.size\`.
* - * @param toIndex the end of the range (exclusive), must be in \`fromIndex..array.size\`.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline inline fun
ULongArray.copyOfRange(fromIndex: Int, toIndex: Int): ULongArray {
    return ULongArray(storage.copyOfRange(fromIndex, toIndex))
* Returns a new array which is a copy of the specified range of the original array.
* - * param fromIndex the start of the range (inclusive), must be in \`0..array.size\`.
* - * @param toIndex the end of the range (exclusive), must be in \`fromIndex..array.size\`.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline inline fun
UByteArray.copyOfRange(fromIndex: Int, toIndex: Int): UByteArray {
    return UByteArray(storage.copyOfRange(fromIndex, toIndex))
* Returns a new array which is a copy of the specified range of the original array.
* - * param fromIndex the start of the range (inclusive), must be in \`0..array.size\`.
* - * @param toIndex the end of the range (exclusive), must be in \`fromIndex..array.size\`.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline inline fun
UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {
    return UShortArray(storage.copyOfRange(fromIndex, toIndex))
* Fills this array or its subrange with the specified [element] value.
* - * param fromIndex the start of the range (inclusive), 0 by default.
* - * @param toIndex the end of the range (exclusive), size of this array by default.
* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@public fun UIntArray.fill(element: Uint, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toInt(), fromIndex, toIndex)
* Fills this array or its subrange with the specified [element] value.
* - * param fromIndex the start of the range (inclusive), 0 by default.
* - * @param toIndex the end of the range (exclusive), size of this array by default.
* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@public fun ULongArray.fill(element: ULong, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toLong(), fromIndex, toIndex)
* Fills this array or its subrange with the specified [element] value.
* - * param fromIndex the start of the range (inclusive), 0 by default.
* - * @param toIndex the end of the range (exclusive), size of this array by default.
* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@public fun UByteArray.fill(element: UByte, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toByte(), fromIndex, toIndex)
* Fills this array or its subrange with the specified [element] value.
(inclusive), 0 by default. @param toIndex the end of the range (exclusive), size of this array by default.
* @param toIndex the end of the range (exclusive), size of this array by default. @param toIndex is less than zero or toIndex is greater than the size of this array. @param toIndex is less than zero or toIndex is greater than the size of this array. @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array. @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array. @throws IllegalArgumentException if [fromIndex] is greater than [toIndex]. @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.fill(element: UShort, fromIndex: Int = 0, toIndex: Int = size): Unit { storage.fill(element.toShort(), fromIndex, toIndex) }

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UShortArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UIntArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val ULongArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UByteArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the last valid index for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UIntArray.lastIndex: Int
    get() = storage.lastIndex

/**
 * Returns the last valid index for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val ULongArray.lastIndex: Int
    get() = storage.lastIndex

/**
 * Returns the last valid index for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UByteArray.lastIndex: Int
    get() = storage.lastIndex

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.plus(element: UInt): UIntArray {
    return UIntArray(storage + element.toInt())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.plus(element: ULong): ULongArray {
    return ULongArray(storage + element.toLong())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.plus(element: UByte): UByteArray {
    return UByteArray(storage + element.toByte())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return UShortArray(storage + element.toShort())
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntArray.plus(elements: Collection<UInt>): UIntArray {
    var index = size
    val result = storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toInt()
    return UIntArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongArray.plus(elements: Collection<ULong>): ULongArray {
    var index = size
    val result = storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toLong()
    return ULongArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UByteArray.plus(elements: Collection<UByte>): UByteArray {
    var index = size
    val result = storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toByte()
    return UByteArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UShortArray.plus(elements: Collection<UShort>): UShortArray {
    var index = size
    val result =
storage.copyOf(size + elements.size)

for (element in elements) result[index++] = element.toShort()
return UShortArray(result)

*\n
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.plus(elements: UIntArray): UIntArray {
    return UIntArray(storage + elements.storage)
}

/**
 * Returns an array of type [ByteArray], which is a copy of this array where each element is a signed reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.toByteArray(): ByteArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [IntArray], which is a copy of this array where each element is a signed reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.toIntArray(): IntArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [LongArray], which is a copy of this array where each element is a signed reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.toLongArray(): LongArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [ShortArray], which is a copy of this array where each element is a signed reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.toShortArray(): ShortArray {
    return storage.copyOf()
}

/**
 * Sorts the array in-place.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sort(): Unit {
    if (size > 1) sortArray(this)
}

/**
 * Sorts the array in-place.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sort(): Unit {
    if (size > 1) sortArray(this)
}

/**
 * Sorts the array in-place.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sort(): Unit {
    if (size > 1) sortArray(this)
}

/**
 * Returns a typed object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.toTypedArray(): ByteArray {
    return Array(size) { index -> this[index] }
}

/**
 * Returns a typed object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.toTypedArray(): Array<UInt> {
    return Array(size) { index -> this[index] }
}

/**
 * Returns a typed object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.toTypedArray(): Array<ULong> {
    return Array(size) { index -> this[index] }
}

/**
 * Returns a typed object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.toTypedArray(): Array<UByte> {
    return Array(size) { index -> this[index] }
}

Open Source Used In DNAC Platform Antelope 8081
containing all of the elements of this primitive array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.toTypedArray():
    Array<in UShort> { index -> this[index] }

// The return type of this function is UShortArray.[/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.toUByteArray(): UByteArray {
    return UByteArray(this.copyOf())
}

/**
 * Returns an array of UByte containing
 * all of the elements of this generic array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.toUByteArray(): UByteArray {
    return UByteArray(this.copyOf())
}

/**
 * Returns an array of type [UByteArray], which is a copy of this array where each element is an unsigned
 * reinterpretation[*] of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ByteArray.toUByteArray(): UByteArray {
    return UByteArray(this.copyOf())
}

/**
 * Returns an array of UInt containing all of the elements of this generic array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.toUIntArray(): UIntArray {
    return UIntArray(this.copyOf())
}

/**
 * Returns an array of type [UIntArray], which is a copy of this array where each element is an unsigned reinterpretation[*] of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.toUIntArray(): UIntArray {
    return UIntArray(this.copyOf())
}

/**
 * Returns an array of ULong containing all of the elements of this generic array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.toULongArray(): ULongArray {
    return ULongArray(this.copyOf())
}

/**
 * Returns an array of type [ULongArray], which is a copy of this array where each element is an unsigned reinterpretation[*] of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun LongArray.toULongArray(): ULongArray {
    return ULongArray(this.copyOf())
}

/**
 * Returns an array of UShort containing all of the elements of this generic array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.toUShortArray(): UShortArray {
    return UShortArray(this.copyOf())
}

/**
 * Returns an array of type [UShortArray], which is a copy of this array where each element is an unsigned reinterpretation[*] of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ShortArray.toUShortArray(): UShortArray {
    return UShortArray(this.copyOf())
}
```

Open Source Used In DNAC Platform Antelope 8082
/**
 * Appends all elements yielded from results of [transform] function being invoked on each
 * element of original array, to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UByteArray.flatMapTo(destination: C, transform: (UByte) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each
 * element of original array, to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UShortArray.flatMapTo(destination: C, transform: (UShort) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector]
 * function
 * * applied to each element and returns a map where each group key is associated with a list
 * of corresponding elements.
 * * The returned map preserves the entry iteration order of the keys produced from the
 * original array.
 * * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K> UByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {
    return groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector]
 * function
 * * applied to each element and returns a map where each group key is associated with a list
 * of corresponding elements.
 * * The returned map preserves the entry iteration order of the keys produced from the
 * original array.
 * * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K> UShortArray.groupBy(keySelector: (UShort) -> K): Map<K, List<UShort>> {
    return groupByTo(LinkedHashMap<K, MutableList<UShort>>(), keySelector)
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given
 */

Open Source Used In DNAC Platform Antelope 8083
[keySelector] function applied to the element and returns a map where each group key is associated with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array. The returned map preserves the entry iteration order of the keys produced from the original array.

```kotlin
public inline fun <K, V> UIntArray.groupBy(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array.
 * 
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, V> ULongArray.groupBy(keySelector: (ULong) -> K, valueTransform: (ULong) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the original array.
 * 
 * @return The [destination] map.
 * 
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, M : MutableMap<K, List<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<UInt>()
        }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the original array.
 * 
 * @return The [destination] map.
 * 
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, M : MutableMap<K, List<ULong>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<ULong>()
        }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the original array.
 * 
 * @return The [destination] map.
 * 
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, M : MutableMap<K, List<UByte>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<UByte>()
        }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the original array.
 * 
 * @return The [destination] map.
 * 
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, M : MutableMap<K, List<UShort>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<UShort>()
        }
        list.add(element)
    }
    return destination
}
```
MutableMap<in K, MutableList<UByte>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K):
M { for (element in this) { val key = keySelector(element) val list = destination.getOrPut(key) { ArrayList<UByte>() } list.add(element) } return destination }\n\n/**
 * Groups elements of the original array by the key returned by the given [keySelector] function * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.
 * @return The [destination] map *
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <K, M : MutableMap<in K, MutableList<V>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K):
M { for (element in this) { val key = keySelector(element) val list = destination.getOrPut(key) { ArrayList<UShort>() } list.add(element) } return destination }\n\n/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map *
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K, valueTransform: (UByte) -> V):
M { for (element in this) { val key = keySelector(element) val list = destination.getOrPut(key) { ArrayList<UByte>() } list.add(valueTransform(element)) } return destination }\n\n/**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @return The [result] list *
 * @sample samples.collections.Collections.Transformations.map
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.map(transform: (UInt) -> R): List<R> { return mapTo(ArrayList<R>(size), transform) }\n\n/**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @return The [result] list *
 * @sample samples.collections.Collections.Transformations.map
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.map(transform: (UShort) -> R): List<R> { return mapTo(ArrayList<R>(size), transform) }\n\n/**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @return The [result] list *
 * @sample samples.collections.Collections.Transformations.map
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.map(transform: (UByte) -> R): List<R> { return mapTo(ArrayList<R>(size), transform) }\n\n/**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @return The [result] list *
 * @sample samples.collections.Collections.Transformations.map
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.map(transform: (UShort) -> R): List<R> { return mapTo(ArrayList<R>(size), transform) }\n\n/**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @return The [result] list *
 * @sample samples.collections.Collections.Transformations.map
 */
/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R> ULongArray.map(transform: (ULong) -> R): List<R> {
 * return mapTo(ArrayList<R>(size), transform)
 * Returns a list containing the results of applying the given [transform] function * to each element in the original array.
 * @param [transform] function that takes the index of an element and the element itself * and returns the result of the transformation applied to the element.
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R> UByteArray.map(transform: (UByte) -> R): List<R> {
 * return mapTo(ArrayList<R>(size), transform)
 * Returns a list containing the results of applying the given [transform] function * to each element in the original array.
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R> UShortArray.map(transform: (UShort) -> R): List<R> {
 * return mapTo(ArrayList<R>(size), transform)
 * Returns a list containing the results of applying the given [transform] function * to each element in the original array.
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R> UIntArray.mapIndexed(transform: (index: Int, UInt) -> R): List<R> {
 * return mapIndexedTo(ArrayList<R>(size), transform)
 * Returns a list containing the results of applying the given [transform] function * to each element and its index in the original array.
 * @param [transform] function that takes the index of an element and the element itself * and returns the result of the transformation applied to the element.
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R> ULongArray.mapIndexed(transform: (index: Int, ULong) -> R): List<R> {
 * return mapIndexedTo(ArrayList<R>(size), transform)
 * Returns a list containing the results of applying the given [transform] function * to each element and its index in the original array.
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R> UByteArray.mapIndexed(transform: (index: Int, UByte) -> R): List<R> {
 * return mapIndexedTo(ArrayList<R>(size), transform)
 * Returns a list containing the results of applying the given [transform] function * to each element and its index in the original array.
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R> UShortArray.mapIndexed(transform: (index: Int, UShort) -> R): List<R> {
 * return mapIndexedTo(ArrayList<R>(size), transform)
 * Returns a list containing the results of applying the given [transform] function * to each element and its index in the original array.
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R, C : MutableCollection<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {
 * var index = 0
 * for (item in this)
 * destination.add(transform(index++, item))
 * returns destination
 * Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R, C : MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {
 * var index = 0
 * for (item in this)
 * destination.add(transform(index++, item))
 * returns destination
 * Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R, C : MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {
 * var index = 0
 * for (item in this)
 * destination.add(transform(index++, item))
 * returns destination
 * Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].
 */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * public inline fun <R, C : MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
 * var index = 0
 * for (item in this)
 * destination.add(transform(index++, item))
 * returns destination
 * Applies the given [transform] function to each element and its index in the original array * and appends the results to the given [destination].
 */
var index = 0
for (item in this)
    destination.add(transform(index++, item))
return

/*
 * Applies the given [transform] function to each element and its index in the original array
 * and appends the results to the given [destination].
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.withIndex(): Iterable<IndexedValue<UInt>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.withIndex(): Iterable<IndexedValue<ULong>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.withIndex(): Iterable<IndexedValue<UByte>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.all(predicate: (UInt) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.all(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.all(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.all(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}
samples.collections.Collections.Aggregates.all

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.all(predicate: (UShort) -> Boolean): Boolean {\n    for (element in this) if (!predicate(element)) return false\n    return true\n}\n
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.all(predicate: (UByte) -> Boolean): Boolean {\n    for (element in this) if (!predicate(element)) return false\n    return true\n}

/**
 * Returns \`true\` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.any(): Boolean {
    return storage.any()
}

/**
 * Returns \`true\` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.any(): Boolean {
    return storage.any()
}

/**
 * Returns \`true\` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.any(): Boolean {
    return storage.any()
}

/**
 * Returns \`true\` if at least one element matches the given \[predicate\].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.any(predicate: (UInt) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns \`true\` if at least one element matches the given \[predicate\].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.any(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns \`true\` if at least one element matches the given \[predicate\].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.any(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns \`true\` if at least one element matches the given \[predicate\].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.any(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns the number of elements matching the given \[predicate\].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.count(predicate: (UInt) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given \[predicate\].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.count(predicate: (ULong) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given \[predicate\].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.count(predicate: (UByte) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given \[predicate\].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.count(predicate: (UShort) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Accumulates value starting with \[initial\] value and
 * applying \[operation\] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting
with [initial] value and applying [operation] from left to right to current accumulator value and each element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
```

```kotlin
/*
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element with its index in the original array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.foldRight(initial: R, operation: (index: Int, acc: R, UInt) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.foldRight(initial: R, operation: (index: Int, acc: R, UInt) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.foldRight(initial: R, operation: (index: Int, acc: R, ULong) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```
accumulator = initial
while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
}

return accumulator

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.foldRight(initial: R, operation: (UByte, acc: R) -> R): R {

var index = lastIndex
var accumulator = initial
while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
--index
}

return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.foldRight(initial: R, operation: (UShort, acc: R) -> R): R {

var index = lastIndex
var accumulator = initial
while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
--index
}

return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.foldRightIndexed(initial: R, operation: (index: Int, UInt, acc: R) -> R): R {

var index = lastIndex
var accumulator = initial
while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
--index
}

return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.foldRightIndexed(initial: R, operation: (index: Int, ULong, acc: R) -> R): R {

var index = lastIndex
var accumulator = initial
while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
--index
}

return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.foldRightIndexed(initial: R, operation: (index: Int, UByte, acc: R) -> R): R {

var index = lastIndex
var accumulator = initial
while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
--index
}

return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left to each element and current accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly

var index = lastIndex
var accumulator = initial
while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
--index
}

return accumulator
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.forEach(action: (UInt) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.forEach(action: (ULong) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.forEach(action: (UByte) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.forEach(action: (UShort) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.forEach(action: (UByte) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.forEach(action: (UShort) -> Unit): Unit {
    for (element in this) action(element)
Performs the given [action] on each element.
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun
UShortArray.forEach(action: (UShort) -> Unit): Unit {\n  for (element in this) \n    action(element) * Performs the given [action] on each element, providing sequential index with the element.\n  * @param [action] function that takes the index of an element and the element itself.\n  * and performs the desired action on the element.\n*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun
UIntArray.forEachIndexed(action: (Int, UInt) -> Unit): Unit {\n  for (index in 0 until this.size) \n    action(index, this[index]) * Performs the given [action] on each element, providing sequential index with the element.\n  * @param [action] function that takes the index of an element and the element itself.\n  * and performs the desired action on the element.\n*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun
ULongArray.forEachIndexed(action: (Int, ULong) -> Unit): ULong? {\n  var max = 0L \n  for (index in 0 until this.size) \n    if (max < this[index]) \n      max = this[index] \n  return max * Performs the largest element or ‘null’ if there are no elements.\n*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun
UByteArray.forEachIndexed(action: (Int, UByte) -> Unit): UByte? {\n  var max = 0 \n  for (index in 0 until this.size) \n    if (max < this[index]) \n      max = this[index] \n  return max * Performs the largest element or ‘null’ if there are no elements.\n*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun
UShortArray.forEachIndexed(action: (Int, UShort) -> Unit): UShort? {\n  var max = 0 \n  for (index in 0 until this.size) \n    if (max < this[index]) \n      max = this[index] \n  return max * Performs the first element yielding the largest value of the given function or ‘null’ if there are no elements.\n*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun
Comparable\<R\>.maxBy(selector: (R) -> R): R {\n  if (isEmpty()) return null \n  var maxElem = this[0] \n  var lastIndex = this.lastIndex \n  if (lastIndex == 0) return maxElem \n  var maxVal = selector(maxElem) \n  for (i in 1..lastIndex) { \n    var e = this[i] \n    val v = selector(e) \n    if (maxVal < v) \n      maxElem = e \n      maxVal = v \n  } \n  return maxElem \n  * Performs the first element yielding the largest value of the given function or ‘null’ if there are no elements.\n*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\@public inline fun
Comparable\<R\>.maxBy(selector: (R) -> R): R {\n  if (isEmpty()) return null \n  var maxElem = this[0] \n  var lastIndex = this.lastIndex \n  if (lastIndex == 0) return maxElem \n  var maxVal = selector(maxElem) \n  for (i in 1..lastIndex) { \n    var e = this[i] \n    val v = selector(e) \n    if (maxVal < v) \n      maxElem = e \n      maxVal = v \n  } \n  return maxElem \n  * Performs the first element yielding the largest value of the given function or ‘null’ if there are no elements.
element yielding the largest value of the given function or `null` if there are no elements.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxBy
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.maxWith(comparator: Comparator<in UInt>): UInt? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
```

```kotlin
/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.maxWith(comparator: Comparator<in ULong>): ULong? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
```

```kotlin
/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.maxWith(comparator: Comparator<in UByte>): UByte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
```

```kotlin
/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
```
public fun UIntArray.minWith(comparator: Comparator<in UInt>): UInt? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

public fun ULongArray.minWith(comparator: Comparator<in ULong>): ULong? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

public fun UByteArray.minWith(comparator: Comparator<in UByte>): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

public fun UShortArray.minWith(comparator: Comparator<in UShort>): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

public fun UIntArray.none(): Boolean {
    return isEmpty()}

public fun ULongArray.none(): Boolean {
    return isEmpty()}

public fun UByteArray.none(): Boolean {
    return isEmpty()}

public fun UShortArray.none(): Boolean {
    return isEmpty()}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.minBy(selector: (ULong) -> R): ULong? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.minBy(selector: (UByte) -> R): UByte? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.minBy(selector: (UShort) -> R): UShort? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UIntArray.minWith(comparator: Comparator<in UInt>): UInt? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun ULongArray.minWith(comparator: Comparator<in ULong>): ULong? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UByteArray.minWith(comparator: Comparator<in UByte>): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UShortArray.minWith(comparator: Comparator<in UShort>): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/*
 * Returns `true` if the array has no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.none(): Boolean {
    return isEmpty()}

/*
 * Returns `true` if the array has no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.none(): Boolean {
    return isEmpty()}

/*
 * Returns `true` if the array has no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(): Boolean {
    return isEmpty()}

/*
 * Returns `true` if the array has no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(): Boolean {
    return isEmpty()}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.minBy(selector: (Int) -> Int): Int? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.minBy(selector: (Int) -> Int): Int? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.minBy(selector: (Int) -> Int): Int? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/*
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.minBy(selector: (Int) -> Int): Int? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}
```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.none(predicate: (UInt) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.none(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduce(operation: (acc: UInt, UInt) -> UInt): UInt {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

Open Source Used In DNAC Platform Antelope 8094
UnsupportedOperation<em>Exception("Empty array can't be reduced.")</em>

```kotlin
var accumulator = this[0]
for (index in 1..lastIndex) {
    accumulator = operation(index, accumulator, this[index])
}
return accumulator
}```

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\*\@public inline fun ULongArray.reduceIndexed<em>(operation: (Int, acc: ULong, ULong) -> ULong): ULong { if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\*\@public inline fun UByteArray.reduceIndexed<em>(operation: (Int, acc: UByte, UByte) -> UByte): UByte { if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and the element itself and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\*\@public inline fun UShortArray.reduceIndexed<em>(operation: (Int, acc: UShort, UShort) -> UShort): UShort { if (isEmpty()) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. Returns null if the array is empty.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.3")\@ExperimentalStdlibApi\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\*\@public inline fun UIntArray.reduceOrNull<em>(operation: (acc: UInt, UInt) -> UInt): UInt? { if (isEmpty()) return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. Returns null if the array is empty.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.3")\@ExperimentalStdlibApi\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\*\@public inline fun ULongArray.reduceOrNull<em>(operation: (acc: ULong, ULong) -> ULong): ULong? { if (isEmpty()) return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. Returns null if the array is empty.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.3")\@ExperimentalStdlibApi\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\*\@public inline fun UByteArray.reduceOrNull<em>(operation: (acc: UByte, UByte) -> UByte): UByte? { if (isEmpty()) return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. Returns null if the array is empty.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.3")\@ExperimentalStdlibApi\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\*\@public inline fun UShortArray.reduceOrNull<em>(operation: (acc: UShort, UShort) -> UShort): UShort? { if (isEmpty()) return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. Returns null if the array is empty.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.3")\@ExperimentalStdlibApi\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceRight(operation: (UInt, acc: UInt) -> UInt): UInt {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceRight(operation: (ULong, acc: ULong) -> ULong): ULong {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value.

 * @sample samples.collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value. The accumulator also includes its index in
the original array.

 * @param [operation] function that takes the index of an element, the element itself
* and current accumulator value, and calculates the
next accumulator value.

 * @sample samples.collections.Aggregates.reduceRightIndexed
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceRightIndexed(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with last element and applying [operation] from right to left
to each element and current accumulator value. The accumulator also includes its index in
the original array and current accumulator value.

 * @param [operation] function that takes the index of an element, the element itself
* and current accumulator value, and calculates the
next accumulator value.

 * @sample samples.collections.Aggregates.reduceRightIndexed
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceRightIndexed(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
accumulator value.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline func UByteArray.reduceRightIndexed(op: (index: Int, UByte, acc: UByte) -> UByte): UByte {
    var lastindex = lastIndex
    if (lastindex < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(lastindex)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline func UShortArray.reduceRightIndexed(op: (index: Int, UShort, acc: UShort) -> UShort): UShort {
    var lastindex = lastIndex
    if (lastindex < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(lastindex)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline func UIntArray.reduceRightOrNull(op: (UInt, acc: UInt) -> UInt): UInt? {
    var lastindex = lastIndex
    if (lastindex < 0) return null
    var accumulator = get(lastindex)
    while (index >= 0) {
        accumulator = operation(get(index), accumulator)
    }
    return accumulator
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline func ULongArray.reduceRightOrNull(op: (ULong, acc: ULong) -> ULong): ULong? {
    var lastindex = lastIndex
    if (lastindex < 0) return null
    var accumulator = get(lastindex)
    while (index >= 0) {
        accumulator = operation(get(index), accumulator)
    }
    return accumulator
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline func UByteArray.reduceRightOrNull(op: (UByte, acc: UByte) -> UByte): UByte? {
    var lastindex = lastIndex
    if (lastindex < 0) return null
    var accumulator = get(lastindex)
    while (index >= 0) {
        accumulator = operation(get(index), accumulator)
    }
    return accumulator
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline func UShortArray.reduceRightOrNull(op: (UShort, acc: UShort) -> UShort): UShort? {
    var lastindex = lastIndex
    if (lastindex < 0) return null
    var accumulator = get(lastindex)
    while (index >= 0) {
        accumulator = operation(get(index), accumulator)
    }
    return accumulator
}
```

* @param [operation] function that takes the index of an element, the element itself and current accumulator value, and calculates the next accumulator value. * @sample

```
Open Source Used In DNAC Platform Antelope 8097
```
value.

*\@* Since Kotlin (version 1.3) @ExperimentalStdlibApi @ExperimentalUnsignedTypes

The `UIntArray.scan` function serves a similar purpose to the `IntArray.scan` function, but it is designed for operations on `UIntArray` objects. It takes an initial value and an operation function as arguments. The operation function is applied to each element of the array along with the current accumulator value, which starts with the initial value. The result is a list containing successive accumulation values generated by applying the operation function from left to right. If the array is empty, the function returns a list containing just the initial value.

```kotlin
public inline fun <R> UIntArray.scan(initial: R, operation: (acc: R, UInt) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
        accumulator = operation(accumulator, element)
    result.add(accumulator)
    return result
}
```

The `ULongArray.scan` function is similar to the `UIntArray.scan` function but is designed for operations on `ULongArray` objects.

```kotlin
public inline fun <R> ULongArray.scan(initial: R, operation: (acc: R, ULong) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
        accumulator = operation(accumulator, element)
    result.add(accumulator)
    return result
}
```

The `UByteArray.scan` function is similar to the `UIntArray.scan` function but is designed for operations on `UByteArray` objects.

```kotlin
public inline fun <R> UByteArray.scan(initial: R, operation: (acc: R, UByte) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
        accumulator = operation(accumulator, element)
    result.add(accumulator)
    return result
}
```

The `UShortArray.scan` function is similar to the `UIntArray.scan` function but is designed for operations on `UShortArray` objects.

```kotlin
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
        accumulator = operation(accumulator, element)
    result.add(accumulator)
    return result
}
```

The `UIntArray.scanIndexed` function is similar to the `UIntArray.scan` function but also takes an index as an additional parameter. It is designed for operations on `UIntArray` objects where the index of each element is also considered.

```kotlin
public inline fun <R> UIntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices)
        accumulator = operation(index, accumulator, this[index])
    result.add(accumulator)
    return result
}
```

The `UByteArray.scanIndexed` function is similar to the `UIntArray.scanIndexed` function but is designed for operations on `UByteArray` objects.

```kotlin
public inline fun <R> UByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {
    if (isEmpty())
        return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices)
        accumulator = operation(index, accumulator, this[index])
    result.add(accumulator)
    return result
}
```

The `UShortArray.scanIndexed` function is similar to the `UIntArray.scanIndexed` function but is designed for operations on `UShortArray` objects.
should not be mutated; otherwise it would affect the previous value in resulting list.

*param [operation] function that takes the index of an element, current accumulator value\n and the element itself, and calculates the next accumulator value.\n
*param * @sample samples.collections.Collections.Aggregates.scan

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl
y
\n*

fun <R> ULongArray.scanIndexed(initial: R, operation: (Int, acc: R, ULong) -> R): List<R> {
    \n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}

*param * @sample samples.collections.Collections.Aggregates.scan

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl
y
\n*

fun <R> UByteArray.scanIndexed(initial: R, operation: (Int, acc: R, UByte) -> R): List<R> {
    \n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl
y
\n*

    \n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl
y
\n*

fun <R> UIntArray.scanReduce(operation: (acc: UInt, UInt) -> UInt): List<UInt> {
    \n    if (isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<UInt>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n        accumulator = operation(accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnl
y
\n*

fun <R> ULongArray.scanReduce(operation: (acc: ULong, ULong) -> ULong): List<ULong> {
    \n    if (isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<ULong>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n        accumulator = operation(accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result
}

Open Source Used In DNAC Platform Antelope 8099
values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. Takes current accumulator value and an element, and calculates the next accumulator value. samples.Collections.Aggregates.scanReduce

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
public inline fun UByteArray.scanReduce(operation: (acc: UByte, UByte) -> UByte): List<UByte> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UByte>(size).apply {
        add(accumulator) } for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. Takes current accumulator value and an element, and calculates the next accumulator value. samples.Collections.Aggregates.scanReduce

```
public inline fun UByteArray.scanReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UByte>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to
each element, its index in the original array and current accumulator value that starts with the first element of this
array.

* @param operation function that takes the index of an element, current accumulator value and the element itself, and
calculates the next accumulator value.
* @sample samples.collections.Aggregates.scanReduce
*/
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UShort>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
*/
public inline fun UIntArray.sumBy(selector: (UInt) -> UInt): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
*/
public inline fun ULongArray.sumBy(selector: (ULong) -> UInt): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
*/
public inline fun UByteArray.sumBy(selector: (UByte) -> UInt): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
*/
public inline fun UShortArray.sumBy(selector: (UShort) -> UInt): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
*/
public inline fun UIntArray.sumByDouble(selector: (UInt) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
*/
public inline fun ULongArray.sumByDouble(selector: (ULong) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
*/
public inline fun UByteArray.sumByDouble(selector: (UByte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
*/
public inline fun UShortArray.sumByDouble(selector: (UShort) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns a list of pairs built from the
elements of `this` array and the [other] array with the same index.

* The returned list has length of the shortest collection.

* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UIntArray.zip(other: Array<R>): List<Pair<UInt, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `ULongArray.zip(other: Array<R>): List<Pair<ULong, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UByteArray.zip(other: Array<R>): List<Pair<UByte, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UShortArray.zip(other: Array<R>): List<Pair<UShort, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UIntArray.zip(other: Iterable<R>): List<Pair<UInt, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `ULongArray.zip(other: Iterable<R>): List<Pair<ULong, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UByteArray.zip(other: Iterable<R>): List<Pair<UByte, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UShortArray.zip(other: Iterable<R>): List<Pair<UShort, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `inline` `fun` `<R, V>` `UIntArray.zip(other: Array<R>, transform: (a: UInt, b: R) -> V): List<V> {` `\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) { list.add(transform(this[i], other[i]))\n }\n return list\n}" `* Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterableWithTransform

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@@public` `inline` `fun` `<R, V>` `ULongArray.zip(other: Array<R>, transform: (a: ULong, b: R) -> V): List<V> {` `\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) { list.add(transform(this[i], other[i]))\n }\n return list\n}" `* Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterableWithTransform

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `inline` `fun` `<R, V>` `UByteArray.zip(other: Array<R>, transform: (a: UByte, b: R) -> V): List<V> {` `\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) { list.add(transform(this[i], other[i]))\n }\n return list\n}" `* Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterableWithTransform

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `inline` `fun` `<R, V>` `UShortArray.zip(other: Array<R>, transform: (a: UShort, b: R) -> V): List<V> {` `\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) { list.add(transform(this[i], other[i]))\n }\n return list\n}" `* Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterableWithTransform

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UIntArray.zip(other: Iterable<R>): List<Pair<UInt, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `ULongArray.zip(other: Iterable<R>): List<Pair<ULong, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UByteArray.zip(other: Iterable<R>): List<Pair<UByte, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

* `@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@public` `infix` `fun` `<R>` `UShortArray.zip(other: Iterable<R>): List<Pair<UShort, R>> {` `\n return zip(other) { t1, t2 -> t1 to t2 }\n}" `* Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n
* The returned list has length of the shortest collection.\n
* `@sample` samples.collections.Iterables.Operations.zipIterable

Open Source Used In DNAC Platform Antelope 8102
built from the elements of `this` collection and [other] array with the same index.
* The returned list has length of the shortest collection.

* @sample samples.collections.Iterables.Operations.zipIterable

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public infix fun <R> ULongArray.zip(other: ULongArray) {
    return zip(other) {
        t1, t2 -> t1 to t2
    }
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public infix fun <R> UByteArray.zip(other: UByteArray) {
    return zip(other) {
        t1, t2 -> t1 to t2
    }
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public infix fun <R> UShortArray.zip(other: UShortArray) {
    return zip(other) {
        t1, t2 -> t1 to t2
    }
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public infix fun <R> UIntArray.zip(other: UIntArray) {
    return zip(other) {
        t1, t2 -> t1 to t2
    }
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public infix fun <R> UByteArray.zip(other: UByteArray) {
    return zip(other) {
        t1, t2 -> t1 to t2
    }
}

Open Source Used In DNAC Platform Antelope 8103
List<Pair<ULong, ULong>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}  

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n */\n@sample collections.Operations.zip

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n */\n@sample collections.Operations.zip

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest array.\n */\n@sample collections.Operations.zip

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index using the provided \[transform\] function applied to each pair of elements.\n */\n@sample collections.Operations.zipWithTransform

/**
 * Returns the sum of all elements in the array.\n */\npublic fun UIntArray.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.\n */\npublic fun ULongArray.sum(): ULong {
    var sum: ULong = 0uL
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.\n */\npublic fun UByteArray.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.\n */\npublic fun UShortArray.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}
return sum

/**
 * Returns the sum of all elements in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.sum(): UInt { var sum = 0
    for (element in this) { sum += element
    } return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.sum(): ULong { var sum = 0
    for (element in this) { sum += element
    } return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UInt>.sum(): UInt {
    var sum = 0
    for (element in this) { sum += element
    } return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<ULong>.sum(): ULong {
    var sum = 0
    for (element in this) { sum += element
    } return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UByte>.sum(): UInt {
    var sum = 0
    for (element in this) { sum += element
    } return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UShort>.sum(): UInt {
    var sum = 0
    for (element in this) { sum += element
    } return sum
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: UInt, b: UInt): UInt {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: ULong, b: ULong): ULong {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: UByte): UInt {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: UShort): UInt {
    return if (a >= b) a else b
}

*/

/*
 * Copyright 2010-2019 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("UCollectionsKt")

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*
import kotlin.ranges.

/**
 * Returns an array of UByte containing all of the elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun <T> Collection<T>.toUByteArray(): UByteArray {
    val result = UByteArray(size)
    var index = 0
    for (element in this) { result[index++] = element
    } return result
}

/**
 * Returns an array of UInt containing all of the elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun <T> Collection<T>.toUIntArray(): UIntArray {
    val result = UIntArray(size)
    var index = 0
    for (element in this) { result[index++] = element
    } return result
}

/**
 * Returns an array of ULong containing all of the elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun <T> Collection<T>.toULongArray(): ULongArray {
    val result = ULongArray(size)
    var index = 0
    for (element in this) { result[index++] = element
    } return result
}

/**
 * Returns an array of UShort containing all of the elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun <T> Collection<T>.toUShortArray(): UShortArray {
    val result = UShortArray(size)
    var index = 0
    for (element in this) { result[index++] = element
    } return result
}

/**
 * Returns the sum of all elements in the collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun <T> Iterable<T>.sum(): T {
    var sum = 0
    for (element in this) { sum += element
    } return sum
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("UComparisonsKt")

package kotlin.comparisons

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: UInt, b: UInt): UInt {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: ULong, b: ULong): ULong {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: UByte): UInt {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: UShort): UInt {
    return if (a >= b) a else b
}

/*
 * Copyright 2010-2019 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("UComparisonsKt")

package kotlin.comparisons
```kotlin
/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: UByte, b: UByte): UByte {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun maxOf(a: UShort, b: UShort): UShort {
    return if (a >= b) a else b
}

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun maxOf(a: UInt, b: UInt, c: UInt): UInt {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun maxOf(a: ULong, b: ULong, c: ULong): ULong {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun minOf(a: UInt, b: UInt): UInt {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun minOf(a: ULong, b: ULong): ULong {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun minOf(a: UByte, b: UByte): UByte {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun minOf(a: UShort, b: UShort): UShort {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun minOf(a: UInt, b: UInt, c: UInt): UInt {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun minOf(a: ULong, b: ULong, c: ULong): ULong {
    return minOf(a, minOf(b, c))
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntRange.random(): UInt {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongRange.random(): ULong {
    return random(Random)
}

/**
 * Returns a random element from this range using the specified source of randomness.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntRange.random(random: Random): UInt {
    try {
        return random.nextUInt(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the specified source of randomness.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongRange.random(random: Random): ULong {
    try {
        return random.nextULong(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}
```
progression from this value down to the specified [to] value with the step -1. \n If the [to] value is greater than ‘this’ value the returned progression is empty. \n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public infix fun USHort.downTo(to: USHort): 
 UIntProgression{ \n return UIntProgression.fromClosedRange(this.toUInt(), to.toUInt(), -1)\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public fun UIntProgression.reversed(): 
 UIntProgression{ \n return UIntProgression.fromClosedRange(last, first, -step)\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public fun UIntProgression(step: Int): 
 UIntProgression{ \n checkStepIsPositive(step > 0, step)\n return UIntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public infix fun UIntProgression.step(step: Int): 
 UIntProgression{ \n if (this.step > 0) step else -step\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public fun UIntProgression.until(to: UInt): 
 UIntRange{ \n if (to <= UInt.MIN_VALUE) return UIntRange.EMPTY\n this.toUInt() .. (to - 1u).toUInt()\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public fun UIntProgression.reversed(): 
 UIntProgression{ \n return UIntProgression.fromClosedRange(this.toUInt(), to.toUInt(), -1)\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public fun USHort.coerceAtLeast(minimumValue: USHort): 
 USHort{ \n return if (this < USHort.MIN_VALUE) minimumValue else this\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public fun USHort.coerceAtLeast(minimumValue: USHort): 
 USHort{ \n return if (this < USHort.MIN_VALUE) minimumValue else this\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public fun UByte.coerceAtLeast(minimumValue: UByte): 
 UByte{ \n return if (this < UByte.MIN_VALUE) minimumValue else this\n }\n * \n @SinceKotlin("1.3")\n @ExperimentalUnsignedTypes\n @public fun UByte.coerceAtLeast(minimumValue: UByte): 
 UByte{ \n return if (this < UByte.MIN_VALUE) minimumValue else this\n }
samples.comparisons.ComparableOps.coerceAtLeastUnsigned
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShort.coerceAtLeast(minimumValue: UShort): UShort {
    return if (this < minimumValue) minimumValue else this
}  Ensures that this value is not greater than the specified [minimumValue].\n* \n* @return this value if it's less than or equal to the [minimumValue] or the [maximumValue] otherwise.\n* \n* @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UInt.coerceAtMost(maximumValue: UInt): UInt {
    return if (this > maximumValue) maximumValue else this
}  Ensures that this value is not greater than the specified [maximumValue].\n* \n* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n* \n* @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULong.coerceAtMost(maximumValue: ULong): ULong {
    return if (this > maximumValue) maximumValue else this
}  Ensures that this value is not greater than the specified [maximumValue].\n* \n* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n* \n* @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByte.coerceAtMost(maximumValue: UByte): UByte {
    return if (this > maximumValue) maximumValue else this
}  Ensures that this value is not greater than the specified [maximumValue].\n* \n* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n* \n* @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShort.coerceAtMost(maximumValue: UShort): UShort {
    return if (this > maximumValue) maximumValue else this
}  Ensures that this value is not greater than the specified [maximumValue].\n* \n* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n* \n* @sample samples.comparisons.ComparableOps.coerceAtMostUnsigned

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UInt.coerceIn(minimumValue: UInt, maximumValue: UInt): UInt {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this
}  Ensures that this value lies in the specified range [minimumValue], [maximumValue].\n* \n* @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceInUnsigned

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULong.coerceIn(minimumValue: ULong, maximumValue: ULong): ULong {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this
}  Ensures that this value lies in the specified range [minimumValue], [maximumValue].\n* \n* @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceInUnsigned

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this
}  Ensures that this value lies in the specified range [minimumValue], [maximumValue].\n* \n* @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceInUnsigned

*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this
}  Ensures that this value lies in the specified range [minimumValue], [maximumValue].\n* \n* @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceInUnsigned

Open Source Used In DNAC Platform Antelope 8109
is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].

@sample
samples.comparisons.ComparableOps.coerceInUnsigned

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@public fun UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {
    @sample
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum \$maximumValue\ is less than minimum \$minimumValue\.")
    return minimumValue + (this - minimumValue).min(maximumValue) - maximumValue
}

/**
 * Ensures that this value lies in the specified [range].
 * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`
 * @sample samples.comparisons.ComparableOps.coerceInUnsigned
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShort.coerceIn(range: ClosedRange<UShort>): UShort {
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: \$range\.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }
}

/**
 * Ensures that this value lies in the specified [range].
 * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`
 * @sample samples.comparisons.ComparableOps.coerceInUnsigned
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UInt.coerceIn(range: ClosedRange<UInt>): UInt {
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: \$range\.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }
}

/**
 * Ensures that this value lies in the specified [range].
 * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`
 * @sample samples.comparisons.ComparableOps.coerceInUnsigned
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULong.coerceIn(range: ClosedRange<ULong>): ULong {
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: \$range\.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }
}

/*
 * Copyright 2010-2019 JetBrains
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("USequencesKt")

import kotlin.random.*

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfUInt")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<UInt>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfULong")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<ULong>.sum(): ULong {
    var sum: ULong = 0uL
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfUByte")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<UByte>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfUShort")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<UShort>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/*
 * Copyright 2010-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS"
 * BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and limitations under the License.
 */

package kotlin.collections

import kotlin.internal.PlatformDependent
Returns an iterator over the elements of this collection that supports removing elements during iteration. *param T the type of element being iterated over. The mutable iterator is covariant on its type.

*public interface MutableIterable&lt;T&gt; { ... Returns an iterator over the elements of this collection that supports removing elements during iteration. *param T the type of element being iterated over. The mutable iterator is covariant on its type.

*public interface MutableList&lt;E&gt; { ... Returns an iterator over the elements of this sequence that supports removing elements during iteration. *param E the type of elements contained in the collection. The collection is covariant on its element type.

*public interface Collection&lt;E&gt; { ... A generic collection of elements. Methods in this interface support only read-only access to the collection. *param E the type of elements contained in the collection. The collection is covariant on its element type.

*public interface Iterable&lt;T&gt; { ... Returns an iterator over the elements of this object. *param T the type of element being iterated over. The iterator is covariant on its type.
between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).

* The returned list is backed by this list, so non-structural changes in the returned list are reflected in this list, and vice-versa.

* The elements are appended in the order they appear in the [elements] collection.

* Returns the number of key/value pairs in the map.

* Returns the value corresponding to the key.

* Returns the value corresponding to the given [key], or `null` if such
Open Source Used In DNAC Platform Antelope 8116

http://www.apache.org/licenses/LICENSE-2.0

*Unlicensed by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES
OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language

governing permissions and limitations under the License.
* The type with only one value: the 'Unit' object. This type corresponds to the 'void' type in Java.
* public object Unit { * you may not use this file except in compliance with the
License. * You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
* Unlicensed by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES
OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language

governing permissions and limitations under the License.

package kotlin

/**
 * The type with only
 * one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override
    fun toString() = "kotlin.Unit"
}

package kotlin.annotation

import kotlin.annotation.AnnotationTarget.*

/**
 * Contains the list of code elements which are the possible annotation targets
 */
public enum class AnnotationTarget {
    /**
     * Class, interface or object, annotation class is also included
     */
    CLASS,
    /**
     * Generic type parameter (unsupported yet)
     */
    TYPE_PARAMETER,
    /**
     * Field, including property's backing field
     */
    FIELD,
    /**
     * Local variable
     */
    LOCAL_VARIABLE,
    /**
     * Value parameter of a function or a constructor
     */
    VALUE_PARAMETER,
    /**
     * Constructor only (primary or secondary)
     */
    CONSTRUCTOR,
    /**
     * Function (constructors are not included)
     */
    FUNCTION,
    /**
     * Property getter only
     */
    PROPERTY_GETTER,
    /**
     * Property setter only
     */
    PROPERTY_SETTER,
    /**
     * Type usage
     */
    TYPE,
    /**
     * Any expression
     */
    EXPRESSION,
    /**
     * File
     */
    FILE,
    /**
     * Type alias
     */
    @SinceKotlin("1.1")
    TYPEALIAS
}

/**
 * Contains the list of possible annotation retentions.
 * Determines how an annotation is stored in binary output.
 */
public enum class AnnotationRetention {
    /**
     * Annotation isn't stored in binary output
     */
    SOURCE,
    /**
     * Annotation is stored in binary output, but invisible for reflection
     */
    BINARY,
    /**
     * Annotation is stored in binary output and visible for reflection (default
     * retention)
     */
    RUNTIME,
    /**
     * This meta-annotation indicates the code elements which are the possible targets of an annotation.
     * If the target meta-annotation is not present on an annotation declaration, the
     * annotation is applicable to the following elements:
     * 
     * [CLASS], [PROPERTY], [FIELD],
     * [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION], [PROPERTY_SETTER],
     * [PROPERTY_GETTER]
     */
    @Target(AnnotationTarget.ANNOTATION_CLASS)
    @MustBeDocumented
    public annotation class
    @property allowedTargets list of allowed annotation targets
    @property value @property necessary annotation retention (RUNTIME, BINARY or SOURCE)
    @property Repeatable
    @property MustBeDocumented
    public annotation class Retention(val value: AnnotationRetention = AnnotationRetention.RUNTIME)

    @Target(AnnotationTarget.ANNOTATION_CLASS)
    public annotation class Repeatable

    @Target(AnnotationTarget.ANNOTATION_CLASS)
    @property value necessary annotation retention (RUNTIME, BINARY or SOURCE)

    vararg val allowedTargets: AnnotationTarget
    @property allowedTargets list of allowed annotation targets
    @property value @property necessary annotation retention (RUNTIME, BINARY or SOURCE)
    @property value
    * This meta-annotation determines whether an annotation is stored in binary output and visible for reflection. By default, both are true. * If the target meta-annotation is not present on an annotation declaration, the annotation is applicable to the following elements:
    * [CLASS], [PROPERTY], [FIELD],
    * [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION], [PROPERTY_SETTER],
    * [PROPERTY_GETTER].

    * The type with only
    * one value: the `Unit` object. This type corresponds to the `void` type in Java.
    */

    * public object Unit {
    * you may not use this file except in compliance with the
License. * You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
* Unlicensed by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES
OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language

governing permissions and limitations under the License.

package kotlin

/**
 * The type with only
 * one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override
    fun toString() = "kotlin.Unit"
}

package kotlin

/**
 * The type with only
 * one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override
    fun toString() = "kotlin.Unit"
}

package kotlin

/**
 * The type with only
 * one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override
    fun toString() = "kotlin.Unit"
}

package kotlin

/**
 * The type with only
 * one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override
    fun toString() = "kotlin.Unit"
}

package kotlin

/**
 * The type with only
 * one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override
    fun toString() = "kotlin.Unit"
}

package kotlin

/**
 * The type with only
 * one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override
    fun toString() = "kotlin.Unit"
}
it's completely safe to use generic types as argument for such parameter.

```kotlin
@Target(AnnotationTarget.TYPE_PARAMETER)
@Retention(AnnotationRetention.BINARY)
internal annotation class PureReifiable

/**
 * Specifies that the corresponding built-in method exists depending on
 * Current implementation for JVM looks whether method with same JVM descriptor exists in the
 * module JDK.
 * For example MutableMap.remove(K, V) available only if corresponding method
 * method java/util/Map.remove(Ljava/lang/Object;Ljava/lang/Object;)Z is defined in JDK (i.e. for major versions >= 8)

@Target(AnnotationTarget.FUNCTION)
@Retention(AnnotationRetention.BINARY)
internal annotation class PlatformDependent

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file

/*
 * Copyright 2010-2018 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package kotlin.reflect

/**
 * Represents a parameter passed to a function or a property getter/setter,
 * including `this` and extension receiver parameters.
 */
public interface KParameter : KAnnotatedElement {
    /**
     * 0-based index of this parameter in the parameter list of its containing callable.
     */
    public val index: Int

    /**
     * Name of this parameter as it was declared in the source code,
     * or `null` if the parameter has no name or its name is not available at
     * runtime.

     * Examples of nameless parameters include `this` instance for member functions,
     * extension receiver for extension functions or properties, parameters of Java methods
     * compiled without the debug information, and others.

     * `null` public val name: String
     * Type of this parameter. For a `vararg` parameter, this is the type of the corresponding array.
     * not the individual element.

     * Kind of this parameter.

     * `null` public val kind: Kind
     */
}
public fun contravariant(type: KType): KTypeProjection =
    KTypeProjection(KVariance.IN, type)

/**
 * Creates a covariant projection of a given type, denoted by the `out` modifier applied to a type.
 * For example, in the type `Array<out Number>`, `out Number` is a covariant projection of the type of class `Number`.
 */
public fun covariant(type: KType): KTypeProjection =
    KTypeProjection(KVariance.OUT, type)

 KVariance {
    /**
     * The affected type parameter or type is *invariant*, which means it has no variance applied to it.
     */
    INVARIANT,

    /**
     * The affected type parameter or type is *contravariant*. Denoted by the `in` modifier in the source code.
     */
    IN,

    /**
     * The affected type parameter or type is *covariant*. Denoted by the `out` modifier in the source code.
     */
    OUT,

    /**
     * Visibility of declarations marked with the `public` modifier, or with no modifier at all.
     */
    PUBLIC,

    /**
     * Visibility of declarations marked with the `protected` modifier.
     */
    PROTECTED,

    /**
     * Visibility of declarations marked with the `internal` modifier.
     */
    INTERNAL,

    /**
     * Visibility of declarations marked with the `private` modifier.
     */
    PRIVATE
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@JsName("arrayIterator")
internal fun arrayIterator(array: dynamic, type: String?) = when (type) {
    null -> {
        val arr: Array<dynamic> = array
        object : Iterator<dynamic> {
            var index = 0
            override fun hasNext() = index < arr.size
            override fun next() = if (index < arr.size) arr[index++] else throw NoSuchElementException("$index")
        }
    }
    "BooleanArray" -> booleanArrayIterator(array)
    "ByteArray" -> byteArrayIterator(array)
    "ShortArray" -> shortArrayIterator(array)
    "CharArray" -> charArrayIterator(array)
    "IntArray" -> intArrayIterator(array)
    "LongArray" -> longArrayIterator(array)
    "FloatArray" -> floatArrayIterator(array)
    "DoubleArray" -> doubleArrayIterator(array)
    else -> throw IllegalStateException("Unsupported type argument for arrayIterator: $type")
}

@JsName("booleanArrayIterator")
internal fun booleanArrayIterator(array: BooleanArray) = object : BooleanIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextBoolean() = if (index < array.size) array[index++] else throw NoSuchElementException("$index")
}

@JsName("byteArrayIterator")
internal fun byteArrayIterator(array: ByteArray) = object : ByteArrayIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun next() = if (index < array.size) array[index++] else throw NoSuchElementException("$index")
}

//...
fun <T> primitiveArrayConcat(a: T, b: T): T {
    val args: Array<T> = js("arguments")
    if (a is Array<*> && a.asDynamic().`$type$` === undefined) {
        return concat(args)
    } else {
        var size = 0
        for (i in args.indices) {
            size += args[i].asDynamic().length as Int
        }
        val result = js("new a.constructor(size)")
        kotlin.copyArrayType(a, result)
        size = 0
        for (i in args.indices) {
            val arr = args[i].asDynamic()
            for (j in 0 until arr.length) {
                result[size++].asDynamic() = arr[j]
            }
        }
        return result
    }
}

fun booleanArrayOf() = withType("BooleanArray")

fun charArrayOf() = withType("CharArray", js("new Uint16Array([])"))

fun longArrayOf() = withType("LongArray", js("new LongArray([], 0)"))

fun withType(type: String, array: dynamic): dynamic {
    array.`$type$` = type
    return array
}

fun jsTypeOf(a: Any?): String = js("typeof a")

fun lazy(initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)

fun lazy(mode: LazyThreadSafetyMode, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)

fun lazy(lock: Any?, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)

fun fillFrom(src: dynamic, dst: dynamic): dynamic {
    val srcLen: Int = src.length
    val dstLen: Int = dst.length
    var index: Int = 0
    while (index < srcLen && index < dstLen) dst[index].asDynamic() = src[index++].asDynamic()
    return dst
}

fun arrayCopyResize(source: dynamic, newSize: Int, defaultValue: Any?): dynamic {
    val result = source.slice(0, newSize)
    copyArrayType(source, result)
    var index: Int = source.length
    if (newSize > index) {
        result.length = newSize
        while (index < newSize) result[index++].asDynamic() = defaultValue
    }
    return result
}

fun <T> arrayPlusCollection(array: dynamic, collection: Collection<T>): dynamic {
    val result = arrayPlusCollection(array, dynamic, collection, Collection<T>())
    return result
}

fun <T> fillFromCollection(dst: dynamic, startIndex: Int, collection: Collection<T>)

fun copyArrayType(source: Array<out Any>, result: Array<out Any>)

fun copyArrayGenericType(source: Array<out Any>, result: Array<out Any>, var index: Int)
```
@file:Suppress("UNCHECKED_CAST","RedundantVisibilityModifier")

internal object CompletedContinuation : Continuation<Any?> {
    override val context: CoroutineContext
        get() = error("This continuation is already complete")

    override fun resumeWith(result: Result<Any?>) {
        error("This continuation is already complete")
    }

    override fun toString(): String = "This continuation is already complete"
}

@SinceKotlin("1.3")
@PublishedApi
internal abstract class CoroutineImpl(private val resultContinuation: Continuation<Any?>) : Continuation<Any?> {
    protected var state = 0
    protected var exceptionState = 0
    protected var result: Any? = null
    protected var exception: Throwable? = null
    protected var finallyPath: Array<Int>?

    public override val context: CoroutineContext = resultContinuation.context

    private var intercepted_: Continuation<Any?>? = null

    public fun intercepted(): Continuation<Any?> =
            intercepted_
            ?: (context[ContinuationInterceptor] ?: interceptedContinuation(this) ?: this)
            .also { intercepted_ = it }

    override fun resumeWith(result: Result<Any?>) {
        var current = this
        var currentResult: Any? = result.getOrNull()
        var currentException: Throwable? = result.exceptionOrNull()

        // This loop unrolls recursion in current.resumeWith(param) to make saner and shorter stack traces on resume
        while (true) {
            currentResult = resultContinuation
            currentException = null
            try {
                val outcome = doResume()
                if (outcome === COROUTINE_SUSPENDED)
                    return
                currentResult = outcome
                currentException = null
            } catch (exception: dynamic) { // Catch all exceptions
                currentResult = null
                currentException = exception.unsafeCast<Throwable>()
            }

            releaseIntercepted() // this state machine instance is terminating
            if (completion is CoroutineImpl) {
                current = completion
            } else {
                if (currentException == null) {
                    this.result = currentResult
                } else {
                    state = exceptionState
                    exception = currentException
                }
                try {
                    val outcome = doResume()
                    if (outcome === COROUTINE_SUSPENDED)
                        return
                    currentResult = outcome
                    currentException = null
                } catch (exception: dynamic) { // Catch all exceptions
                    currentResult = null
                    currentException = exception.unsafeCast<Throwable>()
                }

                releaseIntercepted() // this state machine instance is terminating
                if (completion is CoroutineImpl) {
                    current = completion
                } else {
                    if (intercepted_ !== this)
                        completion.resumeWithException(intercepted_.exception)
                    return
                }
            }
        }
    }

    private fun releaseIntercepted() {
        val intercepted = intercepted_
        if (intercepted != null && intercepted !== this) {
            context[ContinuationInterceptor]!!.releaseInterceptedContinuation(intercepted)
        }
        this.intercepted_ = CompletedContinuation // just in case
    }

    protected abstract fun doResume(): Any?
}
```

public fun exceptionOrNull(): Throwable? =
    when (value) {
        is Failure -> value.exception
        else -> null
    }

/**
 * Returns a string \`Success(\$value)\` if this instance represents [success],
 * \`Failure(\$exception)\` if it is [failure], otherwise \`null\`.
 */
public val exceptionOrNull: String
    get() = when (value) {
        is Success -> "Success(\${value})"
        else -> "Failure(\${exception})"
    }

/**
 * Returns a string \`Success(\$value)\` if this instance represents
 * [success] and \`Failure(\$exception)\` if it is [failure], otherwise \`null\`.
 */
public override fun toString(): String =
    when (value) {
        is Success -> "Success(\${value})"
        else -> "Failure(\${exception})"
    }

    // companion with constructors

    /**
     * Companion object for [Result] class that contains its constructor functions
     * \[success\] and \[failure\].
     */
    public companion object {

        @InlineOnly
        @SinceKotlin("1.3")
        public inline fun <T> success(value: T): Result<T> =
            Result(value)

        @InlineOnly
        @SinceKotlin("1.3")
        public inline fun <T> failure(exception: Throwable): Result<T> =
            Result.createFailure(exception)

    internal class Failure(
        @JvmField
        val exception: Throwable
    ) : Serializable {
        override fun equals(other: Any?): Boolean = other is Failure && exception == other.exception
        override fun hashCode(): Int = exception.hashCode()
        override fun toString(): String = "Failure(\${exception})"
    }

    @PublishedApi
    @SinceKotlin("1.3")
    internal fun createFailure(exception: Throwable): Any =
        Result.Failure(exception)

    @PublishedApi
    @SinceKotlin("1.3")
    internal fun Result<*>.throwOnFailure() {
        if (value is Result.Failure) throw value.exception
    }

    @InlineOnly
    @SinceKotlin("1.3")
    public inline fun <R> runCatching(block: () -> R): Result<R> {
        return try {
            Result.success(block())
        } catch (e: Throwable) {
            Result.failure(e)
        }
    }

    @InlineOnly
    @SinceKotlin("1.3")
    public inline fun <T, R> T.runCatching(block: T.() -> R): Result<R> {
        return try {
            Result.success(block())
        } catch (e: Throwable) {
            Result.failure(e)
        }
    }

    // -- extensions ---

    /**
     * Returns the encapsulated value if this instance represents [success] or throws the
     * encapsulated [Throwable] exception if it is [failure].
     * Note, that this function rethrows any [Throwable] exception thrown by [onFailure] function.
     */
    @InlineOnly
    @SinceKotlin("1.3")
    public inline fun <T> Result<T>.getOrElse(onFailure: (exception: Throwable) -> R): R {
        contract {
            callsInPlace(onFailure, InvocationKind.AT_MOST_ONCE)
        }
        return when (val exception = exceptionOrNull()) {
            null ->
                value as T
            else -> onFailure(exception)
        }
    }

    /**
     * Returns the encapsulated value if this instance represents [success] or the
     * [defaultValue] if it is [failure].
     */
    @InlineOnly
    @SinceKotlin("1.3")
    public inline fun <T> Result<T>.getOrDefault(defaultValue: R): R {
        if (isFailure) return defaultValue
        return value as T
    }

    /**
     * Returns the result of [onSuccess] for the encapsulated value if this instance represents [success],
     * or the result of [onFailure] function for the encapsulated [Throwable] exception if it is [failure].
     * Note, that this function rethrows any [Throwable] exception thrown by [onSuccess] or [onFailure] function.
     */
    @InlineOnly
    @SinceKotlin("1.3")
    public inline fun <R, T : R> Result<T>.fold(onSuccess: (value: T) ->
R, onSuccess: (exception: Throwable) -> R
l
onFailure: (exception: Throwable) -> R |null contract { main callsInPlace(onSuccess, InvocationKind.AT_MOST_ONCE) callsInPlace(onFailure, InvocationKind.AT_MOST_ONCE) } |null return when (val exception = exceptionOrNull()) { null -> onSuccess(value as T) null else -> onFailure(exception) } |null } |null */ |null * Returns the encapsulated result of the given [transform] function applied to the encapsulated value
* if this instance represents [success][Result.isSuccess] or the
* original encapsulated [Throwable] exception if it is [failure][Result.isFailure]. * Note, that this function rethrows any [Throwable] exception thrown by [transform] function and encapsulates it as a failure.
* See [mapCatching] for an alternative that encapsulates exceptions.

Result<T>.map(transform: (value: T) -> R): Result<R> { main contract { main callsInPlace(transform, InvocationKind.AT_MOST_ONCE) } |null return when (null isSuccess -> Result.isSuccess(transform(value as T)) null else -> Result(value) } |null */ |null * Returns the encapsulated result of the given [transform] function applied to the encapsulated value
* if this instance represents [success][Result.isSuccess] or the
* original encapsulated [Throwable] exception if it is [failure][Result.isFailure]. * This function catches any [Throwable] exception thrown by [transform] function and encapsulates it as a failure.
* See [map] for an alternative that rethrows exceptions from `transform` function.

Result<T>.mapCatching(transform: (value: T) -> R): Result<R> { main return when (null isSuccess -> runCatching { transform(value as T) } null else -> Result(value) } |null */ |null * Returns the encapsulated result of the given [transform] function applied to the encapsulated [Throwable] exception
* if this instance represents [success][Result.isSuccess] or the
* original encapsulated value if it is [failure][Result.isFailure]. * Note, that this function rethrows any [Throwable] exception thrown by [transform] function and encapsulates it as a failure.
* See [recoverCatching] for an alternative that encases exceptions.

Result<T>.recover(transform: (exception: Throwable) -> R): Result<R> { main return when (null isSuccess -> runCatching { transform(exception) } null else -> Result(exception) } |null */ |null * Returns the encapsulated result of the given [transform] function applied to the encapsulated [Throwable] exception
* if this instance represents [failure][Result.isFailure] or the
* original encapsulated value if it is [success][Result.isSuccess]. * This function catches any [Throwable] exception thrown by [transform] function and encapsulates it as a failure.
* See [recover] for an alternative that rethrows exceptions.

Result<T>.onFailure(action: (exception: Throwable) -> Unit): Result<T> { main contract { main callsInPlace(action, InvocationKind.AT_MOST_ONCE) } |null return when (val exception = exceptionOrNull()) { null -> this
null else -> Result.isSuccess(transform(exception)) |null */ |null * Performs the given [action] on the encapsulated [Throwable] exception if this instance represents [failure][Result.isFailure]. * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.

Result<T>.onSuccess(action: (value: T) -> Unit): Result<T> { main contract { main callsInPlace(action, InvocationKind.AT_MOST_ONCE) } |null return this
null */ |null * Performs the given [action] on the encapsulated value
* if this instance represents [success][Result.isSuccess]. * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.

// -------------------
Result<T>)

/**
 * Classes and interfaces marked with this annotation are restricted when used as receivers
 for extension `suspend` functions. These `suspend` extensions can only invoke other member or extension
 `suspend` functions on this particular receiver and are restricted from calling arbitrary suspension functions.
 */
@Annotated
@Retention(AnnotationRetention.RUNTIME)
public annotation class RestrictsSuspension

/**
 * Resumes the execution of the corresponding coroutine passing [value] as the return value of the last suspension point.
 */
@Annotated
@Retention(AnnotationRetention.RUNTIME)
public inline fun <T> Continuation<T>.resumeWith(value: T): Unit =
    resumeWith(Result.success(value))

/**
 * Resumes the execution of the corresponding coroutine so that the [exception] is thrown right after the last suspension point.
 */
@Annotated
@Retention(AnnotationRetention.RUNTIME)
public inline fun <T> Continuation<T>.resumeWithException(exception: Throwable): Unit =
    resumeWith(Result.failure(exception))

/**
 * Creates a Continuation instance with the given [context] and implementation of [resumeWith] method.
 */
@Annotated
@Retention(AnnotationRetention.RUNTIME)
public inline fun <T> Continuation(
    context: CoroutineContext,
    crossinline resumeWith: (Result<T>) -> Unit
): Continuation<T> =
    object : Continuation<T> {
        override val context: CoroutineContext
            get() = context

        override fun resumeWith(result: Result<T>) =
            resumeWith(result)
    }

/**
 * Creates a coroutine without a receiver and with result type [T].
 */
@Annotated
@Retention(AnnotationRetention.RUNTIME)
public fun <T> (suspend () -> T).createCoroutine(
    completion: Continuation<T>
): Continuation<Unit> =
    SafeContinuation(createCoroutineUnintercepted(completion).intercepted(), COROUTINE_SUSPENDED)

/**
 * Starts a coroutine without a receiver and with result type [T].
 */
@Annotated
@Retention(AnnotationRetention.RUNTIME)
public fun <R, T> (suspend R.() -> T).createCoroutine(
    receiver: R,
    completion: Continuation<T>
): Continuation<Unit> =
    SafeContinuation(createCoroutineUnintercepted(receiver, completion).intercepted(), COROUTINE_SUSPENDED)

/**
 * Obtains the current continuation instance inside suspend functions and suspends
 */
@Annotated
@Retention(AnnotationRetention.RUNTIME)
public suspend inline fun <T> suspendCoroutine(crossinline block: (Continuation<T>) -> Unit): T =
    suspendCoroutineUninterceptedOrReturn {
        c: Continuation<T> ->
        val safe = SafeContinuation(c.intercepted())
        block(safe)
    }
safe.getOrThrow()  
}@SinceKotlin("1.3")@Suppress("WRONG_MODIFIER_TARGET")@InlineOnly
public suspend inline val coroutineContext: CoroutineContext
    get() {
        throw NotImplementedError("Implemented as intrinsic")
    }

/**
 * Returns the context of the current coroutine.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * @SinceKotlin("1.3")
 * @Suppress("WRONG_MODIFIER_TARGET")
 * @InlineOnly
 * @package kotlin.coroutines.intrinsics
 * @import kotlin.coroutines.*

 /**
 * Creates unintercepted coroutine with receiver type [R] and result type [T].
 * This function creates a new, fresh instance of suspendable computation every time it is invoked.
 * The [completion] continuation is invoked when coroutine completes with result or exception.
 * This function returns unintercepted continuation.
 * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of `on * both the coroutine and [completion] happens in the invocation context established by
 * [ContinuationInterceptor].
 */
@SinceKotlin("1.3")
public actual fun <R, T> (suspend R.() -> T).createCoroutineUnintercepted(completion: Continuation<T>): Any? =
    this.asDynamic()(receiver, completion, false)

/**
 * Creates unintercepted coroutine without receiver and with result type [T].
 * This function creates a new, fresh instance of suspendable computation every time it is invoked.
 * Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through
 * [ContinuationInterceptor].
 */
@SinceKotlin("1.3")
public actual fun <T> (suspend () -> T).createCoroutineUnintercepted(completion: Continuation<T>): Any? =
    this.asDynamic()(completion, false)

safe.getOrThrow()  
}@SinceKotlin("1.3")@Suppress("WRONG_MODIFIER_TARGET")@InlineOnly
public suspend inline val coroutineContext: CoroutineContext
    get() {
        throw NotImplementedError("Implemented as intrinsic")
    }

/**
 * Returns the context of the current coroutine.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * @SinceKotlin("1.3")
 * @Suppress("WRONG_MODIFIER_TARGET")
 * @InlineOnly
 * @package kotlin.coroutines.intrinsics
 * @import kotlin.coroutines.*

 /**
 * Creates unintercepted coroutine with receiver type [R] and result type [T].
 * This function creates a new, fresh instance of suspendable computation every time it is invoked.
 * The [completion] continuation is invoked when coroutine completes with result or exception.
 * This function returns unintercepted continuation.
 * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of `on * both the coroutine and [completion] happens in the invocation context established by
 * [ContinuationInterceptor].
 */
@SinceKotlin("1.3")
public actual fun <R, T> (suspend R.() -> T).createCoroutineUnintercepted(completion: Continuation<T>): Any? =
    this.asDynamic()(receiver, completion, false)

/**
 * Creates unintercepted coroutine without receiver and with result type [T].
 * This function creates a new, fresh instance of suspendable computation every time it is invoked.
 * The [completion] continuation is invoked when coroutine completes with result or exception.
 * This function returns unintercepted continuation.
 * Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through
 * [ContinuationInterceptor].
 */
@SinceKotlin("1.3")
public actual fun <T> (suspend () -> T).createCoroutineUnintercepted(completion: Continuation<T>): Any? =
    this.asDynamic()(completion, false)
[ContinuationInterceptor] that might be present in the completion’s [CoroutineContext]. It is the invoker’s responsibility to ensure that a proper invocation context is established. Note that [completion] of this function may get invoked in an arbitrary context. [Continuation.intercepted] can be used to acquire the intercepted continuation. Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of this coroutine and [completion] happens in the invocation context established by invoking [ContinuationInterceptor]. Repeated invocation of any resume function on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or exception. [suspend R.() -> T].createCoroutineUnintercepted(receiver: R, completion: Continuation<T>): Continuation<Unit> = if (this.asDynamic().length == 3) { // When ‘suspended’ is true the continuation is created, but not executed. this.asDynamic()(receiver, completion, true) } else { createCoroutineFromSuspendFunction(completion) } [ContinuationInterceptor].intercepted() can be used to acquire the intercepted continuation. Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of both the coroutine and [completion] happens in the invocation context established by invoking [ContinuationInterceptor]. Repeated invocation of any resume function on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or exception. createCoroutineUnintercepted(receiver: R, completion: Continuation<T>): Continuation<Unit> = // Kotlin/JS suspend lambdas have an extra parameter ‘suspended’ if (this.asDynamic().length == 3) { // When ‘suspended’ is true the continuation is created, but not executed. this.asDynamic()(receiver, completion, true) } else { createCoroutineFromSuspendFunction(completion) } [ContinuationInterceptor].intercepted() can be used to acquire the intercepted continuation. Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of both the coroutine and [completion] happens in the invocation context established by invoking [ContinuationInterceptor]. Repeated invocation of any resume function on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or exception. createCoroutineUnintercepted or [suspendCoroutineUninterceptedOrReturn]. This function shall be used on the immediate result of invoking [createCoroutineUnintercepted] or [suspendCoroutineUninterceptedOrReturn], in which case it checks for [ContinuationInterceptor] in the continuation’s [context]. [ContinuationInterceptor] invokes, caches and returns the result. If this function is invoked on other [Continuation] instances it returns ‘this’ continuation unchanged. Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. `suspend R.() -> T`.createCoroutineUnintercepted receiver: R, completion: Continuation<T>: Continuation<Unit> = ...
Double = js("Kotlin").doubleFromBits(bits).unsafeCast<Double>()

/**
 * Returns a bit representation of the specified floating-point value as [Int] * according to the IEEE 754 floating-point "single format" bit layout.
 * Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent, * so some [Float] values may overflow, underflow or loose their accuracy after conversion to bits and back.
 */
*@
 sinceKotlin("1.2") @ library("floatToBits") @ public actual fun Float.toBits(): Int = definedExternally

**@
 sinceKotlin("1.2") @ library("floatToRawBits") @ public actual fun Float.toRawBits(): Int = definedExternally

**@
 sinceKotlin("1.2") @@ kotlin.internal.InlineOnly @ public actual inline fun Float.Companion.fromBits(bits: Int): Float = js("Kotlin").floatFromBits(bits).unsafeCast<Float>()

internal inline fun Long(low: Int, high: Int) = js("Kotlin").Long.fromBits(low, high).unsafeCast<Long>()
internal inline val Long.low: Int get() = this.asDynamic().getLowBits().unsafeCast<Int>()
internal inline val Long.high: Int get() = this.asDynamic().getHighBits().unsafeCast<Int>()

package kotlin.text

/**
 * Returns a string representation of this [Long] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
sinceKotlin("1.2") @ library("longToString") @ public actual fun Long.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))

package kotlin.collections

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
import kotlin.js.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Reverses elements in the list in-place.
 */
sinceKotlin("1.3") @ experimentalUnsignedTypes @ public actual fun MutableList<T>.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

package kotlin.text

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
import kotlin.js.*

/**
 * Returns a character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char sequence.
 * @sample samples.collections.Collections.Elements.elementAt
 */
sinceKotlin("1.2") @ public actual fun CharSequence.elementAt(index: Int):Char {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, length: $length") }
}

package kotlin.collections

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
import kotlin.js.*

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
sinceKotlin("1.3") @ experimentalUnsignedTypes @ public actual fun UCharArray.elementAt(index: Int): UChar {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

package kotlin.collections

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
import kotlin.js.*

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
sinceKotlin("1.3") @ experimentalUnsignedTypes @ public actual fun ULongArray.elementAt(index: Int): ULong {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

package kotlin.collections

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
import kotlin.js.*

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
sinceKotlin("1.3") @ experimentalUnsignedTypes @ public actual fun UIntArray.elementAt(index: Int): UInt {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UByteArray.elementAt(index: Int): UByte {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @since 1.3
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UShortArray.elementAt(index: Int): UShort {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

/**
 * Returns a [List] that wraps the original array.
 * @since 1.3
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UIntArray.asList(): List<UInt> {
    return object : AbstractList<UInt>(), RandomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: UInt): Boolean = this@asList.contains(element)
        override fun get(index: Int): UInt {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: UInt): Int {
            if ((element as Any?) !is UInt) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: UInt): Int {
            if ((element as Any?) !is UInt) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}

/**
 * Returns a [List] that wraps the original array.
 * @since 1.3
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun ULongArray.asList(): List<ULong> {
    return object : AbstractList<ULong>(), RandomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: ULong): Boolean = this@asList.contains(element)
        override fun get(index: Int): ULong {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: ULong): Int {
            if ((element as Any?) !is ULong) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: ULong): Int {
            if ((element as Any?) !is ULong) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @since 1.3
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UByteArray.elementAt(index: Int): UByte {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @since 1.3
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UShortArray.elementAt(index: Int): UShort {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

/**
 * Returns a [List] that wraps the original array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UIntArray.asList(): List<UInt> {
    return object : AbstractList<UInt>(), RandomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: UInt): Boolean = this@asList.contains(element)
        override fun get(index: Int): UInt {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: UInt): Int {
            if ((element as Any?) !is UInt) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: UInt): Int {
            if ((element as Any?) !is UInt) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}

/**
 * Returns a [List] that wraps the original array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun ULongArray.asList(): List<ULong> {
    return object : AbstractList<ULong>(), RandomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: ULong): Boolean = this@asList.contains(element)
        override fun get(index: Int): ULong {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: ULong): Int {
            if ((element as Any?) !is ULong) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: ULong): Int {
            if ((element as Any?) !is ULong) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @since 1.3
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UByteArray.elementAt(index: Int): UByte {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @since 1.3
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UShortArray.elementAt(index: Int): UShort {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}
Json): JQuery = asDynamic().dialog(params)\n
@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery' package instead.")\n\npublic inline fun JQuery.dialog(mode: String, param: String): Any? = asDynamic().dialog(mode, param)\n
@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery' package instead.")\n\npublic inline fun JQuery.dialog(mode: String, value: Any?): JQuery = asDynamic().dialog(mode, value)\n
@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery' package instead.")\n\npublic inline fun JQuery.button(): JQuery = asDynamic().button()\n
@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery' package instead.")\n\npublic inline fun JQuery.accordion(): JQuery = asDynamic().accordion()\n
@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery' package instead.")\n\npublic inline fun JQuery.draggable(params: Json): JQuery = asDynamic().draggable(params)\n
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.js

import kotlin.annotation.AnnotationTarget.*

@Target(CLASS, FUNCTION, PROPERTY, CONSTRUCTOR, VALUE_PARAMETER, PROPERTY_GETTER, PROPERTY_SETTER)
@Deprecated("Use `external` modifier instead", level = DeprecationLevel.ERROR)\n\npublic annotation class native(public val name: String = "")

@Target(FUNCTION)
@Deprecated("Use inline extension function with body using dynamic")\n\npublic annotation class nativeGetter

@Target(FUNCTION)
@Deprecated("Use inline extension function with body using dynamic")\n\npublic annotation class nativeSetter

@Target(FUNCTION)
@Deprecated("Use inline extension function with body using dynamic")\n\npublic annotation class nativeInvoke

@Target(CLASS, FUNCTION, PROPERTY)
\ninternal annotation class library(public val name: String = "")

@Target(CLASS)
\ninternal annotation class marker

/**
 * Gives a declaration (a function, a property or a class) specific name in JavaScript.
 *
 * This may be useful in the following cases:
 *
 *   * There are two functions for which the compiler gives same name in JavaScript, you can mark one with `@JsName(...)` to
 *     prevent the compiler from reporting error.
 *   * You are writing a JavaScript library in Kotlin. The compiler produces mangled names for functions with parameters, which is unnatural for usual JavaScript developer.
 *   * For some reason you want to rename declaration, e.g. there's common term in JavaScript for a concept provided by the declaration, which in uncommon in Kotlin.
 *
 * Example:
 *
 * ``` kotlin
 * class Person(val name: String) {
 *     fun hello() {
 *         println("Hello $name!")
 *     }
 *
 *     @JsName("helloWithGreeting")
 *     fun hello(greeting: String) {
 *         println("$greeting $name!")
 *     }
 * }
 * ```
 *
 * @property name the name which compiler uses both for declaration itself and for all references to the declaration.
 *           It's required to denote a valid JavaScript identifier.
 *
 * @see

Open Source Used In DNAC Platform Antelope 8130
JsNonModule

@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)
public annotation class JsModule(val import: String)

@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)
public annotation class JsModule

/**
 * Denotes an `external` declaration that can be used without module system. By default, an `external` declaration is available regardless your target module system. However, by applying [JsModule] annotation you can make a declaration unavailable to *plain* module system. Some Java Script libraries are distributed both as a standalone downloadable piece of JavaScript and as a module available* as an npm package. To tell the Kotlin compiler to accept both cases, you can augment [JsModule] with the `@JsNonModule` annotation. For example:

```kotlin
@JsModule("jquery")
@JsNonModule
@JsName("$")
external abstract class JQuery() {
    // some declarations here
}

@JsModule("jquery")
@JsNonModule
@JsName("$")
external fun JQuery(element: Element): JQuery
```

@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)
public annotation class JsNonModule

/**
 * Adds prefix to `external` declarations in a source file.
 * JavaScript does not have concept of packages (namespaces). They are usually emulated by nested objects. The compiler turns references to `external` declarations either to plain unprefixed names (in case of *plain* modules) or to plain imports. However, if a JavaScript library provides its declarations in packages, you won't be satisfied with this.
 * You can tell the compiler to generate additional prefix before references to `external` declarations using the `@JsQualifier(...)` annotation. For example:

```kotlin
@JsModule("jquery")
@JsNonModule
@JsName("$")
@JsQualifier("my.jsPackageName")
external fun foo(x: Int): String
```

@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.FILE)
public annotation class JsQualifier(val value: String)

/**
 * Marks experimental JS export annotations.
 * Note that behaviour of these annotations will likely be changed in the future. Usages of such annotations will be reported as warnings unless an explicit opt-in with [OptIn] annotation, e.g. `@OptIn(ExperimentalJsExport::class)`, or with the `-Xopt-in=kotlin.js.ExperimentalJsExport` compiler option is given.

@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)
public annotation class ExperimentalJsExport

/**
 * Exports top-level declaration.
 * Used in future IR-based backend.
 * Has no effect in current JS backend.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.
 */

@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY_SETTER)
@Retention(AnnotationRetention.BINARY)
public annotation class Synchronized

@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)
public annotation class Volatile

* * *
var modified = false
for (element in elements) {
    if (add(element)) modified = true
}
return modified
}

actual override fun removeAll(elements: Collection<E>): Boolean = (this as MutableList<E>).removeAll { it in elements }

actual override fun retainAll(elements: Collection<E>): Boolean = removeAll {
    if (get(index) == element) {
        return index
    }
}
return -1
}

actual override fun lastIndexOf(element: E): Int {
    for (index in lastIndex downTo 0) {
        if (get(index) == element) {
            return index
        }
    }
    return -1
}

actual override fun indexOf(element: E): Int {
    for (index in 0..lastIndex) {
        if (get(index) == element) {
            return index
        }
    }
    return -1
}

actual override fun iterator(): MutableListIterator<E> = listIterator(0)

actual override fun listIterator(index: Int): MutableListIterator<E> = listIteratorImpl(index)

actual override fun listIterator(): MutableListIterator<E> = listIteratorImpl(0)

actual override fun removeAt(index: Int): E

abstract override fun add(index: Int, element: E): Unit

abstract override fun removeAt(index: Int): Unit

@JsName("toJSON")
open fun toJSON(): Any = this.toArray().toArray()
override fun previousIndex(): Int = index - 1

override fun add(element: E) { add(index, element) }  

override fun get(index: Int): E = AbstractList.checkElementIndex(index, _size) 

override fun removeAt(index: Int): E {
    AbstractList.checkElementIndex(index, _size) 
    val result = list.removeAt(fromIndex + index) 
    _size--
    return result
}

override fun set(index: Int, element: E): E {
    AbstractList.checkElementIndex(index, _size) 
    return list.set(fromIndex + index, element, element) 
}

override fun size: Int get() = _size

actual override fun clear() {
    entries.clear()
}

actual override fun put(key: K, value: V): V? {
}

actual override fun putAll(from: Map<out K, V>) {
    for ((key, value) in from) {
        put(key, value)
    }
}

override fun hasNext(): Boolean = entryIterator.hasNext()

override fun next(): K = entryIterator.next().key

override fun remove() = entryIterator.remove()

override fun add(element: K): Boolean = throw UnsupportedOperationException("Add is not supported on keys")

override fun clear() {
    this@AbstractMutableMap.clear()
}

override operator fun contains(element: K): Boolean = containsKey(element)

override operator fun iterator(): Iterator<K> = entryIterator.iterator()

override fun remove(element: K): Boolean {
    if (containsKey(element)) {
        val result = list.removeAt(fromIndex + index) 
        _size--
        return result
    }
    return false
}

override val size: Int get() = _size

actual override fun putAll(from: Collection<out K, V>) {
    for ((key, value) in from) {
        put(key, value)
    }
}

override fun hasPrevious(): Boolean = entryIterator.hasPrevious()

override fun previous(): K = entryIterator.previous().key

override fun remove() = entryIterator.remove()

override fun add(element: K): Boolean {
    if (containsKey(element)) {
        val result = list.removeAt(fromIndex + index) 
        _size--
        return result
    }
    return false
}

override fun clear() {
    entries.clear()
}

override fun putAll(from: Collection<out K, V>) {
    for ((key, value) in from) {
        put(key, value)
    }
}

override fun hasPrevious(): Boolean = entryIterator.hasPrevious()

override fun previous(): K = entryIterator.previous().key

override fun remove() = entryIterator.remove()

override fun add(element: K): Boolean {
    if (containsKey(element)) {
        val result = list.removeAt(fromIndex + index) 
        _size--
        return result
    }
    return false
}

override fun clear() {
    entries.clear()
}

override fun putAll(from: Collection<out K, V>) {
    for ((key, value) in from) {
        put(key, value)
    }
}

override fun hasPrevious(): Boolean = entryIterator.hasPrevious()

override fun previous(): K = entryIterator.previous().key

override fun remove() = entryIterator.remove()

override fun add(element: K): Boolean {
    if (containsKey(element)) {
        val result = list.removeAt(fromIndex + index) 
        _size--
        return result
    }
    return false
}

override fun clear() {
    entries.clear()
}

override fun putAll(from: Collection<out K, V>) {
    for ((key, value) in from) {
        put(key, value)
    }
}

override fun hasPrevious(): Boolean = entryIterator.hasPrevious()

override fun previous(): K = entryIterator.previous().key

override fun remove() = entryIterator.remove()

override fun add(element: K): Boolean {
    if (containsKey(element)) {
        val result = list.removeAt(fromIndex + index) 
        _size--
        return result
    }
    return false
}

override fun clear() {
    entries.clear()
}

override fun putAll(from: Collection<out K, V>) {
    for ((key, value) in from) {
        put(key, value)
    }
}

override fun hasPrevious(): Boolean = entryIterator.hasPrevious()

override fun previous(): K = entryIterator.previous().key

override fun remove() = entryIterator.remove()
hasNext(): Boolean = entryIterator.hasNext()

override fun remove(): V = entryIterator.remove().value

override fun remove(key: K): V? {
    val iter = entries.iterator()
    while (iter.hasNext()) {
        val entry = iter.next()
        if (key == entry.key) {
            val value = entry.value
            iter.remove()
            return value
        }
    }
    return null
}

actual override fun add(key: K, value: V): Boolean {
    val iter = entries.iterator()
    while (iter.hasNext()) {
        val entry = iter.next()
        if (key == entry.key) {
            return false
        }
    }
    val newEntry = entryIterator.next()
    entryIterator = entryIterator.iterator()
    entryIterator.add(newEntry.key, newEntry.value)
    return true
}

actual override fun contains(key: K): Boolean {
    return findEntry(key) != null
}

actual override fun containsAll(elements: Collection<K>): Boolean {
    return elements.all { contains(it) }
}

actual override fun equals(other: Any?): Boolean {
    if (this === other) return true
    if (other !is Collection<*>) return false
    return AbstractList.orderedEquals(this, other)
}

override fun hashCode(): Int = AbstractList.orderedHashCode(this)

actual actual override fun remove(key: K): V? {
    val iter = entries.iterator()
    while (iter.hasNext()) {
        val entry = iter.next()
        val k = entry.key
        if (key == k) {
            val value = entry.value
            iter.remove()
            return value
        }
    }
    return null
}


rangeCheck(index)
modCount++
return if (index == lastIndex)
array.asDynamic().splice(index, 1)

override fun remove(element: Any?): Boolean {
for (index in array.indices) {
if (array[index] == element) {
array.asDynamic().splice(index, 1)
modCount++
return true
}
}
return false
}

override fun removeRange(fromIndex: Int, toIndex: Int) {
modCount++
array.asDynamic().splice(fromIndex, toIndex - fromIndex)
actual override fun clear() {
array = emptyArray()
modCount++
}

actual override fun remove(element: E): Boolean {
for (index in array.indices) {
if (array[index] == element) {
array.asDynamic().splice(index, 1)
modCount++
return true
}
}
return false
}

override fun clear() {
array = emptyArray()
modCount++
}

actual override fun indexOf(element: E): Int = array.indexOf(element)

actual override fun lastIndexOf(element: E): Int = array.lastIndexOf(element)

override fun toString() = arrayToString(array)

override fun toArray(): Array<Any?> = js("["]").slice.call(array)

private fun rangeCheck(index: Int) = index.apply {
AbstractList.checkElementIndex(index, size)
}

private fun insertionRangeCheck(index: Int) = index.apply {
AbstractList.checkPositionIndex(index, size)
}

package kotlin.collections

internal fun sortArrayWith(array: Array<out T>, comparison: (T, T) -> Int) {
if (getStableSortingIsSupported()) {
array.asDynamic().sort(comparison)
} else {
mergeSort(array.unsafeCast<Array<T>>(0), 0, array.lastIndex, Comparator(comparison))
}

internal fun sortArrayWith(array: Array<out T>, comparator: Comparator<in T>) {
if (getStableSortingIsSupported()) {
val comparison = { a: T, b: T -> comparator.compare(a, b) }
array.asDynamic().sort(comparison)
} else {
mergeSort(array.unsafeCast<Array<T>>(0), 0, array.lastIndex, comparator)
}

internal fun<T> sortArray(array: Array<out T>) {
if (getStableSortingIsSupported()) {
val comparison = { a: T, b: T -> a.compareTo(b) }
array.asDynamic().sort(comparison)
} else {
mergeSort(array.unsafeCast<Array<T>>(0), 0, array.lastIndex, naturalOrder())
}
}

private var _stableSortingIsSupported: Boolean? = null
private fun getStableSortingIsSupported(): Boolean {
_stableSortingIsSupported?.let { return it }
_stableSortingIsSupported = false
val array = js("["]").unsafeCast<Array<Int>()
for (index in 0 until 600) {
array.asDynamic().push(index)
val comparison = { a: Int, b: Int -> (a and 3) - (b and 3) }
array.asDynamic().sort(comparison)
for (index in 1 until array.size) {
val a = array[index - 1]
val b = array[index]
if (a and 3 == (b and 3) && a >= b) return false
}
_stableSortingIsSupported = true
return true
}

private fun mergeSort(array: Array<T>, start: Int, end: Int, comparator: Comparator<in T>): Array<T> {
if (start == end) {
return array
}
val median = (start + end) / 2
val left = mergeSort(array, buffer, start, median, comparator)
val right = mergeSort(array, buffer, median + 1, end, comparator)
val target = if (left <= buffer) array else buffer
// Merge.
var leftIndex = start
var rightIndex = median + 1
for (i in start..end) {
val leftValue = left[leftIndex]
val rightValue = right[rightIndex]
if (comparator.compare(leftValue, rightValue) <= 0) {
val target[i] = leftValue
leftIndex++
} else {
val target[i] = rightValue
rightIndex++
}
}
return target
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.collections

@UseExperimental(ExperimentalUnsignedTypes::class)
@SinceKotlin("1.3")
@kotlin.js.JsName("contentDeepHashCodeImpl")
internal fun<T> Array<out T>.contentDeepHashCodeImpl(): Int {...
```java
var result = 1
for (element in this) {
    val elementHash = when {
        element == null -> 0
        isArrayish(element) -> (element.unsafeCast<Array<*>>()).contentDeepHashCodeImpl()
        element is UByteArray -> element.contentHashCode()
        element is UShortArray -> element.contentHashCode()
        element is UIntArray -> element.contentHashCode()
        element is ULongArray ->
        else -> element.hashCode()
    }

    result = 31 * result + elementHash
}
return result
} /*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.collections

internal interface EqualityComparator {
    /**
     * Subclasses must override to return a value indicating
     * whether or not two keys or values are equal.
     */
    abstract fun equals(value1: Any?, value2: Any?): Boolean

    /**
     * Subclasses must override to return the
     * hash code of a given key.
     */
    abstract fun getHashCode(value: Any?): Int

    object HashCode : EqualityComparator {
        override fun equals(value1: Any?, value2: Any?): Boolean = value1 == value2

        override fun getHashCode(value: Any?): Int = value?.hashCode() ?: 0
    }
} /*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */
/*
 * Based on GWT
 * Copyright 2008 Google Inc.
 */

package kotlin.collections

import kotlin.collections.MutableMap.MutableEntry

/**
 * Hash table based implementation of the [MutableMap] interface.
 * This implementation makes no guarantees regarding the order of enumeration of [keys], [values] and [entries] collections.
 * @param initialCapacity the initial capacity (ignored)
 * @param loadFactor      the load factor (ignored)
 * @throws IllegalArgumentException if the initial capacity or load factor are negative
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
actual constructor(initialCapacity: Int, loadFactor: Float = 0.0f) : this() {
        // This implementation of HashMap has no need of load factors or capacities.
        require(initialCapacity >= 0) { "Negative initial capacity: $initialCapacity" }

        require(loadFactor >= 0) { "Non-positive load factor: $loadFactor" }
    }

    /**
     * Constructs an instance of [HashMap] filled with the contents of the
     * specified [original] map.
     */
    actual constructor(original: Map<out K, V>) : this() {
        this.putAll(original)
    }

    actual override fun clear() {
        internalMap.clear()
郛StructureChanged(this@HashMap)
    }

    actual override fun containsKey(key: K): Boolean = internalMap.containsKey(key)

    actual override fun containsValue(value: V): Boolean = internalMap.any {
        equality.equals(it.value, value) }

    actual override fun containsEntry(element: MutableEntry<K, V>): Boolean = containsEntry(element)

    actual override operator fun get(key: K): V? = internalMap.get(key)

    actual override fun put(key: K, value: V): V? =
```
internalMap.put(key, value)\n\nactual override fun remove(key: K): V? = internalMap.remove(key)\n\nactual override val size: Int get() = internalMap.size\n\n/**\n * Constructs the specialized implementation of [HashMap] with [String] keys, which stores the keys as properties of\n * JS object without hashing them.\n */\n@unstable fun <V> stringMapOf(vararg pairs: Pair<String, V>): HashMap<String, V> {\n\n    return\n    HashMap<String, V>(InternalStringMap(EqualityComparator.HashCode)).apply { putAll(pairs).null }\n}/\n\nCopyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. */\n\n/*\n * Based on GWT HashSet\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\n/**\n * Constructs a new empty [HashSet].\n */\npublic fun <V> stringSetOf(vararg elements: String): HashSet<String> {
    return
    HashSet(stringMapOf<Any>()).apply { addAll(elements) }\n}/\n\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\nimport kotlin.collections.AbstractMutableMap.SimpleEntry\n\n/**\n * A simple wrapper around JavaScriptObject to provide [java.util.Map]-like semantics for any\n * key type.\n */\n@unstable\ninternal class InternalHashCodeMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {

    private var backingMap: dynamic = createJsMap()\n    override var size: Int = 0\n        private set \n
    override fun put(key: K, value: V): V? {
        val hashCode = equality.getHashCode(key)\n        val chainOrEntryOrNull = getChainOrEntryOrNull(hashCode)\n        if (chainOrEntryOrNull == null) {
            // This is a new chain, put it to the map.\n            backingMap[hashCode] = SimpleEntry(key, value)\n        } else {
            (chainOrEntryOrNull as Array<*>).findEntryInChain(key)\n        }
    }

    override fun clear() {\n        backingMap.clear()\n    }

    override fun contains(element: K): Boolean = backingMap.containsKey(element)\n
    override fun isEmpty(): Boolean = backingMap.isEmpty()\n
    override fun iterator(): MutableListIterator<K> = backingMap.keys.iterator()
private fun Array<MutableEntry<K, V>>.findEntryInChain(key: K): MutableEntry<K, V>? =
    firstOrNull { entry ->
        equality.equals(entry.key, key)
    }

override fun iterator(): MutableIterator<MutableEntry<K, V>> {
    return object : MutableIterator<MutableEntry<K, V>> {
        var state = -1 // -1 not ready, 0 - ready, 1 - done
        var chainOrEntry: dynamic = null
        var isChain = false
        var itemIndex = -1
        var lastEntry: MutableEntry<K, V>? = null

        private fun computeNext(): Int {
            if (chainOrEntry != null && isChain) {
                val chainSize: Int = chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>().size
                if (++itemIndex < chainSize)
                    return 0
            }

            if (++keyIndex < keys.size) {
                chainOrEntry = backingMap[keys[keyIndex]]
                isChain = chainOrEntry is Array<*>
                itemIndex = 0
                return 0
            } else {
                chainOrEntry = null
                return 1
            }
        }

        override fun hasNext(): Boolean {
            if (state == -1)
                state = computeNext()
            return state == 0
        }

        override fun next(): MutableEntry<K, V> {
            if (!hasNext()) throw NoSuchElementException()
            val lastEntry = if (isChain) {
                chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>()[itemIndex]
            } else {
                chainOrEntry.unsafeCast<MutableEntry<K, V>>()
            }
            this.lastEntry = lastEntry
            state = -1
            return lastEntry
        }

        override fun remove() {
            checkNotNull(lastEntry)
            this@InternalHashCodeMap.remove(lastEntry!!.key)
            lastEntry = null
        }
    }
}
and Kotlin Programming Language contributors. * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. * *

```kotlin
package kotlin.collections

import kotlin.collections.MutableMap.MutableEntry

/**
 * A simple wrapper around JavaScript Map for key type is string. * Though this map is instantiated only with K=String, the K type is not fixed to String statically, * because we want to have it erased to Any? in order not to generate type-safe override bridges for * [get], [contains], [remove] etc, if they ever are generated. * * An internal class InternalStringMap<K, V> override val equality: EqualityComparator : InternalMap<K, V> {

    private var backingMap: dynamic = createJsMap() override override var size : Int = 0

    override operator fun get(key: K): V? {
        if (key !is String) return null
        return backingMap[key] !== undefined ??

    override operator fun put(key: K, value: V): V? {
        require(key is String)
        val oldValue = backingMap[key]
        backingMap[key] = value

        if (oldValue === undefined) {
            size++
            return null
        } else {
            oldValue.unsafeCast<V>() return oldValue
        }

    override override fun clear() {
        backingMap = createJsMap()
        size = 0
    }

    override override fun iterator(): MutableIterator<MutableEntry<K, V>> {
        return object :
```

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. *
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. *
 * *
"
private inner class EntrySet : AbstractMutableSet<MutableEntry<K, V>>() {

    private inner class EntryIterator : MutableIterator<MutableEntry<K, V>> {
        // The last entry that was returned from this iterator.
        var last: ChainEntry<K, V>? = null

        // The next entry to return from this iterator.
        var next: ChainEntry<K, V>? = null

        init {
            next = head
            recordLastKnownStructure(map, this)
        }

        override fun hasNext(): Boolean {
            return next !== null
        }

        override fun next(): MutableEntry<K, V> {
            if (!hasNext()) throw NoSuchElementException()

            val current = next!!
            last = current
            next = current.nexttakeIf { it !== head }
            return current
        }

        override fun remove() {
            check(last != null)

            last!!.remove()
            map.remove(last!!.key)
            recordLastKnownStructure(map, this)
            last = null
        }
    }

    override fun add(element: MutableEntry<K, V>): Boolean = throw UnsupportedOperationException("Add is not supported on entries")

    override fun clear() {
        this@LinkedHashMap.clear()
    }

    override operator fun contains(element: MutableEntry<K, V>): Boolean = containsEntry(element)

    override operator fun iterator(): MutableIterator<MutableEntry<K, V>> = EntryIterator()

    override fun remove(element: MutableEntry<K, V>): Boolean {
        if (contains(element)) {
            this@LinkedHashMap.remove(element.key)
            return true
        }
        return false
    }

    override val size: Int get() = this@LinkedHashMap.size

    private var head: ChainEntry<K, V>? = null

    /**
     * Add this node to the end of the chain.
     */
    private fun ChainEntry<K, V>.addToEnd() {
        // This entry is not in the list.
        check(next == null && prev == null)

        val _head = head
        if (_head == null) {
            head = this
            next = this
            prev = this
        } else {
            // Chain is valid.
            val _tail = checkNotNull(_head.prev)
            // Update me.
            prev = _tail
            next = _head
            // Update my new siblings: current head and old tail.
            _head.prev = this
            _tail.next = this
        }
    }

    /**
     * Remove this node from the chain it is a part of.
     */
    private fun ChainEntry<K, V>.remove() {
        if (this.next === this) {
            // if this is single element, remove head
            head = null
        } else {
            if (head === this) {
                // if this is first element, move head to next
                head = next
            }
            next!!.prev = prev
            prev!!.next = next
        }
        next = null
        prev = null
    }

    /*
     * The hashmap that keeps track of our entries and the chain. Note that we
     * duplicate the key here to eliminate changes to HashMap and minimize the
     * code here, at the expense of additional space.
     */
    private val map: HashMap<K, ChainEntry<K, V>>

    /**
     * Constructs an empty [LinkedHashMap] instance.
     */
    constructor() : super()

    constructor(backingMap: HashMap<K, Any>) : super() {
        @Suppress("UNCHECKED_CAST") // expected to work due to erasure
        map = backingMap as HashMap<K, ChainEntry<K, V>>
    }

    constructor(initialCapacity: Int, loadFactor: Float = 0.0f) : super(initialCapacity, loadFactor) {
        map = HashMap<K, ChainEntry<K, V>>()
    }

    constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)

    constructor(original: Map<out K, V>) {
        map = HashMap<K, ChainEntry<K, V>>()
        this.putAll(original)
    }

    actual override fun clear() {
        map.clear()
        head = null
        override fun clone(): Any
            return LinkedHashMap(this)
    }

    actual override fun containsKey(key: K): Boolean
        return map.containsKey(key)

    actual override fun containsValue(value: V): Boolean {
        var node: ChainEntry<K, V> = head
        do {
            if (node.value == value) {
                return true
            }
            node = node.next!!
        } while (node != head)
        return false
    }

    actual override fun forEachBlock(block: (MutableEntry<K, V>) -> Unit) {
        if (map == null) return
        map.forEach { key, entry -> block(MutableEntry(key, entry.value)) }
    }

    actual override fun get(key: K): V?
        return map[key]

    actual override fun getOrDefault(key: K, defaultValue: V): V
        return map.getOrElse(key) { defaultValue }

    actual override fun remove(key: K): V?
        return map.remove(key)

    actual override fun remove(key: K, value: V): Boolean
        return map.remove(key, value)

    actual override fun put(key: K, value: V): V
        return map[key] = value

    actual override fun putAll(map: Map<out K, V>): Boolean
        return map.putAll(map)

    actual override fun putIfAbsent(key: K, value: V): Boolean
        return map.putIfAbsent(key, value)

    actual override fun putIfPresent(key: K, value: V): Boolean
        return map.putIfPresent(key, value)

    actual override fun size(): Int
        return map.size

    actual override fun values(): Set<V?>
        return map.values

    actual override fun values(): List<V?>
        return map.values.toList()

    actual override fun containsEntry(element: MutableEntry<K, V>): Boolean
        return map.containsKey(element.key)

    actual override fun containsValue(value: V) = containsEntry(MutableEntry(null, value))

    internal override fun linkedList(): LinkedList<MutableEntry<K, V>>
        return LinkedList<MutableEntry<K, V>>(map.values.toLinkedList())

    internal override fun linkedListIfAbsent(key: K) = linkedList()

    internal override fun linkedListIfPresent(key: K) = linkedList()

    override fun equals(other: Any?): Boolean
        return other is LinkedHashMap<K, V> && (map == other.map || map.contentEquals(other.map))

    override fun hashCode(): Int
        return map.hashCode()
createEntrySet(): MutableList<MutableMap.MutableEntry<K, V>> = EntrySet()

    actual override operator fun get(key: K): V? = map.get(key)?.value

    actual override fun put(key: K, value: V): V? {
        val old = map.get(key)
        if (old == null) {
            val newEntry = ChainEntry(key, value)
            map.put(key, newEntry)
            newEntry.addToEnd()
            return null
        } else {
            return old.setValue(value)
        }
    }

    actual override fun remove(key: K): V? {
        val entry = map.remove(key)
        if (entry != null) {
            entry.remove()
            return entry.value
        }
        return null
    }

    actual override val size: Int get() = map.size

}

/**
 * Constructs the specialized implementation of [LinkedHashMap] with [String]
 * keys, which stores the keys as properties of [String] object without hashing them.
 * */
public fun <V>
    linkedStringMapOf(vararg pairs: Pair<String, V>): LinkedHashMap<String, V> {
    return LinkedHashMap<String, V>(stringMapOf<Any>()).apply { putAll(pairs) }
}

/**
 * Copyright 2008 Google Inc.
 */
open fun linkedStringSetOf(vararg elements: String): LinkedHashSet<String> {
    return LinkedHashSet(linkedStringMapOf<Any>()).apply { addAll(elements) }
}

/**
 * @kotlin.internal.InlineOnly
 * */
actual inline fun <R> synchronized(lock: Any, block: () -> R): R {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return block()
}

/**
 * @kotlin.io
 * */
open class BaseOutput {
    open fun println() {
        print("\n")
    }

    open fun println(message: Any?) {
        print(message)
        println()
    }

    abstract fun print(message: Any?)

    open fun flush() {}
}

/** JsName used to make the declaration available outside of module to test it */
@JsName("NodeJsOutput")
internal class NodeJsOutput(val outputStream: dynamic) : BaseOutput() {
    override fun print(message: Any?) {
        // TODO: Using local
        val messageString = String(message)
        outputStream.write(messageString)
    }
}

/** JsName used to make the declaration available outside of module to test it */
@JsName("OutputToConsoleLog")
internal class OutputToConsoleLog : BaseOutput() {
    override fun print(message: Any?) {
        console.log(message)
    }

    override fun println(message: Any?) {
        console.log(message)
    }

    override fun println() {
        console.log("")
    }
}

/** JsName used to make the declaration available outside of module to test it */
@JsName("try.kotl.in")
internal class TryOutput : BaseOutput() {
    override fun println(message: Any?) {
        console.log(message)
    }

    override fun println() {
        console.log("")
    }
}


override fun print(message: Any?) {
    buffer += String(message)
}

override fun flush() {
    buffer = ""
}

/**
 * JsName used to make the declaration available outside of module to test it
 */
@JsName("BufferedOutputToConsoleLog")
internal class BufferedOutputToConsoleLog : BufferedOutput() {
    override fun print(message: Any?) {
        var s = String(message)
        val i = s.nativeLastIndexOf("\n", 0)
        if (i >= 0) {
            buffer += s.substring(0, i)
            flush()
            s = s.substring(i + 1)
        }
        buffer += s
    }

    override fun flush() {
        console.log(buffer)
        buffer = ""
    }
}

/**
 * JsName used to make the declaration available outside of module to test it and use at try.kotl.in
 */
@JsName("output")
internal var output = run {
    val isNode: Boolean = js("typeof process !== 'undefined' && process.versions && !!process.versions.node")
    if (isNode) NodeJsOutput(js("process.stdout")) else BufferedOutputToConsoleLog()
}

@kotlin.internal.InlineOnly
private inline fun String(value: Any?): String = js("String")(value)

/** Prints the line separator to the standard output stream. */
public actual fun println() {
    output.println()
}

/** Prints the given [message] and the line separator to the standard output stream. */
public actual fun println(message: Any?) {
    output.println(message)
}

/** Prints the given [message] to the standard output stream. */
public actual fun print(message: Any?) {
    output.print(message)
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.coroutines

import kotlin.coroutines.intrinsics.CoroutineSingletons.*
import kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED

@PublishedApi
@SinceKotlin("1.3")
internal actual class SafeContinuation<in T>
internal actual constructor(
    private val delegate: Continuation<T>,
    initialResult: Any?
) : Continuation<T> {
    @PublishedApi
    internal actual constructor(delegate: Continuation<T>) : this(delegate, UNDECIDED)

    public actual override val context: CoroutineContext
        get() = delegate.context

    private var result: Any? = initialResult

    public actual override fun resumeWith(result: Result<T>) {
        val cur = this.result
        when {
            cur === UNDECIDED -> {
                this.result = result.value
            }
            cur === COROUTINE_SUSPENDED -> {
                this.result = RESUMED
                delegate.resumeWith(result)
            }
            else -> throw IllegalStateException("Already resumed")
        }
    }

    @PublishedApi
    internal actual fun getOrThrow(): Any? {
        if (result === UNDECIDED) {
            result = COROUTINE_SUSPENDED
            return COROUTINE_SUSPENDED
        }
        val result = this.result
        return when {
            result === RESUMED
            delegate.resumeWith(result) } else -> throw IllegalStateException("Already resumed")
    }

    return COROUTINE_SUSPENDED
}

return COROUTINE_SUSPENDED // already called continuation, indicate COROUTINE_SUSPENDED upstream
result is Result.Failure -> throw result.exception
else -> result // either

result == COROUTINE_SUSPENDED
return COROUTINE_SUSPENDED // already called continuation, indicate COROUTINE_SUSPENDED upstream
result is Result.Failure -> throw result.exception
else -> result // either

result == COROUTINE_SUSPENDED or data

result == COROUTINE_SUSPENDED // already called continuation, indicate COROUTINE_SUSPENDED upstream
result is Result.Failure -> throw result.exception
else -> result // either

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/

package kotlin.js

/**
 * Exposes the [Date API](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date) to Kotlin.
 */
@Suppress("NOT_DOCUMENTED")
public external class Date() {
    public constructor(milliseconds: Number)

    public constructor(dateString: String)

    public constructor(year: Int, month: Int)

    public constructor(year: Int, month: Int, day: Int)

    public constructor(year: Int, month: Int, day: Int, hour: Int)

    public constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int)

    public constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int)

    public fun getDate(): Int

    public fun getDay(): Int

    public fun getFullYear(): Int

    public fun getHours(): Int

    public fun getMilliseconds(): Int

    public fun getMinutes(): Int

    public fun getSeconds(): Int

    public fun getMilliseconds(): Int

    public fun getMinutes(): Int

    public fun getSeconds(): Int

    public fun getTime(): Int

    public fun getTimezoneOffset(): Int

    public fun getUTCDate(): Int

    public fun getUTCDay(): Int

    public fun getUTCFullYear(): Int

    public fun getUTCMilliseconds(): Int

    public fun getUTCMinutes(): Int

    public fun getUTCHours(): Int

    public fun getUTCSeconds(): Int

    public fun setDate(): Int

    public fun setTime(): Int

    public fun setMilliseconds(): Int

    public fun setMinutes(): Int

    public fun setSeconds(): Int

    public fun toISOString(): String

    public fun toUTCString(): String

    @PublishedApi
    internal actual fun get() {
        return getUTCDate() + getUTCMilliseconds() / 1000
    }

    @PublishedApi
    internal actual fun getUTC() {
        return getUTCDate() + getUTCMilliseconds() / 1000
    }

    @PublishedApi
    internal actual fun toEpochTime(): Long
    return getUTC() * 1000
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.coroutines.js.internal

import kotlin.coroutines.Continuation
import kotlin.coroutines.EmptyCoroutineContext

@PublishedApi
@SinceKotlin("1.3")
internal val EmptyContinuation = Continuation<Any?>(EmptyCoroutineContext) { result ->
    result.getOrThrow()
}
getMinutes(): Int

public fun getMonth(): Int

public fun getSeconds(): Int

public fun getTime(): Double

public fun getTimezoneOffset(): Int

public fun getUTCDate(): Int

public fun getUTCDay(): Int

public fun getUTCFullYear(): Int

public fun getUTCMinutes(): Int

public fun getUTCMonth(): Int

public fun getUTCMilliseconds(): Int

public fun toDateString(): String

public fun toISOString(): String

public fun toJSON(): Json

public fun toLocaleDateString(locales: Array<String> = definedExternally, options: LocaleOptions = definedExternally): String

public fun toLocaleDateString(locales: String, options: LocaleOptions = definedExternally): String

public fun toLocaleString(locales: Array<String> = definedExternally, options: LocaleOptions = definedExternally): String

public fun toLocaleTimeString(locales: Array<String> = definedExternally, options: LocaleOptions = definedExternally): String

public fun toLocaleTimeString(locales: String, options: LocaleOptions = definedExternally): String

public fun toUTCString(): String

val now: Double = Date().time

val parse(dateString: String): Double

val UTC(year: Int, month: Int): Double

val UTC(year: Int, month: Int, day: Int): Double

val UTC(year: Int, month: Int, day: Int, hour: Int): Double

val UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int): Double


val LocaleOptions(): LocaleOptions = {}
\n\nreturn false;"/>*/n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/
*/n
@file:kotlin.jvm.JvmMultifileClass*/n@file:kotlin.jvm.JvmName("StringsKt")*/n
package
kotlin.text*/n

/**
 * Converts the string into a regular expression [Regex] with the default options.
 */
@kotlin.internal.InlineOnly*/n
public inline fun String.toRegex(): Regex = Regex(this)

/**
 * Converts the string into a regular expression [Regex] with the specified single [option].
 */
@kotlin.internal.InlineOnly*/n
public inline fun String.toRegex(option: RegexOption): Regex = Regex(this, option)

/**
 * Converts the string into a regular expression [Regex] with the specified set of [options].
 */
@kotlin.internal.InlineOnly*/n
public inline fun String.toRegex(options: Set<RegexOption>): Regex = Regex(this, options)

* * *
sequence from which a subsequence is appended. If [value] is `null`, then characters are appended as if [value] contained the four characters `\`null``. The inserted characters go in same order as in the [value] array, starting at [index].

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun StringBuilder.appendRange(value: CharSequence?, startIndex: Int, endIndex: Int): StringBuilder
```

```kotlin
/**
 * Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance.
 *
 * The inserted characters go in same order as in the [value] array, starting at [index].
 *
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.
 */
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder
```

```kotlin
/**
 * Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.
 *
 * The inserted characters go in same order as in the [value] character sequence, starting at [index].
 *
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.
 * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
 */
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public expect fun StringBuilder.insertRange(index: Int, value: CharSequence?, startIndex: Int, endIndex: Int): StringBuilder
```

```kotlin
@Suppress("EXTENSION_SHADOWED_BY_MEMBER")
@Deprecated("Use append(value: Any?) instead", ReplaceWith("append(value = obj)"), DeprecationLevel.WARNING)
@kotlin.internal.InlineOnly
public inline fun StringBuilder.append(obj: Any?): StringBuilder = this.append(obj)
```

```kotlin
/**
 * Builds new string by populating newly created [StringBuilder] using provided [builderAction] and then converting it to [String].
 */
```

```kotlin
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun buildString(capacity: Int, builderAction: StringBuilder.() -> Unit): String =
```

```kotlin
/**
 * Appends all arguments to the given [StringBuilder].
 */
public fun StringBuilder.append(vararg value: String?): StringBuilder {
    for (item in value)
        append(item)
    return this
}
```

```kotlin
/**
 * Appends all arguments to the given [StringBuilder].
 */
public fun StringBuilder.append(vararg value: Any?): StringBuilder {
    for (item in value)
        append(item)
    return this
}
```

```kotlin
package kotlin.dom
```

```kotlin
import org.w3c.dom.*
```

```kotlin
/**
 * Gets a value indicating whether this node is a TEXT_NODE or a CDATA_SECTION_NODE.
 */
public val Node.isText: Boolean
    get() = nodeType == Node.TEXT_NODE || nodeType == Node.CDATA_SECTION_NODE
```

```kotlin
/**
 * Gets a value indicating whether this node is an [Element].
 */
public val Node.isElement: Boolean
    get() = nodeType == Node.ELEMENT_NODE
```

```kotlin
/**
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * LICENSE.txt file. *\n *\n */
```

```kotlin
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the
```
String?) : this(message, null) {
    Error::class.js.asDynamic().call(this, message, null)
}

    actual constructor(cause: Throwable?) : this(cause)
{
    Error::class.js.asDynamic().call(this, cause)
}

    actual constructor(cause: Throwable?) : this(cause)
{
    Error::class.js.asDynamic().call(this, cause)
}

    actual constructor() : this(null)
{
    Error::class.js.asDynamic().call(this, null)
}

public actual open class Exception actual constructor(message: String?, cause: Throwable?) : Throwable(message, cause)
{
    Exception::class.js.asDynamic().call(this, message, cause)
}

public actual open class RuntimeException actual constructor(message: String?, cause: Throwable?) : Exception(message, cause)
{
    RuntimeException::class.js.asDynamic().call(this, message, cause)
}

public actual open class IllegalArgumentException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause)
{
    IllegalArgumentException::class.js.asDynamic().call(this, message, cause)
}

public actual open class IllegalStateException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause)
{
    IllegalStateException::class.js.asDynamic().call(this, message, cause)
}

public actual open class IndexOutOfBoundsException actual constructor(message: String?) : RuntimeException(message)
{
    IndexOutOfBoundsException::class.js.asDynamic().call(this, message)
}

public actual open class ConcurrentModificationException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause)
{
    ConcurrentModificationException::class.js.asDynamic().call(this, message, cause)
}

public actual open class UnsupportedOperationException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause)
{
    UnsupportedOperationException::class.js.asDynamic().call(this, message, cause)
}

public actual open class NumberFormatException actual constructor(message: String?) : IllegalArgumentException(message)
{
    NumberFormatException::class.js.asDynamic().call(this, message)
}

public actual open class NullPointerException actual constructor(message: String?) : RuntimeException(message)
{
    NullPointerException::class.js.asDynamic().call(this, message)
}

public actual open class ClassCastException actual constructor(message: String?) : RuntimeException(message)
{
    ClassCastException::class.js.asDynamic().call(this, message)
}

private actual constructor(message: String?, cause: Throwable?) : Error(message, cause)
{
    private constructor(message: String?, cause: Throwable?) : Error(message, cause)
}

actual constructor() : this()
{
    actual constructor() : this()
}

private constructor(message: String?, cause: Throwable?) : Error(message, cause)
{
    private constructor(message: String?, cause: Throwable?) : Error(message, cause)
}

public actual open class AssertionError private constructor(message: String?, cause: Throwable?) : Error(message, cause)
{
    private constructor(message: String?, cause: Throwable?) : Error(message, cause)
}

public actual open class NumberFormatException actual constructor(message: String?) : IllegalArgumentException(message)
{
    NumberFormatException::class.js.asDynamic().call(this, message)
}

public actual open class NullPointerException actual constructor(message: String?) : RuntimeException(message)
{
    NullPointerException::class.js.asDynamic().call(this, message)
}

public actual open class ClassCastException actual constructor(message: String?) : RuntimeException(message)
{
    ClassCastException::class.js.asDynamic().call(this, message)
}

actual constructor() : this()
{
    actual constructor() : this()
}

private constructor(message: String?, cause: Throwable?) : Error(message, cause)
{
    private constructor(message: String?, cause: Throwable?) : Error(message, cause)
}

public actual open class AssertionError actual constructor(message: String?, cause: Throwable?) : Error(message, cause)
{
    AssertionError::class.js.asDynamic().call(this, message, cause)
}

public actual open class AssertionError actual constructor(message: String?, cause: Throwable?) : Error(message, cause)
{
    AssertionError::class.js.asDynamic().call(this, message, cause)
}

public actual open class AssertionError actual constructor(message: String?, cause: Throwable?) : Error(message, cause)
{
    AssertionError::class.js.asDynamic().call(this, message, cause)
}

public actual open class AssertionError actual constructor(message: String?, cause: Throwable?) : Error(message, cause)
{
    AssertionError::class.js.asDynamic().call(this, message, cause)
}

package kotlin.collections

/**
 * Groups elements from the [Grouping] source by key and counts elements in each
 * group.
 */
public actual fun <T, K> Grouping<T, K>.eachCount(): Map<K, Int> =
    fold(0) {
        acc, _ -> acc + 1 }

/*
 * Groups elements from the [Grouping] source by key and counts elements in each
 * group.
 */
public actual fun <T, K> Grouping<T, K>.eachCount(): Map<K, Int> =
    fold(0) {
        acc, _ -> acc + 1 }
with the count of element in the group.

```kotlin
un */ @SinceKotlin("1.1")/*
un
```
@sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue


@Suppress("UNCHECKED_CAST")
aggregate { key, acc, e, first -> operation(key, if (first)
initialValueSelector(key, e) else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in the given [destination] map.
 * An initial value of accumulator is provided by [initialValueSelector] function.
 * @param initialValueSelector a function that provides an initial value of accumulator for each group.
 * It's invoked with parameters: `key` - the key of the group; `element` - the first element being encountered in that group.
 * If the [destination] map already has a value corresponding to some key, that value is used as an initial value of the accumulator for that group and the [initialValueSelector] function is not called for that group.
 * @param operation a function that is invoked on each element with the following parameters: `key` - the key of the group this element belongs to; `element` - the current value of the accumulator of the group; `acc` - the element from the source being accumulated.
 * @return the [destination] map associating the key of each group with the result of accumulating the group elements.
 * @sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValueTo
 */

@Suppress("UNCHECKED_CAST")
aggregateTo(destination) { key, acc, e, first -> operation(if (first)
initialValueSelector(key, e) else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
 * An initial value of accumulator is the same for each group.
 * @param operation a function that is invoked on each element with the following parameters: `acc` - the current value of the accumulator of the group; `element` - the element from the source being accumulated.
 * @return a [Map] associating the key of each group with the result of accumulating the group elements.
 * @sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue
 */

@Suppress("UNCHECKED_CAST")
aggregate { _, acc, e, first -> operation(if (first)
initialValue else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
 * An initial value of accumulator is the first element of the group.
 * @param operation a function that is invoked on each element with the following parameters: `key` - the key of the group this element belongs to; `element` - the current value of the accumulator of the group.
 * @return the [destination] map associating the key of each group with the result of accumulating the group elements.
 * @sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValueTo
 */

@Suppress("UNCHECKED_CAST")
aggregateTo(destination) { _, acc, e, first -> operation(if (first)
initialValue else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group sequentially starting from the second element of the group, passing the previously accumulated value and the current element as arguments, and stores the results in the given [destination] map.
 * An initial value of accumulator is the first element of the group.
 * @param operation a function that is invoked on each subsequent element of the group with the following parameters: `key` - the key of the group this element belongs to; `element` - the current value of the accumulator of the group.
 * @return the [destination] map associating the key of each group with the result of accumulating the group elements.
 * @sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue
 */
element from the source being accumulated.

@returns a [Map] associating the key of each group with the result of accumulating the group elements.

@sample samples.collections.Grouping.reduceByMaxVowels

@SinceKotlin("1.1")
    aggregate { key, acc, e, first ->
        @Suppress("UNCHECKED_CAST")
        if (first) e else operation(key, acc as S, e)
    }

/**
 * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group
 * sequentially starting from the second element of the group.
 * An initial value of accumulator is the first element of the group.
 * If the [destination] map already has a value corresponding to the key of some group
 * that value is used as an initial value of the accumulator for that group and the first element of that group is also
 * subjected to the [operation].
 * @param operation a function that is invoked on each subsequent element of the group with the following parameters:
 *   - `accumulator`: the current value of the accumulator of the group;
 *   - `element`: the source being folded;
 * @returns the [destination] map associating the key of each group with the result of accumulating the group elements.
 * @sample samples.collections.Grouping.reduceByMaxVowels
 */
@SinceKotlin("1.1")
public inline fun <S, T : S, K, M : MutableMap<in K, S>> Grouping<T, K>.reduceTo(destination: M,
    operation: (key: K, accumulator: S, element: T) -> S): M =
    aggregateTo(destination) { key, acc, e, first ->
        @Suppress("UNCHECKED_CAST")
        if (first) e else operation(key, acc as S, e)
    }

/**
 * Groups elements from the [Grouping] source by key and counts elements in each group to the given [destination] map.
 * An initial value of the counter for that group.
 * @returns the [destination] map associating the key of each group with the count of elements in the group.
 * @sample samples.collections.Grouping.groupingByEachCount
 */
@SinceKotlin("1.1")
public fun <T, K, M : MutableMap<in K, Int>> Grouping<T, K>.eachCountTo(destination: M): M =
    foldTo(destination, 0) { acc, _ -> acc + 1 }

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

/**
 */
@Deprecated("Use top-level functions from kotlin.math package instead.", level = DeprecationLevel.WARNING)
public external object Math {
    @Deprecated("Use kotlin.math.PI instead.", level = DeprecationLevel.ERROR)
    public val PI: Double
    @Deprecated("Use Random.nextDouble instead", ReplaceWith("kotlin.random.Random.nextDouble()", "kotlin.random"), level = DeprecationLevel.WARNING)
    public fun random(): Double
    @Deprecated("Use kotlin.math.abs instead.", ReplaceWith("abs(value)", "kotlin.math.abs"), level = DeprecationLevel.ERROR)
    public fun abs(value: Double): Double
    @Deprecated("Use kotlin.math.acos

instead.

ReplaceWith("acos(value)\", \"kotlin.math.acos\"), level = DeprecationLevel.ERROR)

public fun acos(value: Double): Double

@Deprecated("Use kotlin.math.atan instead.\", ReplaceWith("atan(value)\", \"kotlin.math.atan\"), level = DeprecationLevel.ERROR)

public fun atan(value: Double: Double)

@Deprecated("Use kotlin.math.atan2 instead.\", ReplaceWith("atan2(y, x)\", \"kotlin.math.atan2\"), level = DeprecationLevel.ERROR)

public fun atan2(y: Double, x: Double: Double)

@Deprecated("Use kotlin.math.cos instead.\", ReplaceWith("cos(value)\", \"kotlin.math.cos\"), level = DeprecationLevel.ERROR)

public fun cos(value: Double)

@Deprecated("Use kotlin.math.sin instead.\", ReplaceWith("sin(value)\", \"kotlin.math.sin\"), level = DeprecationLevel.ERROR)

public fun sin(value: Double)

@Deprecated("Use kotlin.math.exp instead.\", ReplaceWith("exp(value)\", \"kotlin.math.exp\"), level = DeprecationLevel.ERROR)

public fun exp(value: Double)

@Deprecated("Use maxOf or kotlin.math.max instead", level = DeprecationLevel.ERROR)

public fun max(vararg values: Int: Int)

@Deprecated("Use maxOf or kotlin.math.max instead", level = DeprecationLevel.ERROR)

public fun max(vararg values: Float: Float)

@Deprecated("Use maxOf or kotlin.math.max instead", level = DeprecationLevel.ERROR)

public fun max(vararg values: Double: Double)

@Deprecated("Use minOf or kotlin.math.min instead", level = DeprecationLevel.ERROR)

public fun min(vararg values: Int: Int)

@Deprecated("Use minOf or kotlin.math.min instead", level = DeprecationLevel.ERROR)

public fun min(vararg values: Float: Float)

@Deprecated("Use minOf or kotlin.math.min instead", level = DeprecationLevel.ERROR)

public fun min(vararg values: Double: Double)

@Deprecated("Use kotlin.math.sqrt instead.\", ReplaceWith("sqrt(value)\", \"kotlin.math.sqrt\"), level = DeprecationLevel.ERROR)

public fun sqrt(value: Double)

@Deprecated("Use kotlin.math.tan instead.\", ReplaceWith("tan(value)\", \"kotlin.math.tan\"), level = DeprecationLevel.ERROR)

public fun tan(value: Double)

@Deprecated("Use kotlin.math.ln instead.\", ReplaceWith("ln(value)\", \"kotlin.math.ln\"), level = DeprecationLevel.ERROR)

public fun log(value: Double)

@Deprecated("Use kotlin.math.pow instead.\", ReplaceWith("base.pow(exp)\", \"kotlin.math.pow\"), level = DeprecationLevel.ERROR)

public fun pow(base: Double, exp: Double)

@PublishedApi

internal fun trunc(value: Number: Number)

@PublishedApi

internal fun sinh(value: Double: Double)

@PublishedApi

internal fun cosh(value: Double: Double)

@PublishedApi

internal fun tanh(value: Double: Double)

@PublishedApi

internal fun asinh(value: Double: Double)

@PublishedApi

internal fun atanh(value: Double: Double)

@PublishedApi

internal fun hypot(x: Double: Double, y: Double: Double)

@PublishedApi

internal fun clz32(value: Int: Int)

/**
 * Returns the smaller of two values.
 */
@Suppress("DEPRECATION")
@Deprecated("Use minOf or kotlin.math.min instead", level = DeprecationLevel.ERROR)

public fun minOf(a: Long, b: Long: Long)

/**
 * Returns the greater of two values.
 */
@Suppress("DEPRECATION")
@Deprecated("Use maxOf or kotlin.math.max instead", level = DeprecationLevel.ERROR)

public fun maxOf(a: Long, b: Long: Long)

* Use of this source code is governed by the Apache 2.0 license that can be found in the

Open Source Used In DNAC Platform Antelope 8153

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
*/
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
 * Contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * LICENSE.txt file.
 */

package kotlin.js

/**
 * An interface for indexing access to a collection of
 * key-value pairs, where type of key is [String] and type of value is [Any?].
 */
public external interface Json {
    /**
     * Calls to the function will be translated to indexing operation (square brackets) on the receiver
     * with [propertyName] as the argument. E.g. for next code:
     * ```js
     * j["prop"] + j.get(p)
     * ```
     * will be generated:
     * ```js
     * function test(j, p) {
     *     return j["prop"] + j[p];
     * }
     * ```
     */
    operator fun get(propertyName: String): Any?

    /**
     * Calls of the function will be translated to an assignment of [value] to the receiver indexed
     * (with square brackets/index operation) with [propertyName].
     */
    operator fun set(propertyName: String, value: Any?): Unit
}

/**
 * Returns a simple JavaScript object (as [Json]) using provided key-value pairs as names and values of its properties.
 */
public fun json(vararg pairs: Pair<String, Any?>): Json {
    val res: dynamic = js("({})")
    for ((name, value) in pairs) {
        res[name] = value
    }
    return res
}

/**
 * Adds key-value pairs from [other] to [this]. Returns the original receiver.
 */
public fun Json.add(other: Json): Json {
    val keys: Array<String> = js("Object").keys(other)
    for (key in keys) {
        if (other.asDynamic().hasOwnProperty(key)) {
            this[key] = other[key]
        }
    }
    return this
}

/**
 */
@Suppress("NOT_DOCUMENTED")
public external object JSON {
    public fun stringify(o: Any?): String
    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)?): String
    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? = definedExternally, space: Int): String
    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? = definedExternally, space: String): String
    public fun stringify(o: Any?, replacer: Array<String>): String
    public fun stringify(o: Any?, replacer: Array<String>, space: Int): String
    public fun stringify(o: Any?, replacer: Array<String>, space: String): String
    public fun <T> parse(text: String): T
    public fun <T> parse(text: String, reviver: ((key: String, value: Any?) -> Any?)?): T
}

// region 
// region Double

/** Computes the sine of the angle [x] given in radians. Special cases:
 * * `sin(NaN)` is NaN
 */
public actual fun sin(x: Double): Double = nativeMath.sin(x)

/** Computes the cosine of the angle [x] given in radians. Special cases:
 * * `cos(NaN)` is NaN
 */
public actual fun cos(x: Double): Double = nativeMath.cos(x)

/** Computes the tangent of the angle [x] given in radians. Special cases:
 * * `tan(NaN)` is NaN
 */
public actual fun tan(x: Double): Double = nativeMath.tan(x)

/** Computes the arc cosine of the value [x] given in radians. Special cases:
 * * `acos(0)` is 0
 */
public actual fun acos(x: Double): Double = nativeMath.acos(x)

/** Computes the arc sine of the value [x] given in radians. Special cases:
 * * `asin(1)` is PI/2
 */
public actual fun asin(x: Double): Double = nativeMath.asin(x)

/** Computes the arc tangent of the value [x] given in radians. Special cases:
 * * `atan(0)` is 0
 */
public actual fun atan(x: Double): Double = nativeMath.atan(x)

// endregion
// endregion
'0.0' for \(x > 0\) and 'PI' for \(x < 0\)\n*  - \(\text{atan2}(0.0, 0.0)\) is '0.0' for \(x > 0\) and 'PI' for \(x < 0\)\n*  - \(\text{atan2}(y, +Inf)\) is '0.0' for \(x < y + Inf\) and '-PI' for \(x < y - Inf\)\n*  - \(\text{atan2}(y, -Inf)\) is 'PI' for \(x > y + Inf\) and '-PI' for \(x > y - Inf\)\n*  - \(\text{atan2}(0.0, y)\) is 'PI/2' for \(y > 0\) and '-PI/2' for \(y < 0\)\n*  - \(\text{atan2}(+Inf, x)\) is 'PI/2' for finite \(x\)\n*  - \(\text{atan2}(-Inf, x)\) is '-PI/2' for finite \(x\)\n*  - \(\text{atan2}(NaN, x)\) and \(\text{atan2}(y, NaN)\) is 'NaN'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun atan2(y: Double, x: Double): Double = nativeMath.atan2(y, x)\n
/**
 * Computes the hyperbolic sine of the value \[x\].\n*  Special cases:\n*  - '\sinh(NaN)' is 'NaN'\n*  - '\sinh(+Inf)' is '+Inf'\n*  - '\sinh(-Inf)' is '-Inf'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun sinh(x: Double): Double = nativeMath.sinh(x)\n
/**
 * Computes the hyperbolic cosine of the value \[x\].\n*  Special cases:\n*  - '\cosh(NaN)' is 'NaN'\n*  - '\cosh(+Inf)' is '+Inf'\n*  - '\cosh(-Inf)' is '-Inf'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun cosh(x: Double): Double = nativeMath.cosh(x)\n
/**
 * Computes the hyperbolic tangent of the value \[x\].\n*  Special cases:\n*  - '\tanh(NaN)' is 'NaN'\n*  - '\tanh(+Inf)' is '+1.0'\n*  - '\tanh(-Inf)' is '-1.0'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun tanh(x: Double): Double = nativeMath.tanh(x)\n
/**
 * Computes the inverse hyperbolic sine of the value \[x\].\n* Special cases:\n*  - '\asinh(NaN)' is 'NaN'\n*  - '\asinh(+Inf)' is '+Inf'\n*  - '\asinh(-Inf)' is '-Inf'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun asinh(x: Double): Double = nativeMath.asinh(x)\n
/**
 * Computes the inverse hyperbolic cosine of the value \[x\].\n*  Special cases:\n*  - '\acosh(NaN)' is 'NaN'\n*  - '\acosh(+Inf)' is '+Inf'\n*  - '\acosh(-Inf)' is '-Inf'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun acosh(x: Double): Double = nativeMath.acosh(x)\n
/**
 * Computes the inverse hyperbolic tangent of the value \[x\].\n*  Special cases:\n*  - '\atanh(NaN)' is 'NaN'\n*  - '\atanh(+Inf)' is '+1.0'\n*  - '\atanh(-Inf)' is '-1.0'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun atanh(x: Double): Double = nativeMath.atanh(x)\n
/**
 * Computes the positive square root of the value \[x\].\n*  Special cases:\n*  - '\sqrt(x)' is 'NaN' when \(x < 0\) or \(x\) is 'NaN'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun sqrt(x: Double): Double = nativeMath.sqrt(x)\n
/**
 * Computes the natural logarithm (base \(E\)) of the value \[x\].\n*  Special cases:\n*  - '\ln(NaN)' is 'NaN'\n*  - '\ln(+Inf)' is '+Inf'\n*  - '\ln(-Inf)' is '0.0'\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual inline fun log(x: Double, base: Double) = nativeMath.log(x)\n
/**
 * Computes the logarithm of the value \[x\] with base \(b\).\n*  See also logarithm functions for common fixed bases: [ln], [log10] and [log2].\n
*\@SinceKotlin("1.2")\@InlineOnly\n
public actual function log(x: Double, base: Double) = nativeMath.log(x) \n
**
nativeMath.log2(x)\ln(n)^{*n} \text{ Computes } \ln(x + 1) \\ln(n)^{*n} \text{ This function can be implemented to produce more precise result for } [x] \text{ near zero.} \\ln(n)^{*n} \text{ Special cases:} \ln(n)^{*n} \text{ - } \ln(p(NaN)) \text{ is } NaN \\ln(n)^{*n} \text{ - } \ln(p(x)) \text{ is } NaN \text{ where } x < -1.0 \\ln(n)^{*n} \text{ - } \ln(p(1.0)) \text{ is } -Inf \\ln(n)^{*n} \text{ - } \ln(p(Inf)) \text{ is } +Inf \\ln(n)^{*n} \text{ @see [ln] function} \ln(n)^{*n} \text{ @see [expm1] function} \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun ln1p(x: Double): Double = nativeMath.log1p(x)\ln(n)^{*n} \text{ Rounds the given value } [x] \text{ to an integer towards positive infinity.} \ln(n)^{*n} \text{ @return the smallest double value that is greater than or equal to the given value } [x] \text{ and is a mathematical integer.} \ln(n)^{*n} \text{ Special cases:} \ln(n)^{*n} \text{ - } `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer. \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun ceil(x: Double): Double = nativeMath.ceil(x).unsafeCast<Double>() // TODO: Remove unsafe cast after removing public js.math\ln(n)^{*n} \text{ Rounds the given value } [x] \text{ to an integer towards negative infinity.} \ln(n)^{*n} \text{ @return the largest double value that is smaller than or equal to the given value } [x] \text{ and is a mathematical integer.} \ln(n)^{*n} \text{ Special cases:} \ln(n)^{*n} \text{ - } `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer. \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun floor(x: Double): Double = nativeMath.floor(x).unsafeCast<Double>() // TODO: Remove unsafe cast after removing public js.math\ln(n)^{*n} \text{ Returns the absolute value of the given value } [x]. \ln(n)^{*n} \text{ Special cases:} \ln(n)^{*n} \text{ - } `abs(NaN)` is `NaN` \ln(n)^{*n} \text{ @see absoluteValue extension property for [Double] \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun abs(x: Double): Double = nativeMath.abs(x).unsafeCast<Double>() \ln(n)^{*n} \text{ Returns the sign of the given value } [x] \text{ towards the closest integer with ties rounded towards even integer.} \ln(n)^{*n} \text{ Special cases:} \ln(n)^{*n} \text{ - } `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer. \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun round(x: Double): Double = \ln(n)^{*n} \text{ if } (x \% 0.5 != 0.0) \text{ } \ln(n)^{*n} \text{ floor} = floor(x) \text{ } \ln(n)^{*n} \text{ return floor if } (floor \% 2 == 0.0) \text{ } \ln(n)^{*n} \text{ floor else ceil(x)} \ln(n)^{*n} \text{ Returns the smallest of two values.} \ln(n)^{*n} \text{ If either value is `NaN`, then the result is `NaN`.} \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun min(a: Double, b: Double): Double = nativeMath.min(a, b).unsafeCast<Double>() \ln(n)^{*n} \text{ Returns the greater of two values.} \ln(n)^{*n} \text{ If either value is `NaN`, then the result is `NaN`.} \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun max(a: Double, b: Double): Double = nativeMath.max(a, b).unsafeCast<Double>() \ln(n)^{*n} \text{ Raises this value to the power } [x]. \ln(n)^{*n} \text{ Special cases:} \ln(n)^{*n} \text{ - } `b.pow(0.0)` is `1.0` \ln(n)^{*n} \text{ - } `b.pow(1.0)` == `b` \ln(n)^{*n} \text{ - } `b.pow(NaN)` is `NaN` \ln(n)^{*n} \text{ - } `NaN.pow(x)` is `NaN` for `x != 0.0` \ln(n)^{*n} \text{ - } `b.pow(Inf)` is `NaN` for `abs(b) == 1.0` \ln(n)^{*n} \text{ - } `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun Double.pow(x: Double): Double = nativeMath.pow(this, x).unsafeCast<Double>() \ln(n)^{*n} \text{ Raises this value to the integer power } [n]. \ln(n)^{*n} \text{ See the other overload of [pow] for details.} \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun Double.pow(n: Int): Double = nativeMath.pow(this, n.toDouble()).unsafeCast<Double>() \ln(n)^{*n} \text{ Returns the absolute value of this value.} \ln(n)^{*n} \text{ Special cases:} \ln(n)^{*n} \text{ - } `NaN.absoluteValue` is `NaN` \ln(n)^{*n} \text{ @see abs function} \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline val Double.absoluteValue: Double = Double.abs(this).unsafeCast<Double>() \ln(n)^{*n} \text{ Returns this sign with the sign bit same as of the [sign] value.} \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual inline fun Double.withSign(sign: Int): Double = this.withSign(sign.toDouble()).unsafeCast<Double>() \ln(n)^{*n} \text{ Returns the ulp (unit in the last place) of this value.} \ln(n)^{*n} \text{ An ulp is a positive distance between this value and the next nearest [Double] value larger in magnitude.} \ln(n)^{*n} \text{ Special Cases:} \ln(n)^{*n} \text{ - } `NaN.ulp` is `NaN` \ln(n)^{*n} \text{ - } `x.ulp` is `+Inf` when `x` is `+Inf` or `-Inf` \ln(n)^{*n} \text{ - } `0.0.ulp` is `Double.MIN_VALUE` \ln(n)^{*n} \text{@SinceKotlin("1.2")}@InLineOnly@public actual val Double.ulp: Double = Double.withSign(0, this) \ln(n)^{*n} \text{ This is NaN if the ulp is not equal to zero.}
Double.MAX_VALUE -> this - this.nextDown()
else -> this.nextUp() - this

/**
 * Returns the [Double] value nearest to this value in direction of positive infinity.
 */
@SinceKotlin("1.2")
public actual fun Double.nextUp(): Double = when {
    this.isNaN() || this == Double.POSITIVE_INFINITY -> this
    this > 0 else -> Double.fromBits(this.toRawBits() + if (this > 0) 1 else -1)
}

/**
 * Returns the [Double] value nearest to this value in direction of negative infinity.
 */
@SinceKotlin("1.2")
public actual fun Double.nextDown(): Double = when {
    this.isNaN() || this == Double.NEGATIVE_INFINITY -> this
    this < 0 else -> Double.fromBits(this.toRawBits() + if (this < 0) -1 else 1)
}

/**
 * Returns the [Double] value nearest to this value in direction from this value towards the value [to].
 */
@SinceKotlin("1.2")
public actual fun Double.nextTowards(to: Double): Double = when {
    this.isNaN() || to.isNaN() -> Double.NaN
    to == this -> to
    to > this -> this.nextUp()
    else /* to < this */ -> this.nextDown()
}

/**
 * Rounds this [Double] value to the nearest integer and converts the result to [Int].
 */
@SinceKotlin("1.2")
public actual fun Double.roundToInt(): Int = when {
    isNaN() -> throw IllegalArgumentException("Cannot round NaN value.")
    this > Int.MAX_VALUE -> Int.MAX_VALUE
    this < Int.MIN_VALUE -> Int.MIN_VALUE
    else ->
        nativeMath.round(this).unsafeCast<Double>().toInt()
}

/**
 * Rounds this [Double] value to the nearest integer and converts the result to [Long].
 */
@SinceKotlin("1.2")
public actual fun Double.roundToLong(): Long = when {
    isNaN() -> throw IllegalArgumentException("Cannot round NaN value.")
    this > Long.MAX_VALUE -> Long.MAX_VALUE
    this < Long.MIN_VALUE -> Long.MIN_VALUE
    else ->
        nativeMath.round(this).unsafeCast<Double>().toLong()
}

// endregion

// region Float Math ========================================

/** Computes the sine of the angle [x] given in radians.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sin(x: Float): Float =
    nativeMath.sin(x.toDouble()).toFloat()

/** Computes the cosine of the angle [x] given in radians.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun cos(x: Float): Float =
    nativeMath.cos(x.toDouble()).toFloat()

/** Computes the tangent of the angle [x] given in radians.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun tan(x: Float): Float =
    nativeMath.tan(x.toDouble()).toFloat()

/** Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond to the rectangular coordinates `(x, y)` by computing the arc tangent of the value `y`/`x`.
 */
@SinceKotlin("1.2")
public actual fun atan2(y: Float, x: Float): Float =
    nativeMath.atan2(y.toDouble(), x.toDouble()).toFloat()
for $0 < y < +\infty$ and `-PT` for `-\infty < y < 0` such as `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(+Inf, x)` is `-\pi/2` for finite `x` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, +Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.

* `-atan2(y, 0.0)` is `-\pi/2` for `y > 0` and `-\pi/2` for `y < 0` * `-atan2(-Inf, x)` is `-\pi/2` for finite `x` * `-atan2(0.0, -Inf)` is `-\pi/2` for finite `x` * `-atan2(0.0, 0.0)` is `-\pi/2` for finite `x`.
\*\*\n@SinceKotlin("1.2\")\n@InlineOnly\npublic actual inline fun log2(x: Float): Float =
    nativeMath.log2(x.toDouble()).toFloat()

/**
 * Computes \(\log_2(x)\)
 * Special cases:
 * \(-\infty\) is `NaN` if \(x \leq 0\)
 * \(+0\) is `0.0` if \(x = 1\)
 * \(\pm\infty\) is `\pm\infty` if \(x > 0\)
 * 0 is `1.0` if \(x = 1\)
 * NaN is `NaN`
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log2(x: Float): Float =
    nativeMath.log2(x.toDouble()).toFloat()
converts the result to [Int].

* Ties are rounded towards positive infinity.

`x.roundToInt()` == Int.MAX_VALUE when `x > Int.MAX_VALUE`

`x.roundToInt()` == Int.MIN_VALUE when `x < Int.MIN_VALUE`

 postfix @SinceKotlin("1.2")@InlineOnly@public actual inline fun Float.roundToInt(): Int =
toDouble().roundToInt()@n@n@n

* Rounds this [Float] value to the nearest integer and converts the result to [Long].

* Ties are rounded towards positive infinity.

`x.roundToLong()` == Long.MAX_VALUE when `x > Long.MAX_VALUE`

`x.roundToLong()` == Long.MIN_VALUE when `x < Long.MIN_VALUE`

 postfix @SinceKotlin("1.2")@InlineOnly@public actual inline fun Float.roundToLong(): Long =
toDouble().roundToLong()@n@n@n

** Absolute Value Method

`abs(n)` is 'Int.MIN_VALUE' due to an overflow @n

`abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an overflow

** Min Method

`min(a, b)` is the smaller of two values.

** Max Method

`max(a, b)` is the greater of two values.

** Abs Method

`abs(n)` is the absolute value of the given value [n].

** Sign Method

`sign` is `-1` if the value is negative, `0` if the value is zero, and `1` if the value is positive.

** DoubleNaN

`isNaN()` returns `true` if the specified number is a Not-A-Number (NaN) value, `false` otherwise.

** DoubleInfinity

`isInfinite()` returns `true` if the specified number is not a finite number (NaN or infinity).

** DoubleNaN
does not throw an exception when `isNaN()` or `isInfinite()` called.

** DoubleIsNaN

Floating-point NaN values are not ordered, so they cannot be compared directly.

** DoubleCompare

Floating-point NaN values are not meaningful, so they cannot be compared directly.

** FloatNaN

Floating-point NaN values are not meaningful, so they cannot be compared directly.

** FloatIsNaN
does not throw an exception when `isNaN()` called.

** FloatInfinity

Floating-point infinity values are not meaningful, so they cannot be compared directly.

** FloatIsInfinity
does not throw an exception when `isInfinite()` called.

** FloatCompare

Floating-point infinity values are not meaningful, so they cannot be compared directly.

** FloatNaN
does not throw an exception when `isNaN()` called.

** FloatInfinity
does not throw an exception when `isInfinite()` called.

** FloatCompare

Floating-point infinity values are not meaningful, so they cannot be compared directly.
```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.countOneBits(): Int {
    // Hacker's Delight 5-1 algorithm
    var v = this
    v = (v and 0x55555555) + (v.ushr(1) and 0x55555555)
    v = (v and 0x33333333) + (v.ushr(2) and 0x33333333)
    v = (v and 0x0F0F0F0F) + (v.ushr(4) and 0x0F0F0F0F)
    v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)
    v = (v and 0x0000FFFF) + (v.ushr(16))
    return v
} // Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@Suppress("DEPRECATION")
@kotlin.internal.InlineOnly
public actual inline fun Int.countLeadingZeroBits(): Int = kotlin.js.Math.clz32(this)

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.countTrailingZeroBits(): Int =
    // Hacker's Delight 5-4 algorithm for expressing countTrailingZeroBits with countLeadingZeroBits
    Int.SIZE_BITS - (this or -this).inv().countLeadingZeroBits()

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.takeHighestOneBit(): Int =
    if (this == 0) 0 else 1 shl Int.SIZE_BITS - 1 - countLeadingZeroBits()

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.takeLowestOneBit(): Int =
    // Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit
    this and -this

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.rotateLeft(bitCount: Int): Int =
    shl(bitCount) or ushr(Int.SIZE_BITS - bitCount)

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Int.rotateRight(bitCount: Int): Int =
    shl(Int.SIZE_BITS - bitCount) or ushr(bitCount)

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Long.countOneBits(): Int =
    high.countOneBits() + low.countOneBits()

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Long.countLeadingZeroBits(): Int =
    when (val high = this.high) {
        0 -> Int.SIZE_BITS + low.countLeadingZeroBits()
        else -> high.countLeadingZeroBits(high) + low.countLeadingZeroBits()
    }

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Long.countTrailingZeroBits(): Int =
    when (val low = this.low) {
        0 -> Int.SIZE_BITS + high.countTrailingZeroBits(high) + low.countTrailingZeroBits()
    }

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Long.takeHighestOneBit(): Long =
    when (val high = this.high) {
        0 -> Int.SIZE_BITS + low.countLeadingZeroBits()
    }

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public actual fun Long.takeLowestOneBit(): Long =
    when (val low = this.low) {
        0 -> Long(0, high.takeLowestOneBit())
    }
```

Long(low.takeLowestOneBit(), 0)\n    }\n/*\n * Rotates the binary representation of this [Long] number left by the specified [bitCount]
 * number.rotateLeft(-n) == number.rotateRight(n)\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count: \n * number.rotateLeft(-n) == number.rotateRight(n)\n */
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic actual fun Long.rotateLeft(bitCount: Int): Long {
    if ((bitCount and 31) != 0) {\n        val low = this.low\n        val high = this.high\n        val newLow =\n            low.shl(bitCount) or high.ushr(-bitCount)\n        val newHigh = high.shl(bitCount) or low.ushr(-bitCount)\n        return if ((bitCount and 32) == 0) Long(newLow, newHigh) else Long(newHigh, newLow)\n    } else {\n        return if ((bitCount and 32) == 0) this else Long(high, low)\n    }\n}
false)

@JsName("getStarKTypeProjection")
internal fun getStarKTypeProjection(): KTypeProjection =
    KTypeProjection.STAR

@JsName("createCovariantKTypeProjection")
internal fun createCovariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.covariant(type)

@JsName("createInvariantKTypeProjection")
internal fun createInvariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.invariant(type)

@JsName("createContravariantKTypeProjection")
internal fun createContravariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.contravariant(type)

@JsName("getStarKTypeProjection")
internal fun getStarKTypeProjection(): KTypeProjection =
    KTypeProjection.STAR

@JsName("createCovariantKTypeProjection")
internal fun createCovariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.covariant(type)

@JsName("createInvariantKTypeProjection")
internal fun createInvariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.invariant(type)

@JsName("createContravariantKTypeProjection")
internal fun createContravariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.contravariant(type)

@JsName("getStarKTypeProjection")
internal fun getStarKTypeProjection(): KTypeProjection =
    KTypeProjection.STAR

@JsName("createCovariantKTypeProjection")
internal fun createCovariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.covariant(type)

@JsName("createInvariantKTypeProjection")
internal fun createInvariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.invariant(type)

@JsName("createContravariantKTypeProjection")
internal fun createContravariantKTypeProjection(type: KType): KTypeProjection =
    KTypeProjection.contravariant(type)

-------------------------------
@JsName("arrayClass")
val arrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<Array<*>>(), "Array", { it is Array<* } }

@JsName("stringClass")
val stringClass = PrimitiveKClassImpl(js("String").unsafeCast<JsClass<String>>(), "String", { it is String } }

@JsName("throwableClass")
val throwableClass = PrimitiveKClassImpl(js("Error").unsafeCast<JsClass<Throwable>>(), "Throwable", { it is Throwable } }

@JsName("booleanArrayClass")
val booleanArrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<BooleanArray>>(), "BooleanArray", { it is BooleanArray } }

@JsName("charArrayClass")
val charArrayClass = PrimitiveKClassImpl(js("Uint16Array").unsafeCast<JsClass<CharArray>>(), "CharArray", { it is CharArray } }

@JsName("byteArrayClass")
val byteArrayClass = PrimitiveKClassImpl(js("Int8Array").unsafeCast<JsClass<ByteArray>>(), "ByteArray", { it is ByteArray } }

@JsName("shortArrayClass")
val shortArrayClass = PrimitiveKClassImpl(js("Int16Array").unsafeCast<JsClass<ShortArray>>(), "ShortArray", { it is ShortArray } }

@JsName("intArrayClass")
val intArrayClass = PrimitiveKClassImpl(js("Int32Array").unsafeCast<JsClass<IntArray>>(), "IntArray", { it is IntArray } }

@JsName("longArrayClass")
val longArrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<LongArray>>(), "LongArray", { it is LongArray } }

@JsName("floatArrayClass")
val floatArrayClass = PrimitiveKClassImpl(js("Float32Array").unsafeCast<JsClass<FloatArray>>(), "FloatArray", { it is FloatArray } }

val functionClass(arity: Int): KClassImpl<Any> {
    return functionClasses.get(arity) ?: run {
        val result = PrimitiveKClassImpl(js("Function").unsafeCast<JsClass<Any>>(), "Function
            { jsTypeOf(it) === "function" && it.asDynamic().length === arity }
        functionClasses.asDynamic()[arity] = result
        result
    }
}
```
val metadata = jClass.asDynamic().`$metadata$`
return if (metadata != null) {
    if (metadata.`$kClass$` == null) {
        val kClass = SimpleKClassImpl(jClass)
        metadata.`$kClass$` = kClass
        kClass
    } else {
        metadata.`$kClass$`
    }
} else {
    SimpleKClassImpl(jClass)
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

/**
 */
@Suppress("NOT_DOCUMENTED")
public external class RegExp(pattern: String, flags: String? = definedExternally) {
    public fun test(str: String): Boolean
    public fun exec(str: String): RegExpMatch?
    public override fun toString(): String
    public var lastIndex: Int
    public val global: Boolean
    public val ignoreCase: Boolean
    public val multiline: Boolean
}

// TODO: Inherit from array or introduce asArray() extension
/**
 * Represents the return value of [RegExp.exec].
 */
@Suppress("NOT_DOCUMENTED")
public external interface RegExpMatch {
    public val index: Int
    public val input: String
    public val length: Int
}

/**
 * Returns the entire text matched by [RegExp.exec] if the [index] parameter is 0, or the text matched by each capturing parenthesis.
 */

/**
 * Converts the result of [RegExp.exec] to an array where the first element contains the entire matched text and each subsequent element is the text matched by each capturing parenthesis.
 */
public inline fun RegExpMatch.asArray(): Array<out String?> = unsafeCast<Array<out String?>>()

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.sequences

internal actual class ConstrainedOnceSequence<T> actual constructor(sequence: Sequence<T>) : Sequence<T> {
    private var sequenceRef: Sequence<T>? = sequence
    
    actual override fun iterator(): Iterator<T> {
        val sequence = sequenceRef ?: throw IllegalStateException("This sequence can be consumed only once.")
        sequenceRef = null
        return sequence.iterator()
    }
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.text

/**
 * A mutable sequence of characters.
 * String builder can be used to efficiently perform multiple string manipulation operations.
 */
public actual class StringBuilder actual constructor(content: String) : Appendable, CharSequence {
    /**
     * Constructs an empty string builder with the specified initial [capacity].
     * In Kotlin/JS implementation of StringBuilder the initial capacity has no effect on the further performance of operations.
     */
    actual constructor(capacity: Int) : this() {
        this.asDynamic()._capacity = capacity
    }

    /** Constructs a string builder that contains the same characters as the specified [content] char sequence. */
    actual constructor(content: String) : this(content) {
        this.asDynamic().capacity = capacity
    }

    actual override val length: Int
        get() = string.asDynamic().length
    
    actual override fun get(index: Int): Char
    
    actual override fun set(index: Int, value: Char) { throw IndexOutOfBoundsException("index: $index, length: $length") }
    
    actual override fun subSequence(startIndex: Int, endIndex: Int): CharSequence
    
    actual override fun append(value: Char): StringBuilder
    
    actual override fun append(value: CharSequence): StringBuilder
    
    actual override fun toString(): String
    
    actual override fun equals(other: Any?): Boolean { throw NotImplementedError("Not implemented in this kotlin.js module") }
    
    actual override fun hashCode(): Int { throw NotImplementedError("Not implemented in this kotlin.js module") }
    
    actual override fun asArray(): Array<out String?>
    
    actual override fun asArray(): Array<out CharSequence>
    
    actual override fun asArray(): Array<out Any>
}
```

Open Source Used In DNAC Platform Antelope 8166
@UseExperimental(ExperimentalStdlibApi::class)
actual override fun append(value: CharSequence?,
startIndex: Int, endIndex: Int): StringBuilder = this.appendRange(value, startIndex, endIndex)

* Returns the contents of this string builder and returns this instance.

* Surrogate pairs included in this string builder are treated as single characters.

* Therefore, the order of the high-low surrogates is never reversed.

* Note that the reverse operation may produce new surrogate pairs that were unpaired low-
surrogates and high-surrogates before the operation.

* For example, reversing `"\\uDC00\\uD800"` produces `"\\uD800\\uDC00"` which is a valid surrogate pair.

* actual fun reverse(): StringBuilder {
    var reversed = ""
    var index = string.length - 1
    while (index >= 0) {
        val low = string[index--]
        val high = string[index--]
        if (low.isLowSurrogate() && index >= 0) {
            reversed += reversed + high + low
        } else {
            reversed = reversed + high + low
        }
    }
    string = reversed
    return this
} /*
* Appends the string representation of the specified object [value] to this string builder and returns this instance.

* The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
* and then that string was appended to this string builder.

* * Returns the index within this string builder of the last occurrence of the specified [string].

* Returns `-1` if the specified [string] does not occur in this string builder.

* The last occurrence of empty string `""` is considered to be at the index equal to `this.length`.

* Returns `-1` if the specified [string] does not occur in this string builder.

* Returns the index within this string builder of the first occurrence of the specified [string].

* Returns `-1` if the specified [string] does not occur in this string builder.

* In Kotlin/JS implementation of StringBuilder the value returned from this method may not indicate the actual size of the backing storage.

* Returns the index within this string builder of the last occurrence of the specified [string].

* The last occurrence of empty string `""` is considered to be at the index equal to `this.length`.

* Returns `-1` if the specified [string] does not occur in this string builder.

* Returns the index within this string builder of the first occurrence of the specified [string].

* Returns `-1` if the specified [string] does not occur in this string builder.

* In Kotlin/JS implementation of StringBuilder the size of the backing storage is not extended to comply the given [minimumCapacity],
* thus calling this method has no effect on the further performance of operations.

* Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity].

* If the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.
* Otherwise, this method takes no action and simply returns.

* In Kotlin/JS implementation of StringBuilder the size of the backing storage is not extended to comply the given [minimumCapacity],
* thus calling this method has no effect on the further performance of operations.

* Returns the index within this string builder of the first occurrence of the specified [string].

* Returns `-1` if the specified [string] does not occur in this string builder.

* @SinceKotlin("1.3")
@ExperimentalStdlibApi
actual fun capacity(): Int = if (this.asDynamic().capacity !== undefined)
    maxOf(this.asDynamic().capacity, length) else length

* Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity].

* If the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.
* Otherwise, this method takes no action and simply returns.

* In Kotlin/JS implementation of StringBuilder the size of the backing storage is not extended to comply the given [minimumCapacity],
* thus calling this method has no effect on the further performance of operations.

* Returns the index within this string builder of the first occurrence of the specified [string].

* Returns `-1` if the specified [string] does not occur in this string builder.

* Returns the index within this string builder of the last occurrence of the specified [string].
occurrence of the specified [string].
   * starting from the specified [startIndex] toward the beginning.
   * Returns `-1` if the specified [string] does not occur in this string builder starting at the specified [startIndex].
   *
   @SinceKotlin("1.3")
   @ExperimentalStdlibApi
   actual fun lastIndexOf(string: String, startIndex: Int): Int {
        if (string.isEmpty() && startIndex < 0) return -1
        return this.string.asDynamic().lastIndexOf(string, startIndex)
    }

    /**
     * Inserts the specified characters 
     * [value] into this string builder at the specified [index] and returns this instance.
     *
     * The overall effect is exactly as if the
     * [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index] and returns this instance.
     *
     * @throws IndexOutOfBoundsException or [IllegalArgumentException] if
     * [index] is larger than the length of this string builder.
     *
     * @since 1.3
     * @throws IndexOutOfBoundsException or [IllegalArgumentException] if
     * [index] is less than zero or greater than the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
     * and then that string was inserted into this string builder at the specified [index].
     *
     * @throws IndexOutOfBoundsException if [value] is an empty sequence and [index] is less than
     * the length of this string builder.
     *
     * @param index the position in this string builder to insert at.
     * @param value the character sequence from which
     * characters are inserted. If [value] is `null`, then the four characters `"null"` are inserted.
     *
     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,
setLength(newLength: Int) {
    if (newLength < 0) {
        throw IllegalArgumentException("Negative new length: $newLength.")
    } else {
        for (i in length until newLength) {
            string += \u0000
        }
    }
}

/**
 * Returns a new [String] that contains characters in this string builder at [startIndex]
 * (inclusive) and up to the [length] (exclusive).
 */
@ExperimentalStdlibApi
actual fun substring(startIndex: Int): String {
    return string.substring(startIndex)
}

/**
 * Returns a new [String] that contains characters in this string builder at [startIndex]
 * and up to the [endIndex] (exclusive).
 */
@ExperimentalStdlibApi
actual fun substring(startIndex: Int, endIndex: Int): String {
    checkReplaceRange(startIndex, endIndex)
    return string.substring(startIndex, endIndex)
}

/**
 * Attempts to reduce storage used for this string builder.
 */
@ExperimentalStdlibApi
public fun clear(): StringBuilder {
    string = ""
    return this
}

/**
 * Sets the character at the specified [index] to the specified [value].
 */
@ExperimentalStdlibApi
public operator fun set(index: Int, value: Char) {
    string = string.substring(0, index) + value + string.substring(index + 1)
}

/**
 * Removes the character at the specified [index] from this string builder.
 */
@ExperimentalStdlibApi
public fun deleteAt(index: Int): StringBuilder {
    return string.substring(0, index) + string.substring(index + 1)
}

/**
 * Removes characters in the specified range from this string builder and
 * returns this instance.
 */
@ExperimentalStdlibApi
public fun setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder {
    checkReplaceRange(startIndex, endIndex)
    return string.substring(0, startIndex) + value + string.substring(endIndex)
}

private fun checkReplaceRange(startIndex: Int, endIndex: Int) {
    if (startIndex < 0 || startIndex > endIndex) {
        throw IndexOutOfBoundsException("startIndex: $startIndex, endIndex: $endIndex")
    }
}

val standex = StringBuilder(10)
standex += "Hello, World!"
standex += "\n"
standex.trimToSize()
endIndex'.

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun deleteRange(startIndex: Int, endIndex: Int): StringBuilder {
    string = string.substring(0, startIndex) + string.substring(endIndex)
    return this
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun appendRange(value: CharArray?, startIndex: Int, endIndex: Int): StringBuilder {
    val stringCsq = stringCsq.substring(startIndex, endIndex)
    string += stringCsq
    return this
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {
    AbstractList.checkPositionIndex(index, this.length)

    string = string.substring(0, index) + value + string.substring(index)
    return this
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {
    AbstractList.checkPositionIndex(startIndex, endIndex, this.length)

    string += value
    return this
}
```
@param startIndex the beginning (inclusive) of the subsequence to insert.
 * @param endIndex the end (exclusive) of the subsequence to insert.
 * @throws IndexOutOfBoundsException or
 * [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence
 * indices or when `startIndex > endIndex`.
 * @throws IndexOutOfBoundsException if [index] is less than zero or
 * greater than the length of this string builder.
 * @param index the index of `Char` to remove.
 * @throws IndexOutOfBoundsException when the range doesn't fit into the [destination] character array starting at the specified [destinationOffset],
 * or when that index is out of the [destination] array indices range.

@sinceKotlin("1.3")
@ExperimentalStdlibApi
public fun insertRange(index: Int, value: CharSequence?, startIndex: Int, endIndex: Int): StringBuilder {
    val stringCsq = value.toString()
    AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)
    return this
}

/**
 * Clears
 * the content of this string builder making it empty and returns this instance.
 * @sample
 * samples.text.Strings.clearStringBuilder
 */
@sinceKotlin("1.3")
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.clear(): StringBuilder = this.clear()

/**
 * Sets the character at the specified [index] to the specified [value].
 * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.
 */
@sinceKotlin("1.3")
@ExperimentalStdlibApi
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline operator fun StringBuilder.set(index: Int, value: Char) =
    this.set(index, value)

/**
 * Replaces characters in the specified range of this string builder with characters in
 * the specified string [value] and returns this instance.
 * @param startIndex the beginning (inclusive) of the range to replace.
 * @param endIndex the end (exclusive) of the range to replace.
 * @param value the string to replace with.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less
 * than zero, greater than the length of this string builder, or `startIndex > endIndex`.
 */
@sinceKotlin("1.3")
@ExperimentalStdlibApi
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.setRange(startIndex: Int, endIndex: Int,
    value: String): StringBuilder =
    this.setRange(startIndex, endIndex, value)

/**
 * Removes the character at the specified [index] from this string builder and returns this instance.
 * @param index the index of `Char` to remove.
 * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.
 */
@sinceKotlin("1.3")
@ExperimentalStdlibApi
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.deleteAt(index: Int): StringBuilder =
    this.deleteAt(index)

/**
 * Removes characters in the specified range from this string builder and returns this
 * instance.
 * @param startIndex the beginning (inclusive) of the range to remove.
 * @param endIndex the end (exclusive) of the range to remove.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of
 * range of this string builder indices or when `startIndex > endIndex`.
 */
@sinceKotlin("1.3")
@ExperimentalStdlibApi
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.deleteRange(startIndex: Int, endIndex: Int):
    StringBuilder =
    this.deleteRange(startIndex, endIndex)

/**
 * Copies characters from this string builder into the [destination] character array.
 * @param destination the array to copy to.
 * @param destinationOffset the position in the array to copy to, 0 by default.
 * @param destinationOffset the beginning (inclusive) of the range to copy, 0 by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out
 * of range of this string builder indices or when `startIndex > endIndex`.
 */
@sinceKotlin("1.3")
@ExperimentalStdlibApi
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0,
Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance.

*params* Characters are appended in order, starting at specified [startIndex].

*param startindex the beginning (inclusive) of the subarray to append.*

*param endindex the end (exclusive) of the subarray to append.*

*Throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range or not null.

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =
    this.appendRange(value, startIndex, endIndex)

/**
 * Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance.

*param value the character sequence from which a subsequence is appended.*

*Throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when [startIndex > endIndex].

```
radix for number to string conversion.

```kotlin
public actual fun Byte.toString(radix: Int): String = this.toInt().toString(radix)

public actual fun Short.toString(radix: Int): String = this.toInt().toString(radix)

public actual fun Int.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))

private fun String.isNaN(): Boolean = when (this.toLowerCase()) {
    "nan", "+nan", "-nan" -> true
    else -> false
}

private fun checkRadix(radix: Int): Int {
    if (radix !in 2..36) {
        throw IllegalArgumentException("radix $radix was not in valid range 2..36")
    }
    return radix
}

private fun digitOf(char: Char, radix: Int): Int = when {
    char >= '0' && char <= '9' -> char - '0'
    char >= 'A' && char <= 'Z' -> char - 'A' + 10
    char >= 'a' && char <= 'z' -> char - 'a' + 10
    else -> -1
}.let { if (it >= radix) -1 else it }

```

**Open Source Used In DNAC Platform Antelope**

---

The text above is a natural representation of the content of the document. It includes code blocks and explanations of the code, as well as some comments and documentation that provide context and usage examples. The document appears to be related to Kotlin programming, with examples of how to convert numbers to strings in different radices and handling cases like NaN values. The code snippets are interspersed with explanations and documentation comments, providing a comprehensive overview of the functionality being demonstrated. The natural text format helps in understanding the context and usage of the code, making it easier to follow and learn from.
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
    public actual fun findAll(input: CharSequence, start: Int = 0): Sequence<MatchResult> =
        generateSequence({ find(input, start) }, { match -> match.next() })
        return find(input) else return

    /**
     * Attempts to match the entire [input] CharSequence against the pattern.
     */
    public actual fun matchEntire(input: CharSequence): MatchResult? {
        return find(input) !== null
    }

    /**
     * Replaces all occurrences of this regular expression in the specified [input] string with specified [replacement] expression.
     */
    public actual fun replace(input: CharSequence, replacement: String): String =
        input.toString().nativeReplace(nativePattern, replacement)

    public actual inline fun replace(input: CharSequence, transform: (MatchResult) -> CharSequence): String {
        var match = find(input)
        if (match == null) return input.toString()

        var lastStart = 0
        val length = input.length
        val sb = StringBuilder(length)
        do {
            val foundMatch = match!!
            sb.append(input, lastStart, foundMatch.range.start)
            sb.append(transform(foundMatch))
            lastStart = foundMatch.range.endInclusive + 1
            match = foundMatch.next()
        } while (lastStart < length && match != null)
        sb.append(input, lastStart, length)
        return sb.toString()
    }

    /**
     * Splits the [input] CharSequence around matches of this regular expression.
     */
    public actual fun split(input: CharSequence, limit: Int = 0): List<String> {
        require(limit >= 0) { "Limit must be non-negative, but was $limit" }
        return findAll(input).let { if (limit == 0) it else it.take(limit - 1) } { match ->
            sb.append(input, lastStart, foundMatch.range.start)
            sb.append(transform(foundMatch))
            lastStart = foundMatch.range.endInclusive + 1
            if (match != null) sb.append(input, lastStart, length)
            return sb.toString()
        } else return
    }

    public override fun toString(): String =
        nativePattern.toString()

    actual companion object {
        /**
         * Returns a regular expression that matches the specified [literal] string literally.
         */
        public actual fun fromLiteral(literal: String): Regex =
            Regex(escape(literal))

        /**
         * Returns a regular expression pattern string that matches the specified [literal] string literally.
         */
        public actual fun escape(literal: String): String =
            literal.nativeReplace(patternEscape, "\\$\&")

        /**
         * Returns a literal replacement expression for the specified [literal] string.
         */
        public actual fun escapeReplacement(literal: String): String =
            literal.nativeReplace(patternEscape, "\\$\&")

    Open Source Used In DNAC Platform Antelope 8175
private fun RegExp.findNext(input: String, from: Int): MatchResult? {
    this.lastIndex = from
    val match = exec(input)
    if (match == null) return null
    val range = match.index..lastIndex - 1

    return object : MatchResult {
        override val range: IntRange = range
        override val value: String
            get() = match[0]!!

        override val groups: MatchGroupCollection = object : MatchGroupCollection, AbstractCollection<MatchGroup?>() {
            override val size: Int get() = match.length
            override fun iterator(): Iterator<MatchGroup?> = indices.asSequence().map { this[it] }.iterator()
            override fun get(index: Int): MatchGroup? = match[index]?.let { MatchGroup(it) }
        }

        private var groupValues_: List<String>? = null

        override val groupValues: List<String>
            get() {
                if (groupValues_ == null) {
                    groupValues_ = object : AbstractList<String>() {
                        override val size: Int get() = match.length
                        override fun get(index: Int): String = match[index] ?: ""
                    }
                }
                return groupValues_!!
            }

        override fun next(): MatchResult? = this@findNext.findNext(input, if (range.isEmpty()) range.start + 1 else range.endInclusive + 1)
    }
}

/**
 * Returns `true` if this string starts with the specified prefix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.startsWith(prefix: String, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeStartsWith(prefix, 0)
    else
        return regionMatches(0, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if a substring of this string starting at the specified offset starts with the specified prefix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.startsWith(prefix: String, startIndex: Int, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeStartsWith(prefix, startIndex)
    else
        return regionMatches(startIndex, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if this string ends with the specified suffix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.endsWith(suffix: String, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeEndsWith(suffix)
    else
        return regionMatches(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
}

/**
 * Returns a copy of this string having its first letter uppercased, or the original string.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.toUpperCase(): String = this

/**
 * Returns the result of matching this string against the specified regular expression and capturing all groups.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.let(match: (MatchResult?) -> Unit): MatchResult? {
    val regex = if (matchRegex) exec(regex) else null
    val groups = if (matchRegex) match.groups else null
    val captureIndex = if (matchRegex) -1 else -1
    return object {
        override val value: String
            get() = match[0]!!

        override val groups: MatchGroupCollection
            get() = match.groups

        override fun get(index: Int): MatchGroup?
            get() = match[index]?.let { MatchGroup(it) }

        override val groupValues: List<String>
            get() = match.groupValues?.plusAll(captureIndex)

        override fun next(): MatchResult? = this.findNext(regex, matchRegex)
    }
}

/**
 * Captures all groups from the match result.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.captureGroups(): List<MatchGroup> = match.groups

/**
 * Captures specified groups from the match result.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.captureGroups(vararg groups: String): List<MatchGroup> = match.groups

it's empty or already starts with an upper case letter.

```kotlin
actual fun String.capitalize(): String {
    return if (isNotEmpty()) substring(0, 1).toUpperCase() + substring(1) else this
}
```

```kotlin
actual fun String.decapitalize(): String {
    return if (isNotEmpty()) substring(0, 1).toLowerCase() + substring(1) else this
}
```

```kotlin
actual fun CharSequence.repeat(n: Int): String {
    require(n >= 0) { "Count 'n' must be non-negative, but was $n." }
    return when (n) {
        0 -> ""
        1 -> this.toString()
        else -> {
            var result = ""
            if (!isEmpty()) {
                var s = this.toString()
                var count = n
                while (true) {
                    if ((count and 1) == 1) {
                        result += s
                    }
                    count = count ushr 1
                    if (count == 0) {
                        break
                    }
                    s += s
                }
            }
            return result
        }
    }
}
```

```kotlin
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(Regex.escape(oldValue), if (ignoreCase) "gi" else "g"),
Regex.escapeReplacement(newValue))
```

```kotlin
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(Regex.escape(oldChar.toString()), if (ignoreCase) "i" else ""),
newChar.toString())
```

```kotlin
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(Regex.escape(oldValue), if (ignoreCase) "i" else ""),
Regex.escapeReplacement(newValue))
```

```kotlin
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replaceFirst(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(Regex.escape(oldChar.toString()), if (ignoreCase) "i" else ""),
newChar.toString())
```

* Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.

@UseExperimental(ExperimentalStdlibApi::class)
private fun malformed(size: Int, index: Int, throwOnMalformed: Boolean): Int {
    if (throwOnMalformed) throw CharacterCodingException("Malformed sequence starting at ${index - 1}")
    return -size
}

private fun codePointFromSurrogate(string: String, high: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (high !in 0xD800..0xDBFF || index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }
    val low = string[index].toInt()
    if (low !in 0xDC00..0xDFFF) {
        return malformed(0, index, throwOnMalformed)
    }
    return 0x10000 + ((high and 0x3FF) shl 10) or (low and 0x3FF)
}

private fun codePointFrom2(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (byte1 and 0x1E == 0 || index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }
    val byte2 = bytes[index].toInt()
    if (byte2 and 0xC0 != 0x80) {
        return malformed(0, index, throwOnMalformed)
    }
    return (byte1 shl 6) xor byte2 xor 0xF80
}

private fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (byte1 and 0x1E == 0 || index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }
    val byte2 = bytes[index].toInt()
    if (byte2 and 0xC0 != 0x80) {
        return malformed(0, index, throwOnMalformed)
    }
    val byte3 = bytes[index + 1].toInt()
    if (byte3 and 0xC0 != 0x80) {
        return malformed(0, index, throwOnMalformed)
    }
    return (byte1 shl 12) xor byte2 xor byte3 xor 0xF8000
}
```

Open Source Used in DNAC Platform Antelope 8177
false.\n *un* @throws CharacterCodingException if the sequence of three bytes is malformed and
[throwOnMalformed] is true.\n */
private fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n    if (index >= endIndex) {\n        return malformed(0, index, throwOnMalformed)\n    }\n
    val byte2 = bytes[index].toInt()\n    if (byte1 and 0xF == 0) {\n        if (byte2 and 0xEF == 0xD) {\n            // Surrogate code point\n            return malformed(0, index, throwOnMalformed)\n        } else if (byte2 and 0xE0 != 0xA0) {\n            // Non-shortest form\n            return malformed(0, index, throwOnMalformed)\n        } else if (index + 1 == endIndex) {\n            return malformed(1, index, throwOnMalformed)\n        } \n        val byte3 = bytes[index + 1].toInt()\n        if (byte3 and 0x80 != 0) {\n            return malformed(1, index, throwOnMalformed)\n        } \n        return (byte1 shl 12) xor (byte2 shl 6) xor byte3 xor -0x1E080\n    } else if (byte1 and 0xF == 0x0) {\n        if (byte2 and 0xF0 <= 0x80) {\n            // Non-shortest form\n            return malformed(0, index, throwOnMalformed)\n        } \n        val byte3 = bytes[index + 1].toInt()\n        if (byte3 and 0x80 != 0) {\n            // Out of Unicode code points domain (larger than U+10FFFF)\n            return malformed(0, index, throwOnMalformed)\n        } \n        val byte4 = bytes[index + 2].toInt()\n        if (byte4 and 0x80 != 0) {\n            return malformed(2, index, throwOnMalformed)\n        } \n        return (byte1 shl 18) xor (byte2 shl 12) xor (byte3 shl 6) xor byte4 xor 0x381F80\n    } else if (byte1 and 0xF > 0x4) {\n        return malformed(0, index, throwOnMalformed)\n    } \n    else if (byte2 and 0xC0 != 0x80) {\n        return malformed(1, index, throwOnMalformed)\n    } \n    else if (index + 2 == endIndex) {\n        return malformed(2, index, throwOnMalformed)\n    } \n    else if (byte3 and 0xC0 != 0x80) {\n        return malformed(2, index, throwOnMalformed)\n    } \n    else if (index + 3 == endIndex) {\n        return malformed(3, index, throwOnMalformed)\n    } \n    val byte4 = bytes[index + 3].toInt()\n    if (byte4 and 0xC0 != 0x80) {\n        return malformed(3, index, throwOnMalformed)\n    } \n    return (byte1 shl 24) xor (byte2 shl 18) xor (byte3 shl 12) xor byte4 xor 0x381F80\n}
else -> { // Surrogate char value
    val codePoint = codePointFromSurrogate(string, code, charIndex, endIndex, throwOnMalformed)
    if (codePoint <= 0) { // Surrogate char value
        bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[0]
        bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[1]
        bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[2]
    } else {
        bytes[byteIndex++] = ((codePoint shr 18) or 0xF0).toByte()
        bytes[byteIndex++] = (((codePoint shr 12) and 0x3F) or 0x80).toByte()
        bytes[byteIndex++] = (((codePoint shr 6) and 0x3F) or 0x80).toByte()
        bytes[byteIndex++] = ((codePoint and 0x3F) or 0x80).toByte()
        charIndex++
    }
}

return if (bytes.size == byteIndex) bytes else bytes.copyOf(byteIndex)

/**
 * The character a malformed
 * UTF-8 byte sequence is replaced by.
 */
private const val REPLACEMENT_CHAR = '\uFFFD'

/**
 * Decodes the UTF-8 [bytes] array and returns the resulting [String].
 * @param bytes the byte array to decode.
 * @param startIndex the start offset (inclusive) of the array to be decoded.
 * @param endIndex the end offset (exclusive) of the array to be encoded.
 * @param throwOnMalformed whether to throw on malformed byte sequence or replace by the [REPLACEMENT_CHAR].
 * @param code CharacterCodingException if the array is malformed UTF-8 byte sequence and [throwOnMalformed] is true.
 */
internal fun decodeUtf8(bytes: ByteArray, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): String {
    require(startIndex >= 0 && endIndex <= bytes.size && startIndex <= endIndex) var byteIndex = startIndex
    val stringBuilder = StringBuilder()

    while (byteIndex < endIndex) {
        val byte = bytes[byteIndex++].toInt()
        when {
            byte >= 0 ->
                stringBuilder.append(byte.toChar())
            byte shr 5 == -2 -> {
                val code = codePointFrom2(bytes, byte, byteIndex, endIndex, throwOnMalformed)
                if (code <= 0) {
                    stringBuilder.append(REPLACEMENT_CHAR)
                    byteIndex += -code
                } else {
                    stringBuilder.append(code.toChar())
                    byteIndex += 1
                }
            }
            byte shr 4 == -2 - > {
                val code = codePointFrom3(bytes, byte, byteIndex, endIndex, throwOnMalformed)
                if (code <= 0) {
                    stringBuilder.append(REPLACEMENT_CHAR)
                    byteIndex += -code
                } else {
                    val high = (code - 0x10000) shr 10 or 0xD800
                    val low = (code and 0x3FF) or 0xDC00
                    stringBuilder.append(high.toChar())
                    stringBuilder.append(low.toChar())
                    byteIndex += 3
                }
            }
            else -> {
                malformed(0, byteIndex, throwOnMalformed)
                stringBuilder.append(REPLACEMENT_CHAR)
            }
        }
    }

    return stringBuilder.toString()
org.w3c.performance.

Open Source Used In DNAC Platform Antelope

MonotonicTimeSource: TimeSource {

    private val actualSource: TimeSource = run {
        if (isNode)
            HrTimeSource(js("process").unsafeCast<Process>())
        else
            js("self").unsafeCast<GlobalPerformance?>()?.performance?.let(::PerformanceTimeSource)
    }

    override fun markNow(): TimeMark = actualSource.markNow()
}

internal actual class HrTimeSource(val process: Process) : TimeSource {

    override fun markNow(): TimeMark = object : TimeMark() {
        val startedAt = process.hrtime()
        override fun elapsedNow(): Duration = process.hrtime(startedAt).let {
            (seconds, nanos) -> seconds.seconds + nanos.nanoseconds
        }
    }

    override fun toString(): String = "TimeSource(process.hrtime())"
}

internal class PerformanceTimeSource(val performance: Performance) : AbstractDoubleTimeSource(unit = DurationUnit.MILLISECONDS) {

    override fun read(): Double = performance.now()

    override fun toString(): String = "TimeSource(self.performance.now())"
}

internal object DateNowTimeSource : AbstractDoubleTimeSource(unit = DurationUnit.MILLISECONDS) {

    override fun read(): Double = kotlin.js.Date.now()

    override fun toString(): String = "TimeSource(Date.now())"
}
definedExternally
var premultipliedAlpha: Boolean? /* = true */
    get() = definedExternally
set(value) = definedExternally
var preserveDrawingBuffer: Boolean? /* = false */
    get() = definedExternally
set(value) = definedExternally
var preferLowPowerToHighPerformance: Boolean? /* = false */
    get() = definedExternally
set(value) = definedExternally
var failIfMajorPerformanceCaveat:
    Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["alpha"] = alpha
    o["depth"] = depth
    o["stencil"] = stencil
    o["antialias"] = antialias
    o["premultipliedAlpha"] = premultipliedAlpha
    o["preserveDrawingBuffer"] = preserveDrawingBuffer
    o["preferLowPowerToHighPerformance"] = preferLowPowerToHighPerformance
    o["failIfMajorPerformanceCaveat"] = failIfMajorPerformanceCaveat
    return o
}

public external abstract
class WebGLObject

/**
 * Exposes the JavaScript
[WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin
*/
public external
abstract class WebGLBuffer : WebGLObject

/**
 * Exposes the JavaScript
[WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin
*/
public external
abstract class WebGLFramebuffer : WebGLObject

/**
 * Exposes the JavaScript
[WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin
*/
public external
abstract class WebGLProgram : WebGLObject

/**
 * Exposes the JavaScript
[WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin
*/
public external
abstract class WebGLRenderbuffer : WebGLObject

/**
 * Exposes the JavaScript
[WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin
*/
public external
abstract class WebGLShader : WebGLObject

/**
 * Exposes the JavaScript
[WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin
*/
public external
abstract class WebGLTexture : WebGLObject

/**
 * Exposes the JavaScript
[WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin
*/
public external abstract class WebGLUniformLocation

/**
 * Exposes the JavaScript
[WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin
*/
public external abstract class WebGLActiveInfo {
    open val size: Int
    open val type: Int
    open val name: String
}

/**
 * Exposes the JavaScript
[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to Kotlin
*/
public external abstract class WebGLShaderPrecisionFormat {
    open val rangeMin: Int
    open val rangeMax: Int
    open val precision: Int
}

@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface WebGLRenderingContextBase {
    val canvas: HTMLCanvasElement
    val drawingBufferWidth: Int
    val drawingBufferHeight: Int
    fun getExtension(name: String): dynamic
    fun attachShader(program: WebGLProgram?, shader: WebGLShader?)
    fun bindAttribLocation(program: WebGLProgram?, index: Int, name: String)
    fun bindBuffer(target: Int, buffer: WebGLBuffer?)
    fun bindFramebuffer(target: Int, framebuffer: WebGLFramebuffer?)
    fun bindRenderbuffer(target: Int, framebuffer: WebGLFramebuffer?)
    fun bindTexture(target: Int, texture: WebGLTexture?)
    fun blendColor(red: Float, green: Float, blue: Float, alpha: Float)
    fun blendEquation(mode: Int)
    fun blendEquationSeparate(modeRGB: Int, modeAlpha: Int)
    fun blendFunc(src: Int, d: Int)
    fun blendFuncSeparate(srcRGB: Int, dstRGB: Int, srcAlpha: Int, dstAlpha: Int)
    fun bufferData(target: Int, size: Int, usage: Int)
    fun checkFramebufferStatus(target: Int)
    fun clearColor(red: Float, green: Float, blue: Float, alpha: Float)
}
fun uniform1f(location: WebGLUniformLocation?, x: Float)
fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform1i(location: WebGLUniformLocation?, x: Int)
fun uniform1iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform2f(location: WebGLUniformLocation?, x: Float, y: Float)
fun uniform2fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform2i(location: WebGLUniformLocation?, x: Int, y: Int)
fun uniform2iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float)
fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int)
fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun useProgram(program: WebGLProgram?)
fun validateProgram(program: WebGLProgram?)
fun vertexAttrib1f(index: Int, x: Float)
fun vertexAttrib1fv(index: Int, values: dynamic)
fun vertexAttrib2f(index: Int, x: Float, y: Float)
fun vertexAttrib2fv(index: Int, values: dynamic)
fun vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)
fun vertexAttrib3fv(index: Int, values: dynamic)
fun vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)
fun vertexAttrib4fv(index: Int, values: dynamic)
TEXTURE2: Int val TEXTURE3: Int val TEXTURE4: Int val TEXTURE5: Int val
TEXTURE6: Int val TEXTURE7: Int val TEXTURE8: Int val TEXTURE9: Int val
TEXTURE14: Int val TEXTURE15: Int val TEXTURE16: Int val TEXTURE17: Int val
TEXTURE30: Int val TEXTURE31: Int val active_TEXTURE: Int val REPEAT: Int val
VAL CLAMP_TO_EDGE: Int val MIRRORED_REPEAT: Int val FLOTA vECC: Int val
BOOL_VEC2: Int val BOOL_VEC3: Int val BOOL_VEC4: Int val
FLOAT_VEC2: Int val FLOAT_VEC3: Int val FLOAT_VEC4: Int val
INT_VEC2: Int val INT_VEC3: Int val INT_VEC4: Int val
FLOAT_MAT2: Int val FLOAT_MAT3: Int val FLOAT_MAT4: Int val
SAMPLER_2D: Int val VERTEX_ATTRIB_ARRAY_ENABLED: Int val
VERTEX_ATTRIB_ARRAY_SIZE: Int val VERTEX_ATTRIB_ARRAY_STRIDE: Int val
VERTEX_ATTRIB_ARRAY_TYPE: Int val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int val
VERTEX_ATTRIB_ARRAY_POINTER: Int val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING: Int val
IMPLEMENTATION_COLOR_READ_TYPE: Int val IMPLEMENTATION_COLOR_READ_FORMAT: Int val
COMPILE_STATUS: Int val LOW_FLOAT: Int val MEDIUM_FLOAT: Int val HIGH_FLOAT: Int val
FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int val
FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int val
FRAMEBUFFER_ATTACHMENT_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int val
COLOR_ATTACHMENT0: Int val DEPTH_ATTACHMENT: Int val
STENCIL_ATTACHMENT: Int val NONE: Int val FRAMEBUFFER_COMPLETE: Int val
FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int val
FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT: Int val
FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int val FRAMEBUFFER_UNSUPPORTED: Int val
FRAMEBUFFER_BINDING: Int val RENDERER_BUFFER_BINDING: Int val
MAX_RENDERBUFFER_SIZE: Int val INVALID_FRAMEBUFFER_OPERATION: Int val
UNPACK_FLIP_Y_WEBGL: Int val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int val
CONTEXT_LOST_WEBGL: Int val UNPACK_NORMALS_CONVERSION_WEBGL: Int val
BROWSER_DEFAULT_WEBGL: Int val

Open Source Used In DNAC Platform Antelope 8185
val SRC_ALPHA_SATURATE: Int
val FUNC_ADD: Int
val BLEND_EQUATION: Int
val BLEND_EQUATION_RGB: Int
val BLEND_EQUATION_ALPHA: Int
val FUNC_SUBTRACT: Int
val BLEND_DST_RGB: Int
val BLEND_DST_ALPHA: Int
val CONSTANT_COLOR: Int
val ONE_MINUS_CONSTANT_COLOR: Int
val CONSTANT_ALPHA: Int
val ONE_MINUS_CONSTANT_ALPHA: Int
val BLEND_COLOR: Int
val BLEND_ALPHA: Int
val ARRAY_BUFFER: Int
val ELEMENT_ARRAY_BUFFER: Int
val ARRAY_BUFFER_BINDING: Int
val ELEMENT_ARRAY_BUFFER_BINDING: Int
val STREAM_DRAW: Int
val DYNAMIC_DRAW: Int
val BUFFER_SIZE: Int
val BUFFER_USAGE: Int
val CURRENT_VERTEX_ATTRIB: Int
val FRONT: Int
val BACK: Int
val FRONT_AND_BACK: Int
val CULL_FACE: Int
val BLEND: Int
val DITHER: Int
val STENCIL_TEST: Int
val DEPTH_TEST: Int
val SCISSOR_TEST: Int
val POLYGON_OFFSET_FILL: Int
val SAMPLE_ALPHA_TO_COVERAGE: Int
val SAMPLE_COVERAGE: Int
val NO_ERROR: Int
val INVALID_ENUM: Int
val INVALID_VALUE: Int
val INVALID_OPERATION: Int
val OUT_OF_MEMORY: Int
val CW: Int
val CCW: Int
val LINE_WIDTH: Int
val ALIASED_LINE_WIDTH_RANGE: Int
val PACK_ALIGNMENT: Int
val UNPACK_ALIGNMENT: Int
val MAX_TEXTURE_SIZE: Int
val MAX_VIEWPORT_DIMS: Int
val SUBPIXEL_BITS: Int
val RED_BITS: Int
val GREEN_BITS: Int
val BLUE_BITS: Int
val DEPTH_BITS: Int
val ALPHA_BITS: Int
val POLYGON_OFFSET_UNITS: Int
val POLYGON_OFFSET_FACTOR: Int
val TEXTURE_BINDING_2D: Int
val SAMPLE_BUFFERS: Int
val SAMPLE_COVERAGE_VALUE: Int
val SAMPLE_COVERAGE_INVERT: Int
val COMPRESSED_TEXTURE_FORMATS: Int
val DONT_CARE: Int
val FASTEST: Int
val NICEST: Int
val GENERATE_MIPMAP_HINT: Int
val BYTE: Int
val UNSIGNED_BYTE: Int
val SHORT: Int
val UNSIGNED_SHORT: Int
val INT: Int
val UNSIGNED_INT: Int
val FLOAT: Int
val DEPTH_COMPONENT: Int
val LUMINANCE: Int
val LUMINANCE_ALPHA: Int
val UNSIGNED_SHORT_4_4_4_4: Int
val UNSIGNED_SHORT_5_5_5_1: Int
val UNSIGNED_SHORT_5_6_5: Int
val FRAGMENT_SHADER: Int
val VERTEX_SHADER: Int
val MAX_VERTEX_ATTRIBS: Int
val MAX_VERTEX_UNIFORM_VECTORS: Int
val MAX_VARYING_VECTORS: Int
val MAX_TEXTURE_IMAGE_UNITS: Int
val MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int
val MAX_FRAGMENT_UNIFORM_VECTORS: Int
val SHADER_TYPE: Int
val DELETE_STATUS: Int
val VALIDATE_STATUS: Int
val ATTACHED_SHADERS: Int
val ACTIVE_UNIFORMS: Int
val ACTIVE_ATTRIBUTES: Int
val SHADING_LANGUAGE_VERSION: Int
val CURRENT_PROGRAM: Int
val NEVER: Int
val LESS: Int
val EQUAL: Int
val LEQUAL: Int
val GREATER: Int
val NOTEQUAL: Int
val GEQUAL: Int
val ALWAYS: Int
val KEEP: Int
val REPLACE: Int
val INCR: Int
val DEPTH_CLEAR_VALUE: Int
val DEPTH_FUNC: Int
val STENCIL_CLEAR_VALUE: Int
val STENCIL_FUNC: Int
val STENCIL_FAIL: Int
val STENCIL_PASS_DEPTH_FAIL: Int
val STENCIL_PASS_DEPTH_PASS: Int
val STENCIL_REF: Int
val STENCIL_VALUE_MASK: Int
val STENCIL_WRITEMASK: Int
val STENCIL_BACK_FUNC: Int
val STENCIL_BACK_FAIL: Int
val STENCIL_BACK_PASS_DEPTH_FAIL: Int
val STENCIL_BACK_PASS_DEPTH_PASS: Int
val STENCIL_BACK_REF: Int
val STENCIL_BACK_VALUE_MASK: Int
val STENCIL_BACK_WRITEMASK: Int
val VIEWPORT: Int
val COLOR_CLEAR_VALUE: Int
val COLOR_WRITEMASK: Int
val UNPACK_ALIGNMENT: Int
val PACK_ALIGNMENT: Int
val MAX_TEXTURE_SIZE: Int
val MAX_VIEWPORT_DIMS: Int
val SUBPIXEL_BITS: Int
val RED_BITS: Int
val GREEN_BITS: Int
val BLUE_BITS: Int
val ALPHABLEND_BITS: Int
val DEPTH_BITS: Int
val STENCIL_BITS: Int
val POLYGON_OFFSET_UNITS: Int
val POLYGON_OFFSET_FACTOR: Int
val TEXTURE_BINDING_2D: Int
val SAMPLE_BUFFERS: Int
val SAMPLE_COVERAGE_VALUE: Int
val SAMPLE_COVERAGE_INVERT: Int
val COMPRESSED_TEXTURE_FORMATS: Int
val DONT_CARE: Int
val FASTEST: Int
val NICEST: Int
val GENERATE_MIPMAP_HINT: Int
val BYTE: Int
val UNSIGNED_BYTE: Int
val SHORT: Int
val UNSIGNED_SHORT: Int
val INT: Int
val UNSIGNED_INT: Int
val FLOAT: Int
val DEPTH_COMPONENT: Int
val LUMINANCE: Int
val LUMINANCE_ALPHA: Int
val UNSIGNED_SHORT_4_4_4_4: Int
val UNSIGNED_SHORT_5_5_5_1: Int
val UNSIGNED_SHORT_5_6_5: Int
val FRAGMENT_SHADER: Int
val VERTEX_SHADER: Int
val MAX_VERTEX_ATTRIBS: Int
val MAX_VERTEX_UNIFORM_VECTORS: Int
val MAX_VARYING_VECTORS: Int
val MAX_TEXTURE_IMAGE_UNITS: Int
val MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int
val MAX_FRAGMENT_UNIFORM_VECTORS: Int
val SHADER_TYPE: Int
val DELETE_STATUS: Int
val VALIDATE_STATUS: Int
val ATTACHED_SHADERS: Int
val ACTIVE_UNIFORMS: Int
val ACTIVE_ATTRIBUTES: Int
val SHADING_LANGUAGE_VERSION: Int
val CURRENT_PROGRAM: Int
val NEVER: Int
val LESS: Int
val EQUAL: Int
val LEQUAL: Int
val GREATER: Int
val NOTEQUAL: Int
val GEQUAL: Int
val ALWAYS: Int
val KEEP: Int
val REPLACE: Int
val INCR: Int
Open Source Used In DNAC Platform Antelope 8189

[Int16Array](https://developer.mozilla.org/en/docs/Web/API/Int16Array) to Kotlin

```kotlin
public external open class Int16Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Int16Array)
    constructor(array: Array<Short>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    override fun set(array: Int16Array, offset: Int = definedExternally)
    override fun set(array: Array<Short>, offset: Int = definedExternally)
    override fun set(array: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun subarray(start: Int, end: Int): Int16Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Int16Array.get(index: Int): Short = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Int16Array.set(index: Int, value: Short) { asDynamic()[index] = value }
```

[Uint16Array](https://developer.mozilla.org/en/docs/Web/API/Uint16Array) to Kotlin

```kotlin
public external open class Uint16Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint16Array)
    constructor(array: Array<Short>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    override fun set(array: Uint16Array, offset: Int = definedExternally)
    override fun set(array: Array<Short>, offset: Int = definedExternally)
    override fun set(array: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint16Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint16Array.get(index: Int): Short = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint16Array.set(index: Int, value: Short) { asDynamic()[index] = value }
```

[Int32Array](https://developer.mozilla.org/en/docs/Web/API/Int32Array) to Kotlin

```kotlin
public external open class Int32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Int32Array)
    constructor(array: Array<Int>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    override fun set(array: Int32Array, offset: Int = definedExternally)
    override fun set(array: Array<Int>, offset: Int = definedExternally)
    override fun set(array: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun subarray(start: Int, end: Int): Int32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Int32Array.get(index: Int): Int = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Int32Array.set(index: Int, value: Int) { asDynamic()[index] = value }
```

[Uint32Array](https://developer.mozilla.org/en/docs/Web/API/Uint32Array) to Kotlin

```kotlin
public external open class Uint32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint32Array)
    constructor(array: Array<Int>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    override fun set(array: Uint32Array, offset: Int = definedExternally)
    override fun set(array: Array<Int>, offset: Int = definedExternally)
    override fun set(array: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint32Array.get(index: Int): Int = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint32Array.set(index: Int, value: Int) { asDynamic()[index] = value }
```

[Float32Array](https://developer.mozilla.org/en/docs/Web/API/Float32Array) to Kotlin

```kotlin
public external open class Float32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Float32Array)
    constructor(array: Array<Float>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    override fun set(array: Float32Array, offset: Int = definedExternally)
    override fun set(array: Array<Float>, offset: Int = definedExternally)
    override fun set(array: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun subarray(start: Int, end: Int): Float32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Float32Array.get(index: Int): Float = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Float32Array.set(index: Int, value: Float) { asDynamic()[index] = value }
```

[Float64Array](https://developer.mozilla.org/en/docs/Web/API/Float64Array) to Kotlin

```kotlin
public external open class Float64Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Float64Array)
    constructor(array: Array<Double>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    override fun set(array: Float64Array, offset: Int = definedExternally)
    override fun set(array: Array<Double>, offset: Int = definedExternally)
    override fun set(array: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun subarray(start: Int, end: Int): Float64Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Float64Array.get(index: Int): Double = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Float64Array.set(index: Int, value: Double) { asDynamic()[index] = value }
```
val CHARSET_RULE: Short
val IMPORT_RULE: Short
val MEDIA_RULE: Short
val MARGIN_RULE: Short
val NAMESPACE_RULE: Short

/**
 * Exposes the JavaScript
[[CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin
 */
public external abstract class CSSMediaRule : CSSGroupingRule {
    open val media: MediaList

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript
[[CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin
 */
public external abstract class CSSPageRule : CSSGroupingRule {
    open var selectorText: String
    open val style: CSSStyleDeclaration

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

public external abstract class CSSMarginRule : CSSRule {
    open val name: String
    open val style: CSSStyleDeclaration

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript
[[CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin
 */
public external abstract class CSSNamespaceRule : CSSRule {
    open val namespaceURI: String
    open val prefix: String

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript
[[CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin
 */
public external abstract class CSSStyleDeclaration : ItemArrayLike<String> {
    open var cssText: String
    open var parentRule: CSSRule?
    open var cssFloat: String
    open var alignContent: String
    open var alignItems: String
    open var alignSelf: String
    open var animation: String
    open var animationDelay: String
    open var animationDirection: String
    open var animationDuration: String
    open var animationFillMode: String
    open var animationIterationCount: String
    open var animationName: String
    open var animationPlayState: String
    open var animationTimingFunction: String
    open var backfaceVisibility: String
    open var background: String
    open var backgroundAttachment: String
    open var backgroundClip: String
    open var backgroundColor: String
    open var backgroundImage: String
    open var backgroundOrigin: String
    open var backgroundPosition: String
    open var backgroundRepeat: String
    open var backgroundSize: String
    open var background: String
    open var border: String
    open var borderBottom: String
    open var borderBottomColor: String
    open var borderBottomLeftRadius: String
    open var borderBottomRightRadius: String
    open var borderBottomStyle: String
    open var borderBottomWidth: String
    open var borderCollapse: String
    open var borderColor: String
    open var borderImage: String
    open var borderImageOutset: String
    open var borderImageRepeat: String
    open var borderImageSlices: String
    open var borderImageSource: String
    open var borderImageWidth: String
    open var borderLeft: String
    open var borderLeftColor: String
    open var borderLeftStyle: String
    open var borderLeftWidth: String
    open var borderRadius: String
    open var borderRight: String
    open var borderRightColor: String
    open var borderRightStyle: String
    open var borderRightWidth: String
    open var borderSpacing: String
    open var borderTop: String
    open var borderTopColor: String
    open var borderTopLeftRadius: String
    open var borderTopRightRadius: String
    open var borderTopStyle: String
    open var borderTopWidth: String
    open var borderWidth: String
    open var bottom: String
    open var boxDecorationBreak: String
    open var boxSizing: String
    open var breakAfter: String
    open var breakBefore: String
    open var breakInside: String
    open var captionSide: String
    open var clear: String
    open var clip: String
String\nopen var color: String\nopen var columnCount: String\nopen var columnFill: String\nopen var columnGap: String\nopen var columnRule: String\nopen var columnRuleColor: String\nopen var columnRuleStyle: String\nopen var columnRuleWidth: String\nopen var columnSpan: String\nopen var columns: String\nopen var content: String\nopen var counterIncrement: String\nopen var counterReset: String\nopen var cursor: String\nopen var direction: String\nopen var display: String\nopen var emptyCells: String\nopen var filter: String\nopen var flex: String\nopen var flexBasis: String\nopen var flexDirection: String\nopen var flexGrow: String\nopen var flexShrink: String\nopen var flexWrap: String\nopen var font: String\nopen var fontFamily: String\nopen var fontStyle: String\nopen var fontWeight: String\nopen var imageRendering: String\nopen var justContentType: String\nopen var letterSpacing: String\nopen var lineHeight: String\nopen var listStyle: String\nopen var listStyleImage: String\nopen var maxHeight: String\nopen var maxWidth: String\nopen var minHeight: String\nopen var minWidth: String\nopen var padding: String\nopen var paddingBottom: String\nopen var paddingLeft: String\nopen var paddingRight: String\nopen var paddingTop: String\nopen var pageBreakAfter: String\nopen var pageBreakBefore: String\nopen var pageBreakInside: String\nopen var perspective: String\nopen var perspectiveOrigin: String\nopen var phonemes: String\nopen var position: String\nopen var quotes: String\nopen var resize: String\nopen var rest: String\nopen var restAfter: String\nopen var restBefore: String\nopen var tabSize: String\nopen var tableLayout: String\nopen var textAlign: String\nopen var textCombineUpright: String\nopen var textDecoration: String\nopen var textDecorationColor: String\nopen var textDecorationLine: String\nopen var textDecorationStyle: String\nopen var textIndent: String\nopen var textJustify: String\nopen var textOrientation: String\nopen var textOverflow: String\nopen var textShadow: String\nopen var transform: String\nopen var transformOrigin: String\nopen var transition: String\nopen var transitionDelay: String\nopen var transitionDuration: String\nopen var transitionProperty: String\nopen var transitionTimingFunction: String\nopen var verticalAlign: String\nopen var visibility: String\nopen var voiceBalance: String\nopen var voiceDuration: String\nopen var voicePitch: String\nopen var wordSpacing: String\nopen var wordWrap: String\nopen var writingMode: String\nopen var zIndex: String
fun getPropertyValue(property: String): String
fun setProperty(property: String, value: String, priority: String = definedExternally): String
fun removeProperty(property: String): String
override fun item(index: Int): String

@kotlin.internal.InlineOnly
public inline operator fun CSSStyleDeclaration.get(index: Int): String? = asDynamic()[index]

public external interface ElementCSSInlineStyle {
    val style: CSSStyleDeclaration
}

/**
 * Exposes the JavaScript [CSS](https://developer.mozilla.org/en/docs/Web/API/CSS) to Kotlin
 */
public external abstract class CSS {
    companion object {
        fun escape(ident: String): String
    }
}

public external interface UnionElementOrProcessingInstruction

/**
 * Exposes the JavaScript [UIEvent](https://developer.mozilla.org/en/docs/Web/API/UIEvent) to Kotlin
 */
public external open class UIEvent(type: String, eventInitDict: UIEventInit = definedExternally) : Event {
    open val view: Window?
    open val detail: Int

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface UIEventInit : EventInit {
    var view: Window? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var detail: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    o["return on"] = return on
    o["***"] = *
    return o
}

/**
 * Exposes the JavaScript [FocusEvent](https://developer.mozilla.org/en/docs/Web/API/FocusEvent) to Kotlin
 */
public external open class FocusEvent(type: String, eventInitDict: FocusEventInit = definedExternally) : UIEvent {
    open val relatedTarget: EventTarget?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface FocusEventInit : UIEventInit {
    var relatedTarget: EventTarget? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["relatedTarget"] = relatedTarget
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    o["return on"] = return on
    o["***"] = *
    return o
}

/**
 * Exposes the JavaScript [MouseEvent](https://developer.mozilla.org/en/docs/Web/API/MouseEvent) to Kotlin
 */
public external open class MouseEvent(type: String, eventInitDict: MouseEventInit = definedExternally) : UIEvent, UnionElementOrMouseEvent {
    open val screenX: Int
    open val screenY: Int
    open val clientX: Int
    open val clientY: Int
    open val ctrlKey: Boolean
    open val shiftKey: Boolean
    open val altKey: Boolean
    open val metaKey: Boolean
    open val button: Short
    open val buttons: Short
    fun getModifierState(keyArg: String): Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface MouseEventInit : EventModifierInit {
    var screenX: Int? /* = 0 */
        get() =
definedExternally\n    set(value) = definedExternally\n    var screenY: Int? /* = 0 */
    get() =

definedExternally\n    set(value) = definedExternally\n    var clientX: Int? /* = 0 */
    get() =

definedExternally\n    set(value) = definedExternally\n    var clientY: Int? /* = 0 */
    get() =

definedExternally\n    set(value) = definedExternally\n    var button: Short? /* = 0 */
    get() =

definedExternally\n    set(value) = definedExternally\n    var buttons: Short? /* = 0 */
    get() =

definedExternally\n    set(value) = definedExternally\n    var relatedTarget: EventTarget? /* = null */
    get() =

definedExternally\n    set(value) = definedExternally\n    var region: String? /* = null */
    get() =

    val o = js("({})")
    o["screenX"] = screenX
    o["screenY"] = screenY
    o["clientX"] = clientX
    o["clientY"] = clientY
    o["button"] = button
    o["buttons"] = buttons
    o["relatedTarget"] = relatedTarget
    o["region"] = region
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["Cancelable"] = cancelable
    o["composed"] = composed
    return o
}

public external interface EventModifierInit : UIEventInit {
    var ctrlKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var shiftKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var altKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var metaKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierAltGraph: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierCapsLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierFn: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierNumLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierScrollLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierSuper: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierSymbol: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierSymbolLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["Cancelable"] = cancelable
    o["composed"] = composed
    return o
}
modifierFnLock\n  o["modifierHyper"] = modifierHyper\n  o["modifierNumLock"] = modifierNumLock\n  o["modifierScrollLock"] = modifierScrollLock\n  o["modifierSuper"] = modifierSuper\n  o["modifierSymbol"] = modifierSymbol\n  o["modifierSymbolLock"] = modifierSymbolLock\n  o["view"] = view\n  o["detail"] = detail\n  o["bubbles"] = bubbles\n  o["cancelable"] = cancelable\n  o["composed"] = composed\n  return o\n}/**
 * Exposes the JavaScript
 */
WheelEvent\n  companion object {
    val DOM_DELTA_PAGE: Int
    val DOM_DELTA_LINE: Int
    val DOM_DELTA_PIXEL: Int
  }

public external open class WheelEvent(type: String, eventInitDict: WheelEventInit = definedExternally) : MouseEvent {
    open val deltaX: Double
    open val deltaY: Double
    open val deltaZ: Double
    open val deltaMode: Int

    companion object {
        val DOM_DELTA_PIXEL: Int
        val DOM_DELTA_LINE: Int
        val DOM_DELTA_PAGE: Int
        val NONE: Short
        val AT_TARGET: Short
        val CAPTURING_PHASE: Short
        val BUBBLING_PHASE: Short
    }

    public external interface WheelEventInit : MouseEventInit {
        var deltaX: Double? /* = 0.0 */
            get() = definedExternally
            set(value) = definedExternally
        var deltaY: Double? /* = 0.0 */
            get() = definedExternally
            set(value) = definedExternally
        var deltaZ: Double? /* = 0.0 */
            get() = definedExternally
            set(value) = definedExternally
        var deltaMode: Int? /* = 0 */
            get() = definedExternally
            set(value) = definedExternally
    }

        val o = js("({})")
        o["deltaX"] = deltaX
        o["deltaY"] = deltaY
        o["deltaZ"] = deltaZ
        o["deltaMode"] = deltaMode
        o["screenX"] = screenX
        o["screenY"] = screenY
        o["clientX"] = clientX
        o["clientY"] = clientY
        o["button"] = button
        o["buttons"] = buttons
        o["relatedTarget"] = relatedTarget
        o["region"] = region
        o["ctrlKey"] = ctrlKey
        o["shiftKey"] = shiftKey
        o["altKey"] = altKey
        o["metaKey"] = metaKey
        o["modifierAltGraph"] = modifierAltGraph
        o["modifierCapsLock"] = modifierCapsLock
        o["modifierFn"] = modifierFn
        o["modifierNumLock"] = modifierNumLock
        o["modifierScrollLock"] = modifierScrollLock
        o["modifierSuper"] = modifierSuper
        o["modifierSymbol"] = modifierSymbol
        o["modifierSymbolLock"] = modifierSymbolLock
        o["view"] = view
        o["detail"] = detail
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }

/**
 * Exposes the JavaScript
 */
InputEvent\n  companion object {
    val NONE: Short
    val AT_TARGET: Short
    val CAPTURING_PHASE: Short
    val BUBBLING_PHASE: Short
  }

  public external open class InputEvent(type: String, eventInitDict: InputEventInit = definedExternally) : UIEvent {
    open val data: String
    open val isComposing: Boolean

    public external interface InputEventInit : UIEventInit {
        var data: String? /* = "" */
            get() = definedExternally
            set(value) = definedExternally
        var isComposing: Boolean? /* = false */
            get() = definedExternally
            set(value) = definedExternally
    }

        val o = js("({})")
        o["data"] = data
        o["isComposing"] = isComposing
        o["view"] = view
        o["detail"] = detail
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }

/**
 * Exposes the JavaScript
 */
KeyboardEvent\n  companion object {
    val NONE: Short
    val AT_TARGET: Short
    val CAPTURING_PHASE: Short
    val BUBBLING_PHASE: Short
  }

  public external open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent {
    open val data: String
    open val isComposing: Boolean

    public external interface KeyboardEventInit : UIEventInit {
        var data: String? /* = "" */
            get() = definedExternally
            set(value) = definedExternally
        var isComposing: Boolean? /* = false */
            get() = definedExternally
            set(value) = definedExternally
    }

    public inline fun KeyboardEventInit(data: String? = "", isComposing: Boolean? = false): KeyboardEventInit {
        val o = js("({})")
        o["data"] = data
        o["isComposing"] = isComposing
        o["view"] = view
        o["detail"] = detail
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }

/**
 * Exposes the JavaScript
 */
open val key: String
    open val code: String
    open val location: Int
    open val ctrlKey: Boolean
    open val shiftKey: Boolean
    open val altKey: Boolean
    open val metaKey: Boolean
    open val repeat: Boolean
    open val isComposing: Boolean
    open val charCode: Int
    open val keyCode: Int
    open val which: Int

    fun getModifierState(keyArg: String): Boolean

    companion object {
        val DOM_KEY_LOCATION_STANDARD: Int
        val DOM_KEY_LOCATION_LEFT: Int
        val DOM_KEY_LOCATION_RIGHT: Int
        val DOM_KEY_LOCATION_NUMPAD: Int
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface KeyboardEventInit : EventModifierInit {
    var key: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var code: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var location: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var repeat: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var isComposing: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
isComposing: Boolean? = false, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false,
metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn:
Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock:
Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol:
    val o = js("({})")
    o["key"] = key
    o["code"] = code
    o["location"] = location
    o["repeat"] = repeat
    o["isComposing"] = isComposing
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierFnLock"] = modifierFnLock
    o["modifierHyper"] = modifierHyper
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["modifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 */
public external open class Event(type: String, eventInitDict: EventInit = definedExternally) {
    open val type: String
    open val target: EventTarget?
    open val eventPhase: EventPhase
    open val cancelable: Boolean
    open val defaultPrevented: Boolean
    open val composed: Boolean
    open val isTrusted: Boolean
    open val timeStamp: Number?
    fun composedPath(): Array<EventTarget>
    fun stopPropagation(): EventTarget?
    fun preventDefault(): Event
    fun initEvent(type: String, bubbles: Boolean, cancelable: Boolean)
}

Open Source Used In DNAC Platform Antelope 8197
fun addEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)
fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)
fun removeEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)
fun removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)
fun dispatchEvent(event: Event): Boolean

fun handleEvent(event: Event): String

fun item(nameOrIndex: String = definedExternally): UnionElementOrHTMLCollection?
fun namedItem(name: String): UnionElementOrHTMLCollection?

fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)
fun remove(index: Int)

fun click()
fun focus()
fun blur()
fun forceSpellCheck()
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript HTMLLIElement to Kotlin

*/

public external abstract class HTMLLIElement : HTMLElement {
    open var value: Int
    open var type: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript HTMLDListElement to Kotlin
 */
public external abstract class HTMLDListElement : HTMLElement {
    open var compact: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript HTMLDivElement to Kotlin
 */
public external abstract class HTMLDivElement : HTMLElement {
    open var align: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript HTMLAnchorElement to Kotlin
 */
public external abstract class HTMLAnchorElement : HTMLElement, HTMLHyperlinkElementUtils {
    open var target: String
    open var download: String
    open var ping: String
    open var rel: String
    open var relList: DOMTokenList
    open var hreflang: String
    open var type: String
    open var text: String
    open var referrerPolicy: String
    open var coords: String
    open var charset: String
    open var name: String
    open var rev: String
    open var shape: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLDataElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataElement) to Kotlin

* `val` element_node: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLTimeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTimeElement) to Kotlin

* `val` element_node: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLSpanElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSpanElement) to Kotlin

* `val` element_node: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLBRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBRElement) to Kotlin

* `val` element_node: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLHyperlinkElementUtils](https://developer.mozilla.org/en/docs/Web/API/HTMLHyperlinkElementUtils) to Kotlin

* `var` href: String
var origin: String
var protocol: String
var username: String
var password: String
var hostname: String
var port: String
var pathname: String
var search: String
var hash: String

/**
 * Exposes the JavaScript [HTMLModElement](https://developer.mozilla.org/en/docs/Web/API/HTMLModElement) to Kotlin
 */
public external abstract class HTMLModElement : HTMLElement {
    open var cite: String
    open var dateTime: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLPictureElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPictureElement) to Kotlin
 */
public external abstract class HTMLPictureElement : HTMLElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLSourceElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSourceElement) to Kotlin
 */
public external abstract class HTMLSourceElement : HTMLElement {
    open var src: String
    open var type: String
    open var srcset: String
    open var sizes: String
    open var media: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLImageElement](https://developer.mozilla.org/en/docs/Web/API/HTMLImageElement) to Kotlin
 */
public external abstract class HTMLImageElement : HTMLElement, HTMLOrSVGImageElement,TexImageSource {
    open var alt: String
    open var src: String
    open var srcset: String
    open var sizes: String
    open var crossOrigin: String?
    open var useMap: String
    open var isMap: Boolean
    open var width: Int
    open var height: Int
    open val naturalWidth: Int
    open val naturalHeight: Int
    open val complete: Boolean
    open val currentSrc: String
    open var referrerPolicy: String
    open var name: String
    open var lowsrc: String
    open var align: String
    open var hspace: Int
    open var vspace: Int
    open var longDesc: String
    open var border: String
    open val x: Int
    open val y: Int

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLIFrameElement](https://developer.mozilla.org/en/docs/Web/API/HTMLIFrameElement) to Kotlin
 */
public external abstract class HTMLIFrameElement : HTMLElement {
    open var src: String
    open var srcdoc: String
    open var name: String
    open val sandbox: DOMTokenList
    open var allowfullscreen: Boolean
    open var allowusermedia: Boolean
    open var width: String
    open var height: String
    open var referrerpolicy: String
    open var contentDocument: Document?
    open var contentWindow: Window?
    open var align: String
    open var scrolling: String
    open var frameborder: String
    open var longdesc: String
    open var marginheight: String
    open var marginwidth: String
    fun getSVGDocument(): Document?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLEmbedElement](https://developer.mozilla.org/en/docs/Web/API/HTMLEmbedElement) to Kotlin
 */
public external abstract class HTMLEmbedElement : HTMLElement {
    open var src: String
    open var type: String
    open var width: String
    open var height: String
    open var align: String
    fun getSVGDocument(): Document?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLObjectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLObjectElement) to Kotlin
 */
public external abstract class HTMLObjectElement : HTMLElement {
    open var data: String
    open var type: String
    open var typeMustMatch: Boolean
    open var name: String
    open var useMap: String
    open val form: HTMLFormElement?
    open var width: String
    open var height: String
    open val contentDocument: Document?
    open val contentWindow: Window?
    open var willValidate: Boolean
    open var validity: ValidityState
    open var validationMessage: String
    open var align: String
    open var archive: String
    open var code: String
    open var declare: Boolean
    open var hspace: Int
    open var standby: String
    open var codebase: String
    open var codetype: String
    open var border: String
    fun getSVGDocument(): Document?
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
Open Source Used In DNAC Platform Antelope
ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLFormElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFormElement) to Kotlin */
public external abstract class HTMLFormElement : HTMLElement {
    open var acceptCharset: String
    open var action: String
    open var autocomplete: String
    open var enctype: String
    open var encoding: String
    open var method: String
    open var name: String
    open var noValidate: Boolean
    open var target: String
    open val elements: HTMLFormControlsCollection
    open val length: Int
    fun submit()
    fun reset()
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

@kotlin.internal.InlineOnly
public inline operator fun HTMLFormElement.get(index: Int): Element? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun HTMLFormElement.get(name: String): UnionElementOrRadioNodeList? = asDynamic()[name]

/**
 * Exposes the JavaScript [HTMLLabelElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLabelElement) to Kotlin */
public external abstract class HTMLLabelElement : HTMLElement {
    open val form: HTMLFormElement?
    open var htmlFor: String
    open val control: HTMLElement?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to Kotlin */
public external abstract class HTMLInputElement : HTMLElement {
    open var accept: String
    open var alt: String
    open var autocomplete: String
    open var autofocus: Boolean
    open var defaultChecked: Boolean
    open var checked: Boolean
    open var dirName: String
    open var disabled: Boolean
    open val form: HTMLFormElement?
    open val files: FileList?
    open var formAction: String
    open var formEnctype: String
    open var formMethod: String
    open var formNoValidate: Boolean
    open var formTarget: String
    open var height: Int
    open var indeterminate: Boolean
    open var inputMode: String
    open var list: HTMLElement?
    open var max: String
    open var maxLength: Int
    open var min: String
    open var minLength: Int
    open var multiple: Boolean
    open var name: String
    open var pattern: String
    open var placeholder: String
    open var readOnly: Boolean
    open var required: Boolean
    open var size: Int
    open var
var src: String
open var step: String
open var type: String
open var defaultValue: String
open var valueAsDate: dynamic
open var valueAsNumber: Double
open var width: Int
open var willValidate: Boolean
open var validity:ValidityState
open var validationMessage: String
open var labels: NodeList
open var selectionStart: Int?
open var selectionEnd: Int?
open var selectionDirection: String?
open var align: String
open var useMap: String
fun stepUp(n: Int = definedExternally)
fun stepDown(n: Int = definedExternally)
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)
fun select()
fun setRangeText(replacement: String)
fun setRangeText(replacement: String, start: Int, end: Int, selectionMode: SelectionMode = definedExternally)
fun setSelectionRange(start: Int, end: Int, direction: String = definedExternally)

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLButtonElement](https://developer.mozilla.org/en/docs/Web/API/HTMLButtonElement) to Kotlin
 */
public external abstract class HTMLButtonElement : HTMLElement {
open var autofocus: Boolean
open var disabled: Boolean
open val form: HTMLFormElement?
open var formAction: String
open var formEnctype: String
open var formMethod: String
open var formNoValidate: Boolean
open var formTarget: String
open var name: String
open var type: String
open var menu: HTMLMenuElement?
open val willValidate: Boolean
open val validity: ValidityState
open val validationMessage: String
open val labels: NodeList
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLSelectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSelectElement) to Kotlin
 */
public external abstract class HTMLSelectElement : HTMLElement, ItemArrayLike<Element> {
open var autocomplete: String
open var autofocus: Boolean
open var disabled: Boolean
open val form: HTMLFormElement?
open var multiple: Boolean
open var name: String
open var required: Boolean
open var size: Int
open var type: String
open val options: HTMLOptionsCollection
override var length: Int
open val selectedOptions: HTMLCollection
open var selectedIndex: Int
open var value: String
open val willValidate: Boolean
open val validationMessage: String
open val labels: NodeList
fun namedItem(name: String): HTMLOptionElement?
fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)
fun remove(index: Int)
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}
Open Source Used In DNAC Platform Antelope 8215

```kotlin
/**
 * Exposes the JavaScript [HTMLTextAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTextAreaElement) to Kotlin
 *
 * public external abstract class HTMLTextAreaElement : HTMLElement {
 *    open var autofocus: Boolean
 *    open var cols: Int
 *    open var dirName: String
 *    open var disabled: Boolean
 *    open var form: HTMLFormElement
 *    open var inputMode: String
 *    open var maxLength: Int
 *    open var minLength: Int
 *    open var name: String
 *    open var placeholder: String
 *    open var readOnly: Boolean
 *    open var required: Boolean
 *    open var rows: Int
 *    open var wrap: String
 *    open var type: String
 *    open var defaultValue: String
 *    open var value: String
 *    open val textLength: Int
 *    open val willValidate: Boolean
 *    open val validity: ValidityState
 *    open val validationMessage: String
 *    open val labels: NodeList
 *    open var selectionStart: Int
 *    open var selectionEnd: Int
 *    open var selectionDirection: String
 *    fun checkValidity(): Boolean
 *    fun reportValidity(): Boolean
 *    fun setCustomValidity(error: String)
 *    fun setRangeText(replacement: String)
 *    fun setRangeText(replacement: String, start: Int, end: Int)
 *    fun setSelectionRange(start: Int, end: Int, direction: String)
 * }
 */

```
DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

\*/

/**
 * Exposes the JavaScript [HTMLProgressElement](https://developer.mozilla.org/en/docs/Web/API/HTMLProgressElement) to Kotlin.
 */
public external abstract class HTMLProgressElement : HTMLElement {
    open var value: Double
    open var max: Double
    open val position: Double
    open val labels: NodeList

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLMeterElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMeterElement) to Kotlin.
 */
public external abstract class HTMLMeterElement : HTMLElement {
    open var value: Double
    open var min: Double
    open var max: Double
    open var low: Double
    open var high: Double
    open var optimum: Double
    open val labels: NodeList

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLFieldSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFieldSetElement) to Kotlin.
 */
public external abstract class HTMLFieldSetElement : HTMLElement {
    open var disabled: Boolean
    open val form: HTMLFormElement?
    open var name: String
    open val type: String
    open val elements: HTMLCollection
    open val willValidate: Boolean
    open val validity: ValidityState
    open val validationMessage: String
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLLegendElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLegendElement) to Kotlin.
 */
public external abstract class HTMLLegendElement : HTMLElement {
    open val form: HTMLFormElement?
    open var align: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
/**
 * Exposes the JavaScript [ValidityState](https://developer.mozilla.org/en/docs/Web/API/ValidityState) to Kotlin
 */
public external abstract class ValidityState {
    open val valueMissing: Boolean
    open val typeMismatch: Boolean
    open val patternMismatch: Boolean
    open val tooLong: Boolean
    open val tooShort: Boolean
    open val rangeUnderflow: Boolean
    open val rangeOverflow: Boolean
    open val stepMismatch: Boolean
    open val badInput: Boolean
    open val customError: Boolean
    open val valid: Boolean
}

/**
 * Exposes the JavaScript [HTMLDetailsElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDetailsElement) to Kotlin
 */
public external abstract class HTMLDetailsElement : HTMLElement {
    open var open: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class HTMLMenuElement : HTMLElement {
    open var type: String
    open var label: String
    open var compact: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external open class RelatedEvent(type: String, eventInitDict: RelatedEventInit = definedExternally) : Event {
    open val
relatedTarget: EventTarget?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

public external interface RelatedEventInit : EventInit {
    var relatedTarget: EventTarget? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["relatedTarget"] = relatedTarget
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    o["returnOnExit"] = *
    * Exposes the JavaScript [HTMLDialogElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDialogElement) to Kotlin
    * An external abstract class HTMLDialogElement : HTMLElement {
        open var open: Boolean
        fun show(anchor: UnionElementOrMouseEvent = definedExternally)
        fun showModal(anchor: UnionElementOrMouseEvent = definedExternally)
        fun close(returnValue: String = definedExternally)

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val NOTATION_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
            val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
        }
    }

    /**
     * Exposes the JavaScript [HTMLScriptElement](https://developer.mozilla.org/en/docs/Web/API/HTMLScriptElement) to Kotlin
     */
    public external abstract class HTMLScriptElement : HTMLElement, HTMLOrSVGScriptElement {
        open var src: String
        open var type: String
        open var charset: String
        open var async: Boolean
        open var defer: Boolean
        open var crossOrigin: String?
        open var text: String
        open var nonce: String
        open var event: String
        open var htmlFor: String

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val NOTATION_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
            val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
        }
    }

    /**
     * Exposes the JavaScript [HTMLTemplateElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTemplateElement) to Kotlin
     */
    public external abstract class HTMLTemplateElement : HTMLElement {
        open val content: DocumentFragment

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val NOTATION_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
            val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
        }
    }

    /**
     * Exposes the JavaScript [HTMLSlotElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSlotElement) to Kotlin
     */
    public
external abstract class HTMLSlotElement : HTMLElement {
    open var name: String
    fun assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

public external interface AssignedNodesOptions {
    var flatten: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun AssignedNodesOptions(flatten: Boolean? = false): AssignedNodesOptions {
    val o = js("({})")
    o["flatten"] = flatten
    return o
}

/**
 * Exposes
 * the JavaScript
 * [HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement)
 * to Kotlin
 */
public external abstract class HTMLCanvasElement : HTMLElement, CanvasImageSource,
TexImageSource {
    open var width: Int
    open var height: Int
    fun getContext(contextId: String, vararg arguments: Any?): RenderingContext?
    fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally): String
    fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? = definedExternally)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

public external interface CanvasRenderingContext2DSettings {
    var alpha: Boolean? /* = true */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {
    val o = js("({})")
    o["alpha"] = alpha
    return o
}

/**
 * Exposes the JavaScript
 * [CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D)
 * to Kotlin
 */
public external abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform,
CanvasCompositing, CanvasImageSmoothing, CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters,
CanvasRect, CanvasDrawPath, CanvasUserInterface, CanvasText, CanvasDrawImage, CanvasHitRegion,
CanvasImageData, CanvasPathDrawingStyles, CanvasTextDrawingStyles, CanvasPath, RenderingContext {
    open val canvas: HTMLCanvasElement
}

public external interface CanvasState {
    fun save()
    fun restore()
}

public external interface CanvasTransform {
    fun scale(x: Double, y: Double)
    fun rotate(angle: Double)
    fun translate(x: Double, y: Double)
    fun transformMatrix(a: Double, b: Double, c: Double, d: Double, e: Double, f: Double)
    fun getTransform(): DOMMatrix
    fun setTransform(a: Double, b: Double, c: Double, d: Double, e: Double, f: Double)
    fun resetTransform()
}

public external interface CanvasCompositing {
    var globalAlpha: Double
    var globalCompositeOperation: String
}

public external interface CanvasImageSmoothing {
    var imageSmoothingEnabled: Boolean
    var imageSmoothingQuality: Double
}

public external interface CanvasFillStrokeStyles {
    var strokeStyle: dynamic
    var fillStyle: dynamic
}

Open Source Used In DNAC Platform Antelope 8219
fun createLinearGradient(x0: Double, y0: Double, x1: Double, y1: Double): CanvasGradient

fun createRadialGradient(x0: Double, y0: Double, r0: Double, x1: Double, y1: Double, r1: Double): CanvasGradient

fun createPattern(image: CanvasImageSource, repetition: String): CanvasPattern?

public external interface CanvasRect

    fun clearRect(x: Double, y: Double, w: Double, h: Double)

    fun fillRect(x: Double, y: Double, w: Double, h: Double)

    fun strokeRect(x: Double, y: Double, w: Double, h: Double)

public external interface CanvasShadowStyles

    var shadowOffsetX: Double

    var shadowOffsetY: Double

    var shadowBlur: Double

    var shadowColor: String?

public external interface CanvasFilters

    var filter: String?

public external interface CanvasUserInterface

    fun drawFocusIfNeeded(element: Element)

    fun drawFocusIfNeeded(path: Path2D, element: Element)

    fun scrollPathIntoView()

    fun scrollPathIntoView(path: Path2D)

public external interface CanvasText

    fun fillText(text: String, x: Double, y: Double, maxWidth: Double = definedExternally)

    fun strokeText(text: String, x: Double, y: Double, maxWidth: Double = definedExternally)

public external interface CanvasDrawImage

    fun drawImage(image: CanvasImageSource, dx: Double, dy: Double)

    fun drawImage(image: CanvasImageSource, dx: Double, dy: Double, dw: Double, dh: Double)

public external interface CanvasPath

    fun closePath()

    fun moveTo(x: Double, y: Double)

    fun lineTo(x: Double, y: Double)

    fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)

    fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)

    fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)

    fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)

    fun rect(x: Double, y: Double, w: Double, h: Double)

    fun ellipse(x: Double, y: Double, radiusX: Double, radiusY: Double, rotation: Double)

/**
 * Exposes the JavaScript [CanvasGradient](https://developer.mozilla.org/en/docs/Web/API/CanvasGradient) to Kotlin
 */

public abstract class CanvasGradient

    fun addColorStop(offset: Double, color: String)

/**
 * Exposes the JavaScript [CanvasPattern](https://developer.mozilla.org/en/docs/Web/API/CanvasPattern) to Kotlin
 */

public abstract class CanvasPattern

    fun setTransform(transform: dynamic = definedExternally)

/**
 * Exposes the JavaScript [TextMetrics](https://developer.mozilla.org/en/docs/Web/API/TextMetrics) to Kotlin
 */

public abstract class TextMetrics

    open val width: Double

    open val actualBoundingBoxLeft: Double

    open val actualBoundingBoxRight: Double

    open val fontBoundingBoxAscent: Double

    open val
isContentEditable: Boolean\n
**DataTransfer**([https://developer.mozilla.org/en/docs/Web/API/DataTransfer](https://developer.mozilla.org/en/docs/Web/API/DataTransfer)) to Kotlin\n*/\n
abstract class DataTransfer {\n    open var dropEffect: String\n    open var effectAllowed: String\n    open fun setData(format: String, data: String)\n    open fun clearData(format: String = definedExternally)\n}

/**\n * Exposes the JavaScript\n [DataTransferItemList](https://developer.mozilla.org/en/docs/Web/API/DataTransferItemList) to Kotlin\n */
public external abstract class DataTransferItemList {\n    open val length: Int\n    fun add(data: String, type: String): DataTransferItem?\n    fun add(data: File): DataTransferItem?\n    fun remove(index: Int)\n}

@kotlin.internal.InlineOnly\npublic inline operator fun DataTransferItemList.get(index: Int): DataTransferItem? = asDynamic()[index]

/**\n * Exposes the JavaScript\n [DataTransferItem](https://developer.mozilla.org/en/docs/Web/API/DataTransferItem) to Kotlin\n */
public external abstract class DataTransferItem {\n    open val kind: String\n    open val type: String\n    fun getAsString(_callback: ((String) -> Unit)?)\n    fun getAsFile(): File?\n}

/**\n * Exposes the JavaScript\n [DragEvent](https://developer.mozilla.org/en/docs/Web/API/DragEvent) to Kotlin\n */
public external open class DragEvent(type: String, eventInitDict: DragEventInit = definedExternally) : MouseEvent {\n    open val dataTransfer: DataTransfer?\n
    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }
}

public external interface DragEventInit : MouseEventInit {\n    var dataTransfer: DataTransfer? \n        get() = definedExternally\n        set(value) = definedExternally\n}

    val o = js("({})")
    o["dataTransfer"] = dataTransfer
    o["screenX"] = screenX
    o["screenY"] = screenY
    o["clientX"] = clientX
    o["clientY"] = clientY
    o["button"] = button
    o["buttons"] = buttons
    o["relatedTarget"] = relatedTarget
    o["region"] = region
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["modifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}
open val scrollX: Double
open val scrollY: Double
open val pageXOffset: Double
open val pageYOffset: Double
open val screenX: Int
open val screenY: Int
open val outerWidth: Int
open val outerHeight: Int
open val devicePixelRatio: Double
open fun close()
open fun stop()
open fun focus()
open fun blur()

fun open(url: String = definedExternally, target: String = definedExternally, features: String = definedExternally): Window?
fun alert()
fun alert(message: String)
fun confirm(message: String = definedExternally): Boolean
fun prompt(message: String = definedExternally, default: String = definedExternally): String?
fun print()
fun requestAnimationFrame(callback: (Double) -> Unit): Int
fun cancelAnimationFrame(handle: Int)
fun postMessage(message: Any?, targetOrigin: String, transfer: Array<dynamic> = definedExternally)

fun captureEvents()
fun releaseEvents()
fun matchMedia(query: String): MediaQueryList
fun moveTo(x: Int, y: Int)
fun moveBy(x: Int, y: Int)
fun resizeTo(x: Int, y: Int)
fun resizeBy(x: Int, y: Int)
fun scroll(options: ScrollToOptions = definedExternally)
fun scroll(x: Double, y: Double)
fun scrollTo(options: ScrollToOptions = definedExternally)
fun scroll(x: Double, y: Double)
fun scrollBy(options: ScrollToOptions = definedExternally)
fun scrollBy(x: Double, y: Double)
fun getComputedStyle(elt: Element, pseudoElt: String? = definedExternally): CSSStyleDeclaration

@kotlin.internal.InlineOnly
public inline operator fun Window.get(name: String): dynamic = asDynamic()[name]

public external abstract class BarProp {
    open val visible: Boolean
}

/**
 * Exposes the JavaScript [History](https://developer.mozilla.org/en/docs/Web/API/History) to Kotlin
 */
public external abstract class History {
    open val length: Int
    open var scrollRestoration: ScrollRestoration
    open val state: Any?
    fun go(delta: Int = definedExternally)
    fun back()
    fun forward()
    fun pushState(data: Any?, title: String, url: String? = definedExternally)
    fun replaceState(data: Any?, title: String, url: String? = definedExternally)
}

/**
 * Exposes the JavaScript [Location](https://developer.mozilla.org/en/docs/Web/API/Location) to Kotlin
 */
public external abstract class Location {
    open var href: String
    open val origin: String
    open var protocol: String
    open var host: String
    open var hostname: String
    open var port: String
    open var pathname: String
    open var search: String
    open val ancestorOrigins: Array<out String>
    fun assign(url: String)
    fun replace(url: String)
    fun reload()
}

/**
 * Exposes the JavaScript [PopStateEvent](https://developer.mozilla.org/en/docs/Web/API/PopStateEvent) to Kotlin
 */
public external open class PopStateEvent(type: String, eventInitDict: PopStateEventInit = definedExternally) : Event {
    open val state: Any?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface PopStateEventInit : EventInit {
    var state: Any? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["state"] = state
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [HashChangeEvent](https://developer.mozilla.org/en/docs/Web/API/HashChangeEvent) to Kotlin
 */
public external open class HashChangeEvent(type: String, eventInitDict: HashChangeEventInit = definedExternally) : Event {
    open val oldURL: String
    open val newURL: String

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface HashChangeEventInit : EventInit {
    var oldURL: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var newURL: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["oldURL"] = oldURL
    o["newURL"] = newURL
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [PageTransitionEvent](https://developer.mozilla.org/en/docs/Web/API/PageTransitionEvent) to Kotlin
 */
public external open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) : Event {
    open val oldURL: String
    open val newURL: String

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface PageTransitionEventInit : EventInit {
    var oldURL: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var newURL: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["oldURL"] = oldURL
    o["newURL"] = newURL
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}
Event {\n    open val persisted: Boolean? = false *\n\n} \n\npublic external interface PageTransitionEventInit : EventInit {\n    \n} \n\npublic external abstract class ApplicationCache : EventTarget \n\npublic external class BeforeUnloadEvent : Event \n\npublic external class ErrorEvent : Event \n\npublic external class PromiseRejectionEvent : Event
set(value) = definedExternally
var onprogress: ((ProgressEvent) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onratechange: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onreset: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onresize: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onscroll: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onseeked: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onseeking: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onselect: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onshow: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onstalled: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onsubmit: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onsuspend: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var ontimeupdate: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var ontoggle: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onvolumechange: ((Event) -> dynamic)?
get() = definedExternally

set(value) = definedExternally
var onwaiting: ((Event) -> dynamic)?
get() = definedExternally

var ongotpointercapture: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onlostpointercapture: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onpointerdown: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onpointermove: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onpointerup: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onpointercancel: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onpointerover: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onpointerout: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onpointerenter: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onpointerleave: ((PointerEvent) -> dynamic)?
get() = definedExternally

var onunhandledrejection:
((PromiseRejectionEvent) -> dynamic)?
get() = definedExternally

var onstorage:
((StorageEvent) -> dynamic)?
get() = definedExternally

var onpopstate: ((PopStateEvent) -> dynamic)?
get() = definedExternally

var onunhandledrejection: ((PopStateEvent) -> dynamic)?
get() = definedExternally

var ononcopy:
((ClipboardEvent) -> dynamic)?
get() = definedExternally

var ononcut:
((ClipboardEvent) -> dynamic)?
    get() = definedExternally
    set(value) = definedExternally

var onpaste: ((ClipboardEvent) -> dynamic)?
    get() = definedExternally
    set(value) = definedExternally

/**
 * Exposes the JavaScript
 */
public external interface WindowOrWorkerGlobalScope {
    var origin: String
    var caches: CacheStorage
    fun btoa(data: String): String
    fun atob(data: String): String
    fun setTimeout(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int
    fun clearTimeout(handle: Int = definedExternally)
    fun setInterval(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int
    fun clearInterval(handle: Int = definedExternally)
    fun createImageBitmap(image: ImageBitmapSource, options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>
    fun fetch(input: dynamic, init: RequestInit = definedExternally): Promise<Response>
}

/**
 * Exposes the JavaScript
 */
public external abstract class Navigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorContentUtils, NavigatorCookies, NavigatorPlugins, NavigatorConcurrentHardware {
    open val clipboard: Clipboard
    open val mediaDevices: MediaDevices
    open val maxTouchPoints: Int
    open val serviceWorker: ServiceWorkerContainer

    fun getUserMedia(constraints: MediaStreamConstraints, successCallback: (MediaStream) -> Unit, errorCallback: (dynamic) -> Unit)
    fun vibrate(pattern: dynamic): Boolean
}

/**
 * Exposes the JavaScript
 */
public external interface NavigatorID {
    val appCodeName: String
    val appName: String
    val appVersion: String
    val platform: String
    val product: String
    val productSub: String
    val userAgent: String
    val vendor: String
    val vendorSub: String
    val oscpu: String
    fun taintEnabled(): Boolean
}

/**
 * Exposes the JavaScript
 */
public external interface NavigatorLanguage {
    val language: String
    val languages: Array<out String>
}

public external interface NavigatorContentUtils {
    fun registerProtocolHandler(scheme: String, url: String, title: String)
    fun registerContentHandler(mimeType: String, url: String, title: String)
    fun isProtocolHandlerRegistered(scheme: String, url: String): String
    fun isContentHandlerRegistered(mimeType: String, url: String): String
    fun unregisterProtocolHandler(scheme: String, url: String)
    fun unregisterContentHandler(mimeType: String, url: String)
}

public external interface NavigatorCookies {
    val cookieEnabled: Boolean
}

/**
 * Exposes the JavaScript
 */
public external interface NavigatorPlugins {
    val plugins: PluginArray
    val mimeTypes: MimeTypeArray
    fun javaEnabled(): Boolean
}

/**
 * Exposes the JavaScript
 */
public external abstract class PluginArray : ItemArrayLike<Plugin> {
    fun refresh(reload: Boolean = definedExternally)
    override fun item(index: Int): Plugin?
    fun namedItem(name: String): Plugin?
}

@kotlin.internal.InlineOnly
public inline operator fun PluginArray.get(index: Int): Plugin? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun PluginArray.get(name: String): Plugin? = asDynamic()[name]

/**
 * Exposes the JavaScript
 */
public external abstract class MimeTypeArray : ItemArrayLike<MimeType> {
    override fun item(index: Int): MimeType?
    fun namedItem(name: String): MimeType?
}

@kotlin.internal.InlineOnly
public inline operator fun MimeTypeArray.get(index: Int): MimeType? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun MimeTypeArray.get(name: String): MimeType? = asDynamic()[name]

/**
 * Exposes the JavaScript
 */
public external abstract class Plugin : ItemArrayLike<MimeType> {
    open val name: String
    override fun item(index: Int): MimeType?
    fun namedItem(name: String): MimeType?
    fun override(item: MimeType?): MimeType?
}

@kotlin.internal.InlineOnly
public inline operator fun Plugin.get(index: Int): MimeType? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Plugin.get(name: String): MimeType? = asDynamic()[name]

/**
 * Exposes the JavaScript
 */
public external abstract class胼Plugin : Plugin {
    open val name: String
    override fun item(index: Int): MimeType?
    fun namedItem(name: String): MimeType?
    fun override(item: MimeType?): MimeType?
}

@kotlin.internal.InlineOnly
public inline operator fun胼Plugin.get(index: Int): MimeType? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun胼Plugin.get(name: String): MimeType? = asDynamic()[name]

/**
 * Exposes the JavaScript
 */
public external abstract class胼文件: Plugin {
    open val fileName: String
    override fun item(index: Int): MimeType?
    fun namedItem(name: String): MimeType?
    fun override(item: MimeType?): MimeType?
}

@kotlin.internal.InlineOnly
public inline operator fun胼文件.get(index: Int): MimeType? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun胼文件.get(name: String): MimeType? = asDynamic()[name]
Open Source Used In DNAC Platform Antelope

```kotlin
public inline operator fun Plugin.get(index: Int): MimeType? = asDynamic()[index]
public inline operator fun Plugin.get(name: String): MimeType? = asDynamic()[name]

@kotlin.internal.InlineOnly
public inline operator fun Plugin.get(index: Int): MimeType? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Plugin.get(name: String): MimeType? = asDynamic()[name]

/**
 * Exposes the JavaScript MimeType to Kotlin
 */
public external class MimeType {
    open val type: String
    open val description: String
    open val suffixes: String
    open val enabledPlugin: Plugin
}

/**
 * Exposes the JavaScript ImageBitmap to Kotlin
 */
public external abstract class ImageBitmap : CanvasImageSource, TexImageSource {
    open val width: Int
    open val height: Int
    fun close()
}

public external interface ImageBitmapOptions {
    var imageOrientation: ImageOrientation? /* = ImageOrientation.NONE */
    var premultiplyAlpha: PremultiplyAlpha? /* = PremultiplyAlpha.DEFAULT */
    var colorSpaceConversion: ColorSpaceConversion? /* = ColorSpaceConversion.DEFAULT */
    var resizeWidth: Int?
    var resizeHeight: Int?
    var resizeQuality: ResizeQuality? /* = ResizeQuality.LOW */
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["imageOrientation"] = imageOrientation
    o["premultiplyAlpha"] = premultiplyAlpha
    o["colorSpaceConversion"] = colorSpaceConversion
    o["resizeWidth"] = resizeWidth
    o["resizeHeight"] = resizeHeight
    o["resizeQuality"] = resizeQuality
    return o
}

/**
 * Exposes the JavaScript MessageEvent to Kotlin
 */
public external open class MessageEvent(type: String, eventInitDict: MessageEventInit = definedExternally) : Event {
    open val data: Any?
    open val origin: String
    open val lastEventId: String
    open val source: UnionMessagePortOrWindowProxy?
    open val ports: Array<MessagePort>
    fun initMessageEvent(type: String, bubbles: Boolean, cancelable: Boolean, data: Any?, origin: String, lastEventId: String, source: UnionMessagePortOrWindowProxy?, ports: Array<MessagePort>)

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface MessageEventInit : EventInit {
        var data: Any? /* = null */
        var origin: String? /* = "" */
        var lastEventId: String? /* = "" */
        var source: UnionMessagePortOrWindowProxy? /* = null */
        var ports: Array<MessagePort>? /* = arrayOf() */
    }

    @kotlin.internal.InlineOnly
        companion object {
            val NONE: Short
            val CAPTURING_PHASE: Short
            val AT_TARGET: Short
            val BUBBLING_PHASE: Short
        }

        fun initMessageEvent(type: String, bubbles: Boolean, cancelable: Boolean, data: Any?, origin: String, lastEventId: String, source: UnionMessagePortOrWindowProxy?, ports: Array<MessagePort>)
```
Short

|\n|\nunpublic external interface EventSourceInit {\n| var withCredentials: Boolean? /* = false */\n| get() = definedExternally\n| set(value) = definedExternally\n|\nun@kotlin.internal.InlineOnly\nunpublic inline fun EventSourceInit(withCredentials: Boolean? = false): EventSourceInit {\n| val o = js("({})")\n| o["withCredentials"] = withCredentials\n| return o\n|}\n
* Exposes the JavaScript

[WebSocket](https://developer.mozilla.org/en/docs/Web/API/WebSocket) to Kotlin

*unpublic external open class WebSocket(url: String, protocols: dynamic = definedExternally) : EventTarget {\n| open val url: String\n| open val readyState: Short\n| open val open bufferedAmount: Number\n|\n| var onopen: ((Event) -> dynamic)?\n| var onerror: ((Event) -> dynamic)?\n| var open onmessage: ((MessageEvent) -> dynamic)?\n|\n| var binaryType: BinaryType\n|\n| fun close(code: Short = definedExternally, reason: String = definedExternally)\n| fun send(data: String)\n| fun send(data: ArrayBuffer)\n| fun send(data: ArrayBufferView)\n
* Exposes the JavaScript

[CloseEvent](https://developer.mozilla.org/en/docs/Web/API/CloseEvent) to Kotlin

*unpublic external open class CloseEvent(type: String, eventInitDict: CloseEventInit = definedExternally) : Event {\n|\n| wasClean: Boolean? /* = false */\n| get() = definedExternally\n| set(value) = definedExternally\n|\n| var code: Short? /* = 0 */\n| get() = definedExternally\n| set(value) = definedExternally\n|\n| var reason: String? /* = "" */\n| get() = definedExternally\n| set(value) = definedExternally\n
* Exposes the JavaScript

[MessageChannel](https://developer.mozilla.org/en/docs/Web/API/MessageChannel) to Kotlin

*unpublic external open class MessageChannel {\n| port1: MessagePort\n| port2: MessagePort\n
* Exposes the JavaScript

[BroadcastChannel](https://developer.mozilla.org/en/docs/Web/API/BroadcastChannel) to Kotlin

*unpublic external open class BroadcastChannel(name: String) : EventTarget {\n|\n| name: String\n|\n| fun postMessage(message: Any?)\n| fun broadcast(message: Any?)\n
* Exposes the JavaScript

[WorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WorkerGlobalScope) to Kotlin

*unpublic external abstract class WorkerGlobalScope : EventTarget, WindowOrWorkerGlobalScope,\n|\n|GlobalPerformance {\n| open val self: WorkerGlobalScope\n| open val location: WorkerLocation\n| open val navigator: WorkerNavigator\n| open val onerror: ((dynamic, String, Int, Any?) -> dynamic)?\n| open var onlanguagechange: ((Event) -> dynamic)?\n| open var onoffline: ((Event) -> dynamic)?\n| open var ononline: ((Event) -> dynamic)?\n| open var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?\n| fun importScripts(vararg urls: String)\n
* Exposes the JavaScript
String
    open val applicationCache: ApplicationCache
    open var onconnect: ((Event) -> dynamic)?
    fun close()
}

/**
 * Exposes the JavaScript
 [AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to Kotlin
 */
public external interface AbstractWorker {
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)

/**
 * Exposes the JavaScript
 [Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin
 */
public external open class Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {
    var onmessage: ((MessageEvent) -> dynamic)?
    override var onerror: ((Event) -> dynamic)?
    fun terminate()
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}

/**
 * Exposes the JavaScript
 [SharedWorker](https://developer.mozilla.org/en/docs/Web/API/SharedWorker) to Kotlin
 */
public external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {
    open val port: MessagePort
    override var onerror: ((Event) -> dynamic)?
}

/**
 * Exposes the JavaScript
 [NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to Kotlin
 */
public external interface NavigatorConcurrentHardware {
    val hardwareConcurrency: Number
}

/**
 * Exposes the JavaScript
 [WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin
 */
public external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorConcurrentHardware {
    open val serviceWorker: ServiceWorkerContainer
}

/**
 * Exposes the JavaScript
 [WorkerLocation](https://developer.mozilla.org/en/docs/Web/API/WorkerLocation) to Kotlin
 */
public external abstract class WorkerLocation {
    open val href: String
    open val origin: String
    open val protocol: String
    open val host: String
    open val hostname: String
    open val port: String
    open val pathname: String
    open val search: String
    open val hash: String
}

/**
 * Exposes the JavaScript
 [Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to Kotlin
 */
public external abstract class Storage {
    open val length: Int
    fun key(index: Int): String?
    fun removeItem(key: String)
    fun clear()
    fun getItem(key: String): String?
    fun setItem(key: String, value: String)
}

@kotlin.internal.InlineOnly
public inline operator fun Storage.get(key: String): String? = asDynamic()[key]

@kotlin.internal.InlineOnly
public inline operator fun Storage.set(key: String, value: String) { asDynamic()[key] = value }

/**
 * Exposes the JavaScript
 [WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin
 */
public external interface WindowSessionStorage {
    val sessionStorage: Storage
}

/**
 * Exposes the JavaScript
 [WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage) to Kotlin
 */
public external interface WindowLocalStorage {
    val localStorage: Storage
}

/**
 * Exposes the JavaScript
 [StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin
 */
public external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) : Event {
    open val key: String?
    open val oldValue: String?
    open val newValue: String?
    open val url: String
    open val storageArea: Storage?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface StorageEventInit : EventInit {
    var key: String? /* = null */
    var oldValue: String? /* = null */
    var newValue: String? /* = null */
    var url: String
}

Open Source Used In DNAC Platform Antelope 8230
definedExternally
var storageArea: Storage? /* = null */
    get() = definedExternally
    set(value) =
definedExternally
    var storageArea: Storage? /* = null */
        get() = definedExternally
        set(value) =

@kotlin.internal.InlineOnly
public inline fun StorageEventInit(key: String? = null,
    val o =
    o["key"] = key
    o["oldValue"] = oldValue
    o["newValue"] = newValue
    o["url"] = url
    o["storageArea"] = storageArea
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    return o
}

public external abstract class HTMLAppletElement :
    HTMLElement {
    open var align: String
    open var alt: String
    open var archive: String
    open var code: String
    open var codeBase: String
    open var height: String
    open var hspace: Int
    open var name: String
    open var _object: String
    open var vspace: Int
    open var width: String

    companion object {
    val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE:
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
 [HTMLMarqueeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMarqueeElement) to Kotlin
 */
public external abstract class HTMLMarqueeElement : HTMLElement {
    open var behavior: String
    open var bgColor: String
    open var direction: String
    open var height: String
    open var hspace: Int
    open var loop: Int
    open var scrollAmount: Int
    open var scrollDelay: Int
    open var trueSpeed: Boolean
    open var vspace: Int
    open var width: String
    open var onbounce: ((Event) -> dynamic)?
    open var onfinish: ((Event) -> dynamic)?
    open var onstart: ((Event) -> dynamic)?
    fun start()
    fun stop()

    companion object {
    val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE:
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript
 [HTMLFrameSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFrameSetElement) to Kotlin
 */
public external abstract class HTMLFrameSetElement : HTMLElement, WindowEventHandlers {
    open var cols: String
    open var rows: String

    companion object {
    val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE:
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class HTMLFrameElement : HTMLElement {
    open var name: String
    open var scrolling: String
    open var src: String
    open var frameborder: String
    open var longDesc: String
    open var noResize: Boolean
    open val
contentDocument: Document?
open val contentWindow: Window?
open var marginHeight: String
open var marginWidth: String

companion object {

val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINEDByUrl
= js(`{"{ }}"\n o["detail"] = detail\n o["bubbles"] = bubbles\n o["cancelable"] = cancelable\n o["composed"] = composed\n return o\n`)

```
public external interface EventListenerOptions {
    var capture: Boolean? /* = false */
    get() = definedExternally\n    set(value) = definedExternally\n    var once: Boolean? /* = false */
    get() = definedExternally\n    set(value) = definedExternally\n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n    \n
definedExternally

set(value) = definedExternally
var characterDataOldValue: Boolean

get() =
definedExternally

set(value) = definedExternally
var attributeFilter: Array<String>

get() =
definedExternally

set(value) = definedExternally
var attributeFilter: Array<String>

get() =
definedExternally

var characterDataOldValue: Boolean?

get() =
definedExternally

set(value) = definedExternally

var attributeFilter: Array<String>?

get() =
definedExternally

set(value) = definedExternally

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["childList"] = childList
    o["attributes"] = attributes
    o["characterData"] = characterData
    o["subtree"] = subtree
    o["attributeOldValue"] = attributeOldValue
    o["characterDataOldValue"] = characterDataOldValue
    o["attributeFilter"] = attributeFilter
    return o
}

/**
 * Exposes the JavaScript [MutationRecord](https://developer.mozilla.org/en/docs/Web/API/MutationRecord) to Kotlin
 */
public external abstract class MutationRecord {
    open val type: String
    open val target: Node
    open val addedNodes: NodeList
    open val removedNodes: NodeList
    open val previousSibling: Node?
    open val nextSibling: Node?
    open val attributeName: String?
    open val attributeNamespace: String?
    open val oldValue: String?
}

/**
 * Exposes the JavaScript [Node](https://developer.mozilla.org/en/docs/Web/API/Node) to Kotlin
 */
public external abstract class Node : EventTarget {
    open val nodeType: Short
    open val nodeName: String
    open val baseURI: String
    open val isConnected: Boolean
    open val ownerDocument: Document?
    open val parentNode: Node?
    open val parentElement: Element?
    open val childNodes: NodeList
    open val firstChild: Node?
    open val lastChild: Node?
    open val previousSibling: Node?
    open val nextSibling: Node?
    open var nodeValue: String?
    open var textContent: String?
    fun getRootNode(options: GetRootNodeOptions = definedExternally): Node
    fun hasChildNodes(): Boolean
    fun normalize(): Node
    fun cloneNode(deep: Boolean = definedExternally): Node
    fun isEqualNode(otherNode: Node): Boolean
    fun isSameNode(otherNode: Node): Boolean
    fun compareDocumentPosition(other: Node): Short
    fun contains(other: Node): Boolean
    fun lookupPrefix(namespace: String): String
    fun lookupNamespaceURI(prefix: String): String
    fun isDefaultNamespace(namespace: String): Boolean
    fun insertBefore(node: Node, child: Node?): Node
    fun appendChild(node: Node): Node
    fun replaceChild(node: Node, child: Node): Node
    fun removeChild(child: Node): Node

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

public external interface GetRootNodeOptions {
    var composed: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun GetRootNodeOptions(composed: Boolean? = false): GetRootNodeOptions {
    val o = js("({})")
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [Document](https://developer.mozilla.org/en/docs/Web/API/Document) to Kotlin
 */
    open val implementation: DOMImplementation
    open val URL: String
    open val documentURI: String
    open val origin: String
    open val compatMode: String
    open val characterSet: String
    open val inputEncoding: String
    open val contentType: String
    open val doctype: DocumentType
    open val documentElement: Element
    open val openValLocation: Location
    open val openValDomain: String
    open val openValReferrer: String
    open val openValCookie: String
    open val openValLastModified: String
    open val openValReadyState: DocumentReadyState
    open val openValTitle: String
    open val openValDir: String
    open val openValBody: HTMLElement
    open val openValHead: HTMLHeadElement
    open val openValImages: HTMLCollection
    open val openValEmbeds: HTMLCollection
    open val openValPlugins: HTMLCollection
    open val openValNodes: NodeList
    open val openValRemovedNodes: NodeList
    open val openValPreviousSibling: Node
    open val openValNextSibling: Node
    open val openValAttributeName: String
    open val openValAttributeNamespace: String
    open val openValOldValue: String
    open val openValTextContent: String
    open val openValRootNode: Node
    open val openValParentNode: Node
    open val openValElement: Element
    open val openValChildNodes: NodeList
    open val openValFirstChild: Node
    open val openValLastChild: Node
    open val openValPreviousSibling: Node
    open val openValNextSibling: Node
    open val openValnodeValue: String
    open val openValtextContent: String
    fun getRootNode(options: GetRootNodeOptions = definedExternally): Node
    fun hasChildNodes(): Boolean
    fun normalize(): Node
    fun cloneNode(deep: Boolean = definedExternally): Node
    fun isEqualNode(otherNode: Node): Boolean
    fun isSameNode(otherNode: Node): Boolean
    fun compareDocumentPosition(other: Node): Short
    fun contains(other: Node): Boolean
    fun lookupPrefix(namespace: String): String
    fun lookupNamespaceURI(prefix: String): String
    fun isDefaultNamespace(namespace: String): Boolean
    fun insertBefore(node: Node, child: Node): Node
    fun appendChild(node: Node): Node
    fun replaceChild(node: Node, child: Node): Node
    fun removeChild(child: Node): Node

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection
fun getElementsByTagName(classNames: String): HTMLCollection
fun createElement(localName: String, options: ElementCreationOptions = definedExternally): Element
fun createDocumentFragment(): DocumentFragment
fun createTextNode(data: String): Text
fun createCDATASection(data: String): CDATASection
fun createComment(data: String): Comment
fun createProcessingInstruction(target: String, data: String): ProcessingInstruction
fun importNode(node: Node, deep: Boolean = definedExternally): Node
fun adoptNode(node: Node): Node
fun createAttribute(localName: String): Attr
fun createAttributeNS(namespace: String?, qualifiedName: String): Attr
fun createEvent(“interface”: String): Event
fun createRange(): Range
fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally): TreeWalker
fun getElementsByName(elementName: String): NodeList
fun open(type: String = definedExternally, replace: String = definedExternally): Document
fun open(url: String, name: String, features: String): Window
fun close(): Unit
fun write(vararg text: String)
fun writeln(vararg text: String)
fun hasFocus(): Boolean
fun execCommand(commandId: String, showUI: Boolean = definedExternally, value: String = definedExternally): Boolean
fun queryCommandEnabled(commandId: String): Boolean
fun queryCommandIndeterm(commandId: String): Boolean
fun queryCommandState(commandId: String): Boolean
fun queryCommandSupported(commandId: String): Boolean
fun clear(): Unit
fun captureEvents(): Unit
fun releaseEvents(): Unit
fun elementFromPoint(x: Double, y: Double): Element?
fun elementsFromPoint(x: Double, y: Double): Array<Element>
fun caretPositionFromPoint(x: Double, y: Double): CaretPosition?
fun createTouchList(vararg touches: Touch): TouchList
fun exitFullscreen(): Promise<Unit>

ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
ElementCreationOptions { /* var `is`: String? */
    get() = definedExternally
    set(value) = definedExternally
}

public external interface ElementCreationOptions {
    var `is`: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ElementCreationOptions(`is`: String? = undefined): ElementCreationOptions {
    val o = js("({})")
    o["is"] = `is`
    return o
}

/**
 * Exposes the JavaScript [DOMImplementation](https://developer.mozilla.org/en/docs/Web/API/DOMImplementation) to Kotlin
 */
public external abstract class DOMImplementation {
    fun createDocumentType(qualifiedName: String, publicId: String, systemId: String): DocumentType
    fun createDocument(namespace: String?, qualifiedName: String, doctype: DocumentType? = definedExternally): XMLDocument
    fun createHTMLDocument(title: String = definedExternally): Document
    fun hasFeature(): Boolean
}

/**
 * Exposes the JavaScript [DocumentType](https://developer.mozilla.org/en/docs/Web/API/DocumentType) to Kotlin
 */
public external abstract class DocumentType : Node, ChildNode {
    open val name: String
    open val publicId: String
    open val systemId: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [DocumentFragment](https://developer.mozilla.org/en/docs/Web/API/DocumentFragment) to Kotlin
 */
public external open class DocumentFragment : Node, NonElementParentNode, ParentNode {
    override val children: HTMLCollection
    override val firstElementChild: Element?
    override val lastElementChild: Element?
    override val childElementCount: Int
    override fun getElementById(elementId: String): Element?
    override fun append(vararg nodes: dynamic)
    override fun prepend(vararg nodes: dynamic)
    override fun.querySelector(selectors: String): Element?
    override fun querySelectorAll(selectors: String): NodeList

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [ShadowRoot](https://developer.mozilla.org/en/docs/Web/API/ShadowRoot) to Kotlin
 */
public external open class ShadowRoot : DocumentFragment, DocumentOrShadowRoot {
    open val mode: ShadowRootMode
    open val host: Element

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
Exposes the JavaScript [Element](https://developer.mozilla.org/en/docs/Web/API/Element) to Kotlin

```kotlin
/**
 * Exposes the JavaScript [Element](https://developer.mozilla.org/en/docs/Web/API/Element) to Kotlin
 */
public external abstract class Element : Node, ParentNode, NonDocumentTypeChildNode, ChildNode, Slotable, GeometryUtils, UnionElementOrHTMLCollection, UnionElementOrRadioNodeList, UnionElementOrMouseEvent, UnionElementOrProcessingInstruction {
    open val namespaceURI: String?
    open val prefix: String?
    open var id: String
    open var className: String
    open val classList: DOMTokenList
    open var slot: String
    open val attributes: NamedNodeMap
    open val shadowRoot: ShadowRoot?
    open var scrollTop: Double
    open var scrollLeft: Double
    open val scrollWidth: Int
    open val scrollHeight: Int
    open val clientTop: Int
    open val clientLeft: Int
    open val clientWidth: Int
    open val clientHeight: Int
    open var innerHTML: String
    open var outerHTML: String

    fun hasAttributes(): Boolean
    fun getAttributeNames(): Array<String>
    fun getAttribute(qualifiedName: String): String?
    fun getAttributeNS(namespace: String?, localName: String): String?
    fun setAttribute(qualifiedName: String, value: String)
    fun setAttributeNS(namespace: String?, qualifiedName: String, value: String)
    fun removeAttribute(qualifiedName: String)
    fun removeAttributeNS(namespace: String?, localName: String)
    fun hasAttribute(qualifiedName: String): Boolean
    fun hasAttributeNS(namespace: String?, localName: String): Boolean
    fun getAttributeNode(qualifiedName: String): Attr?
    fun getAttributeNodeNS(namespace: String?, localName: String): Attr?
    fun setAttributeNode(attr: Attr): Attr?
    fun setAttributeNodeNS(attr: Attr): Attr?
    fun removeAttributeNode(attr: Attr)
    fun attachShadow(init: ShadowRootInit): ShadowRoot
    fun closest(selectors: String): Element?
    fun matches(selectors: String): Boolean
    fun webkitMatchesSelector(selectors: String): Boolean
    fun getElementsByTagName(qualifiedName: String): HTMLCollection
    fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection
    fun getElementsByClassName(classNames: String): HTMLCollection
    fun insertAdjacentElement(where: String, element: Element): Element?
    fun insertAdjacentText(where: String, data: String)
    fun getClientRects(): Array<DOMRect>
    fun getBoundingClientRect(): DOMRect
    fun scrollIntoView()
    fun scrollIntoView(arg: dynamic)
    fun scrollBy(): ScrollToOptions
    fun scrollBy(x: Double, y: Double)
    fun scrollBy(options: ScrollToOptions = definedExternally)
    fun scrollTo(): ScrollToOptions
    fun scrollTo(x: Double, y: Double)
    fun scrollToBy(): ScrollToOptions
    fun scrollToBy(x: Double, y: Double)
    fun insertAdjacentHTML(position: String, text: String)
    fun setPointerCapture(pointerId: Int)
    fun releasePointerCapture(pointerId: Int)
    fun requestFocusMode(): boolean
    fun requestFullscreen(): boolean
    fun requestPointerCapture(pointerId: Int)
    fun releasePointerCapture(pointerId: Int)
    fun hasPointerCapture(pointerId: Int): boolean
    fun getBoundingClientRect(): DOMRect
    fun getClientRects(): Array<DOMRect>
    fun scrollIntoView(): void
    fun scrollBy(): void
    fun scrollBy(x: Double, y: Double)
    fun scrollBy(options: ScrollToOptions = definedExternally)
    fun scrollTo(): void
    fun scrollTo(x: Double, y: Double)
    fun scrollToBy(): void
    fun scrollToBy(x: Double, y: Double)
    fun insertAdjacentHTML(): void
    fun setPointerCapture(): void
    fun releasePointerCapture(): void
    fun requestFocusMode(): void
    fun requestFullscreen(): void
    fun requestPointerCapture(): void
    fun releasePointerCapture(): void
    fun hasPointerCapture(): boolean
    fun getBoundingClientRect(): DOMRect
    fun getClientRects(): Array<DOMRect>
    fun scrollIntoView(): void
    fun scrollBy(): void
    fun scrollBy(x: Double, y: Double)
    fun scrollBy(options: ScrollToOptions = definedExternally)
    fun scrollTo(): void
    fun scrollTo(x: Double, y: Double)
    fun scrollToBy(): void
    fun scrollToBy(x: Double, y: Double)
    fun insertAdjacentHTML(): void
    fun setPointerCapture(): void
    fun releasePointerCapture(): void
    fun requestFocusMode(): void
    fun requestFullscreen(): void
    fun requestPointerCapture(): void
    fun releasePointerCapture(): void
    fun hasPointerCapture(): boolean
    fun getBoundingClientRect(): DOMRect
    fun getClientRects(): Array<DOMRect>
    fun scrollIntoView(): void
    fun scrollBy(): void
    fun scrollBy(x: Double, y: Double)
    fun scrollBy(options: ScrollToOptions = definedExternally)
    fun scrollTo(): void
    fun scrollTo(x: Double, y: Double)
    fun scrollToBy(): void
    fun scrollToBy(x: Double, y: Double)
    fun insertAdjacentHTML(): void
    fun setPointerCapture(): void
    fun releasePointerCapture(): void
    fun requestFocusMode(): void
    fun requestFullscreen(): void
    fun requestPointerCapture(): void
    fun releasePointerCapture(): void
    fun hasPointerCapture(): boolean
    fun getBoundingClientRect(): DOMRect
    fun getClientRects(): Array<DOMRect>
    fun scrollIntoView(): void
    fun scrollBy(): void
    fun scrollBy(x: Double, y: Double)
    fun scrollBy(o...
ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val
PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val
DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val
DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val
DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val
DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val
DOCUMENT_POSITION_CONTAINED_BY: Short
val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
]* Exposes the JavaScript [CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin

ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
]* Exposes the JavaScript [ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin

ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
]* Exposes the JavaScript [Comment](https://developer.mozilla.org/en/docs/Web/API/Comment) to Kotlin

ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
]* Exposes the JavaScript [Range](https://developer.mozilla.org/en/docs/Web/API/Range) to Kotlin

ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
]*
definedExternally

var w: Double? /* = 1.0 */
    get() = definedExternally
    set(value) = definedExternally

var w: Double? /* = 1.0 */
    get() = definedExternally
    set(value) = definedExternally

DOMRect(x: Double = definedExternally, y: Double = definedExternally, width: Double = definedExternally,
    height: Double = definedExternally): DOMRectReadOnly

DOMRectInit(x: Double? /* = 0.0 */
    get() = definedExternally
    set(value) = definedExternally

DOMRectInit(x: Double? /* = 0.0 */
    get() = definedExternally
    set(value) = definedExternally

DOMRectReadOnly(x: Double, y: Double, width: Double, height: Double)

DOMRectList: ItemArrayLike<DOMRect>

DOMMatrixReadOnly(numberSequence: Array<Double>)

DOMMatrix(numberSequence: Array<Double>)
DOMMatrixReadOnly
constructor(arr32: Float32Array)
constructor(arr64: Float64Array)
constructor(numberSequence: Array<Double>)
override var a: Double
override var b: Double
override var c: Double
override var d: Double
override var e: Double
override var f: Double
override var m11: Double
override var m12: Double
override var m13: Double
override var m14: Double
override var m21: Double
override var m22: Double
override var m23: Double
override var m24: Double
override var m31: Double
override var m32: Double
override var m33: Double
override var m34: Double
override var m41: Double
override var m42: Double
override var m43: Double
override var m44: Double
fun multiplySelf(other: DOMMatrix): DOMMatrix
fun preMultiplySelf(other: DOMMatrix): DOMMatrix
fun translateSelf(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix
fun scaleSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
fun scale3dSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
fun scaleNonUniformSelf(scaleX: Double, scaleY: Double = definedExternally, scaleZ: Double = definedExternally): DOMMatrix
fun skewXSelf(sx: Double): DOMMatrix
fun skewYSelf(sy: Double): DOMMatrix
fun invertSelf(): DOMMatrix
fun setMatrixValue(transformList: String): DOMMatrix

public external interface ScrollOptions {
    var behavior: ScrollBehavior? /* = ScrollBehavior.AUTO */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ScrollOptions(behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollOptions {
    val o = js("({})")
    o["behavior"] = behavior
    return o
}

/**
 * Exposes the JavaScript
[ScrollToOptions](https://developer.mozilla.org/en/docs/Web/API/ScrollToOptions) to Kotlin
*/
public external interface ScrollToOptions : ScrollOptions {
    var left: Double?
    get() = definedExternally
    set(value) = definedExternally
    var top: Double?
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["left"] = left
    o["top"] = top
    o["behavior"] = behavior
    return o
}

/**
 * Exposes the JavaScript
[MediaQueryList](https://developer.mozilla.org/en/docs/Web/API/MediaQueryList) to Kotlin
*/
public external abstract class MediaQueryList : EventTarget {
    open val media: String
    open val matches: Boolean
    open var onchange: ((Event) -> dynamic)?
    fun addListener(listener: EventListener?)
    fun addListener(listener: ((Event) -> Unit)?)
    fun removeListener(listener: EventListener?)
    fun removeListener(listener: ((Event) -> Unit)?)
}

/**
 * Exposes the JavaScript
[MediaQueryListEvent](https://developer.mozilla.org/en/docs/Web/API/MediaQueryListEvent) to Kotlin
*/
public external open class MediaQueryListEvent(type: String, eventInitDict: MediaQueryListEventInit = definedExternally) : Event {
    open val media: String
    open val matches: Boolean
}

companion object {
    val NONE: Short
    val CAPTURING_PHASE: Short
    val AT_TARGET: Short
    val BUBBLING_PHASE: Short
}

public external interface MediaQueryListEventInit : EventInit {
    var media: String? /* = "" */
    get() = definedExternally
    set(value) = definedExternally
    var matches: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun MediaQueryListEventInit(media: String? = 
    val o = js("({})")
    o["media"] = media
    o["matches"] = matches
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

val * Exposes the JavaScript
[CaretPosition](https://developer.mozilla.org/en/docs/Web/API/CaretPosition) to Kotlin
*/

val *
external abstract class CaretPosition {
    open val offsetNode: Node
    open val offset: Int
    fun getClientRect(): DOMRect?
}

external abstract class ScrollIntoViewOptions {
    open fun scrollTo(block: ScrollLogicalPosition?, inline: ScrollLogicalPosition?): Unit
}

public external interface ScrollIntoViewOptions {
    val offsetNode: Node
    val offset: Int
    val block: ScrollLogicalPosition?
    val inline: ScrollLogicalPosition?
    fun getClientRect(): DOMRect?
}

public external interface ScrollIntoViewOptions : ScrollIntoViewOptions {
    var block: ScrollLogicalPosition?
    var inline: ScrollLogicalPosition?
}

@kotlin.internal.InlineOnly
public inline fun ScrollIntoViewOptions(block: ScrollLogicalPosition?, inline: ScrollLogicalPosition?): Unit = scrollIntoViewOptions(block).get()

public external interface BoxQuadOptions {
    var box: CSSBoxType?
    var relativeTo: dynamic
}

@kotlin.internal.InlineOnly
public inline fun BoxQuadOptions(box: CSSBoxType?, relativeTo: dynamic): BoxQuadOptions = boxQuadOptions(box, relativeTo)

public external interface ConvertCoordinateOptions {
    var fromBox: CSSBoxType?
    var toBox: CSSBoxType?
}

@kotlin.internal.InlineOnly
public inline fun ConvertCoordinateOptions(fromBox: CSSBoxType?, toBox: CSSBoxType?): ConvertCoordinateOptions = convertCoordinateOptions(fromBox, toBox)

/**
 * Exposes the JavaScript [GeometryUtils](https://developer.mozilla.org/en/docs/Web/API/GeometryUtils) to Kotlin
 */
public external interface GeometryUtils {
    fun getBoxQuads(options: BoxQuadOptions): Array<DOMQuad>
    fun convertQuadFromNode(quad: dynamic, options: ConvertCoordinateOptions): DOMQuad
    fun convertRectFromNode(rect: DOMRectReadOnly, options: ConvertCoordinateOptions): DOMQuad
    fun convertPointFromNode(point: DOMPointInit, options: ConvertCoordinateOptions): DOMPoint
}

/**
 * Exposes the JavaScript [Touch](https://developer.mozilla.org/en/docs/Web/API.Touch) to Kotlin
 */
public external abstract class Touch {
    open val identifier: Int
    open val target: EventTarget
    open val screenX: Int
    open val screenY: Int
    open val clientX: Int
    open val clientY: Int
    open val pageX: Int
    open val pageY: Int
    open val region: String?
}

public external abstract class TouchList : ItemArrayLike<Touch> {
    fun item(index: Int): Touch?
}

@kotlin.internal.InlineOnly
public inline operator fun TouchList.get(index: Int): Touch? = asDynamic()[index]

public external open class TouchEvent : UIEvent {
    open val touches: TouchList
    open val targetTouches: TouchList
    open val changedTouches: TouchList
    open val altKey: Boolean
    open val metaKey: Boolean
    open val ctrlKey: Boolean
    open val shiftKey: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

/**
 * Exposes the JavaScript [Image](https://developer.mozilla.org/en/docs/Web/API/Image) to Kotlin
 */
public external open class Image(width: Int, height: Int) : HTMLImageElement {
    override var onabort: ((Event) -> dynamic)?
    override var onblur: ((FocusEvent) -> dynamic)?
    override var oncancel: ((Event) -> dynamic)?
    override var onchange: ((Event) -> dynamic)?
    override var oncontextmenu: ((MouseEvent) -> dynamic)?
    override var ondblclick: ((MouseEvent) -> dynamic)?
    override var ondrag: ((DragEvent) -> dynamic)?
    override var ondragend: ((DragEvent) -> dynamic)?
    override var ondragenter: ((DragEvent) -> dynamic)?
    override var ondragleave: ((DragEvent) -> dynamic)?
    override var ondragover: ((DragEvent) -> dynamic)?
    override var ondragstart: ((DragEvent) -> dynamic)?
}
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

Audio(src: String = definedExternally): HTMLAudioElement { override var onabort: ((Event) -> dynamic)?
override var onblur: ((FocusEvent) -> dynamic)?
override var oncancel: ((Event) -> dynamic)?
override var oncanplay: ((Event) -> dynamic)?
override var oncanplaythrough: ((Event) -> dynamic)?
override var onchange: ((Event) -> dynamic)?
override var oncomplete: ((Event) -> dynamic)?
override var oncontextmenu: ((MouseEv}
override fun remove() { override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */) :
Array<DOMQuad>
override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */) : DOMQuad
override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */) : DOMQuad

companion object {
    val NETWORK_EMPTY: Short
    val NETWORK_IDLE: Short
    val NETWORK_LOADING: Short
    val NETWORK_NO_SOURCE: Short
    val HAVE_NOTHING: Short
    val HAVE_METADATA: Short
    val HAVE_CURRENT_DATA: Short
    val HAVE_FUTURE_DATA: Short
    val HAVE_ENOUGH_DATA: Short
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [Option](https://developer.mozilla.org/en/docs/Web/API/Option) to Kotlin
 */
public external open class Option(text: String = definedExternally, value: String = definedExternally, defaultSelected: Boolean = definedExternally, selected: Boolean = definedExternally) : HTMLOptionElement {
    override var onabort: ((Event) -> dynamic)?
    override var onblur: ((FocusEvent) -> dynamic)?
    override var oncancel: ((Event) -> dynamic)?
    override var oncanplay: ((Event) -> dynamic)?
    override var oncanplaythrough: ((Event) -> dynamic)?
    override var onchange: ((Event) -> dynamic)?
    override var onclick: ((MouseEvent) -> dynamic)?
    override var onclose: ((Event) -> dynamic)?
    override var oncontextmenu: ((MouseEvent) -> dynamic)?
    override var oncuechange: ((Event) -> dynamic)?
    override var ondblclick: ((MouseEvent) -> dynamic)?
    override var ondrag: ((DragEvent) -> dynamic)?
    override var ondragend: ((DragEvent) -> dynamic)?
    override var ondragenter: ((DragEvent) -> dynamic)?
    override var ondragexit: ((DragEvent) -> dynamic)?
    override var ondragleave: ((DragEvent) -> dynamic)?
    override var ondragover: ((DragEvent) -> dynamic)?
    override var ondragstart: ((DragEvent) -> dynamic)?
    override var ondrop: ((DragEvent) -> dynamic)?
    override var ondurationchange: ((Event) -> dynamic)?
    override var onemptied: ((Event, String, Int, Int, Any?) -> dynamic)?
    override var onended: ((Event) -> dynamic)?
    override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?
    override var onfocus: ((FocusEvent) -> dynamic)?
    override var oninput: ((InputEvent) -> dynamic)?
    override var oninvalid: ((Event) -> dynamic)?
    override var onkeydown: ((KeyboardEvent) -> dynamic)?
    override var onkeypress: ((KeyboardEvent) -> dynamic)?
    override var onkeyup: ((KeyboardEvent) -> dynamic)?
    override var onmousedown: ((MouseEvent) -> dynamic)?
    override var onmouseenter: ((MouseEvent) -> dynamic)?
    override var onmouseleave: ((MouseEvent) -> dynamic)?
    override var onmousemove: ((MouseEvent) -> dynamic)?
    override var onmouseout: ((MouseEvent) -> dynamic)?
    override var onmouseover: ((MouseEvent) -> dynamic)?
    override var onmouseup: ((MouseEvent) -> dynamic)?
    override var onwheel: ((WheelEvent) -> dynamic)?
    override var onpause: ((Event) -> dynamic)?
    override var onplay: ((Event) -> dynamic)?
    override var onplaying: ((Event) -> dynamic)?
    override var onprogress: ((ProgressEvent) -> dynamic)?
    override var onratechange: ((Event) -> dynamic)?
    override var onreset: ((Event) -> dynamic)?
    override var onresize: ((Event) -> dynamic)?
    override var onscroll: ((Event) -> dynamic)?
    override var onseeked: ((Event) -> dynamic)?
    override var onseeking: ((Event) -> dynamic)?
    override var onselect: ((Event) -> dynamic)?
    override var onshow: ((Event) -> dynamic)?
    override var onstalled: ((Event) -> dynamic)?
CanPlayTypeResult {
    companion object
}

public inline val CanPlayTypeResult.Companion.EMPTY: CanPlayTypeResult get() = "".asDynamic().unsafeCast<CanPlayTypeResult>()

public inline val CanPlayTypeResult.Companion.MAYBE: CanPlayTypeResult get() = "maybe".asDynamic().unsafeCast<CanPlayTypeResult>()

public inline val CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() = "probably".asDynamic().unsafeCast<CanPlayTypeResult>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface TextTrackMode {
    companion object
}

public inline val TextTrackMode.Companion.DISABLED: TextTrackMode get() = "disabled".asDynamic().unsafeCast<TextTrackMode>()

public inline val TextTrackMode.Companion.HIDDEN: TextTrackMode get() = "hidden".asDynamic().unsafeCast<TextTrackMode>()

public inline val TextTrackMode.Companion.SHOWING: TextTrackMode get() = "showing".asDynamic().unsafeCast<TextTrackMode>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface TextTrackKind {
    companion object
}

public inline val TextTrackKind.Companion.SUBTITLES: TextTrackKind get() = "subtitles".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.CAPTIONS: TextTrackKind get() = "captions".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.DESCRIPTIONS: TextTrackKind get() = "descriptions".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.CHAPTERS: TextTrackKind get() = "chapters".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.METADATA: TextTrackKind get() = "metadata".asDynamic().unsafeCast<TextTrackKind>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface SelectionMode {
    companion object
}

public inline val SelectionMode.Companion.SELECT: SelectionMode get() = "select".asDynamic().unsafeCast<SelectionMode>()

public inline val SelectionMode.Companion.START: SelectionMode get() = "start".asDynamic().unsafeCast<SelectionMode>()

public inline val SelectionMode.Companion.END: SelectionMode get() = "end".asDynamic().unsafeCast<SelectionMode>()

public inline val SelectionMode.Companion.PRESERVE: SelectionMode get() = "preserve".asDynamic().unsafeCast<SelectionMode>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasFillRule {
    companion object
}

public inline val CanvasFillRule.Companion.NONZERO: CanvasFillRule get() = "nonzero".asDynamic().unsafeCast<CanvasFillRule>()

public inline val CanvasFillRule.Companion.EVENODD: CanvasFillRule get() = "evenodd".asDynamic().unsafeCast<CanvasFillRule>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ImageSmoothingQuality {
    companion object
}


/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasLineCap {
    companion object
}

public inline val CanvasLineCap.Companion.BUTT: CanvasLineCap get() = "butt".asDynamic().unsafeCast<CanvasLineCap>()

public inline val CanvasLineCap.Companion.ROUND: CanvasLineCap get() = "round".asDynamic().unsafeCast<CanvasLineCap>()

public inline val CanvasLineCap.Companion.SQUARE: CanvasLineCap get() =

Open Source Used In DNAC Platform Antelope 8249
/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasLineJoin {
    companion object
}

public inline val CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() = 
    "round".asDynamic().unsafeCast<CanvasLineJoin>()

public inline val CanvasLineJoin.Companion.BEVEL: CanvasLineJoin get() = 
    "bevel".asDynamic().unsafeCast<CanvasLineJoin>()

public inline val CanvasLineJoin.Companion.MITER: CanvasLineJoin get() = 
    "miter".asDynamic().unsafeCast<CanvasLineJoin>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasTextAlign {
    companion object
}

public inline val CanvasTextAlign.Companion.START: CanvasTextAlign get() = 
    "start".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.END: CanvasTextAlign get() = 
    "end".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.LEFT: CanvasTextAlign get() = 
    "left".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.RIGHT: CanvasTextAlign get() = 
    "right".asDynamic().unsafeCast<CanvasTextAlign>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasTextBaseline {
    companion object
}

public inline val CanvasTextBaseline.Companion.TOP: CanvasTextBaseline get() = 
    "top".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.HANGING: CanvasTextBaseline get() = 
    "hanging".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.MIDDLE: CanvasTextBaseline get() = 
    "middle".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.ALPHABETIC: CanvasTextBaseline get() = 
    "alphabetic".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.IDEOGRAPHIC: CanvasTextBaseline get() = 
    "ideographic".asDynamic().unsafeCast<CanvasTextBaseline>()

public inline val CanvasTextBaseline.Companion.BOTTOM: CanvasTextBaseline get() = 
    "bottom".asDynamic().unsafeCast<CanvasTextBaseline>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasDirection {
    companion object
}

public inline val CanvasDirection.Companion.LTR: CanvasDirection get() = 
    "ltr".asDynamic().unsafeCast<CanvasDirection>()

public inline val CanvasDirection.Companion.RTL: CanvasDirection get() = 
    "rtl".asDynamic().unsafeCast<CanvasDirection>()

public inline val CanvasDirection.Companion.INHERIT: CanvasDirection get() = 
    "inherit".asDynamic().unsafeCast<CanvasDirection>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollRestoration {
    companion object
}

public inline val ScrollRestoration.Companion.AUTO: ScrollRestoration get() = 
    "auto".asDynamic().unsafeCast<ScrollRestoration>()

public inline val ScrollRestoration.Companion.MANUAL: ScrollRestoration get() = 
    "manual".asDynamic().unsafeCast<ScrollRestoration>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_INEXTERNALINTERFACE")
public external interface ImageOrientation {
    companion object
}

    "none".asDynamic().unsafeCast<ImageOrientation>()

    "flipY".asDynamic().unsafeCast<ImageOrientation>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface PremultiplyAlpha {
    companion object
}

public inline val PremultiplyAlpha.Companion.NONE: PremultiplyAlpha get() = 
    "none".asDynamic().unsafeCast<PremultiplyAlpha>()

public inline val PremultiplyAlpha.Companion.PREMULTIPLY: PremultiplyAlpha get() =
"premultiply\".asDynamic().unsafeCast<PremultiplyAlpha>()

public inline val PremultiplyAlpha.Companion.DEFAULT: PremultiplyAlpha get() = "default".asDynamic().unsafeCast<PremultiplyAlpha>()

@Suppress("NESTED_CLASS_INEXTERNAL_INTERFACE")
public external interface ColorSpaceConversion {
    companion object
}

public inline val ColorSpaceConversion.Companion.NONE: ColorSpaceConversion get() = "none".asDynamic().unsafeCast<ColorSpaceConversion>()

@Suppress("NESTED_CLASS_INEXTERNAL_INTERFACE")
public external interface ResizeQuality {
    companion object
}

public inline val ResizeQuality.Companion.PIXELATED: ResizeQuality get() = "pixelated".asDynamic().unsafeCast<ResizeQuality>()

public inline val ResizeQuality.Companion.LOW: ResizeQuality get() = "low".asDynamic().unsafeCast<ResizeQuality>()

public inline val ResizeQuality.Companion.MEDIUM: ResizeQuality get() = "medium".asDynamic().unsafeCast<ResizeQuality>()

public inline val ResizeQuality.Companion.HIGH: ResizeQuality get() = "high".asDynamic().unsafeCast<ResizeQuality>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_INEXTERNAL_INTERFACE")
public external interface ShadowRootMode {
    companion object
}

public inline val ShadowRootMode.Companion.OPEN: ShadowRootMode get() = "open".asDynamic().unsafeCast<ShadowRootMode>()

public inline val ShadowRootMode.Companion.CLOSED: ShadowRootMode get() = "closed".asDynamic().unsafeCast<ShadowRootMode>()

@Suppress("NESTED_CLASS_INEXTERNAL_INTERFACE")
public external interface ScrollBehavior {
    companion object
}

public inline val ScrollBehavior.Companion.AUTO: ScrollBehavior get() = "auto".asDynamic().unsafeCast<ScrollBehavior>()

public inline val ScrollBehavior.Companion.INSTANT: ScrollBehavior get() = "instant".asDynamic().unsafeCast<ScrollBehavior>()

public inline val ScrollBehavior.Companion.SMOOTH: ScrollBehavior get() = "smooth".asDynamic().unsafeCast<ScrollBehavior>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_INEXTERNAL_INTERFACE")
public external interface ScrollLogicalPosition {
    companion object
}


public inline val ScrollLogicalPosition.Companion.CENTER: ScrollLogicalPosition get() = "center".asDynamic().unsafeCast<ScrollLogicalPosition>()

public inline val ScrollLogicalPosition.Companion.END: ScrollLogicalPosition get() = "end".asDynamic().unsafeCast<ScrollLogicalPosition>()

public inline val ScrollLogicalPosition.Companion.NEAREST: ScrollLogicalPosition get() = "nearest".asDynamic().unsafeCast<ScrollLogicalPosition>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_INEXTERNAL_INTERFACE")
public external interface CSSBoxType {
    companion object
}

public inline val CSSBoxType.Companion.MARGIN: CSSBoxType get() = "margin".asDynamic().unsafeCast<CSSBoxType>()

public inline val CSSBoxType.Companion.BORDER: CSSBoxType get() = "border".asDynamic().unsafeCast<CSSBoxType>()

public inline val CSSBoxType.Companion.PADDING: CSSBoxType get() =
"padding".asDynamic().unsafeCast<CSSBoxType>().\n
public inline val CSSBoxType.Companion.CONTENT: CSSBoxType get() = "content".asDynamic().unsafeCast<CSSBoxType>()" /*
 * Copyright 2010-2019 JetBrains
 * * Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED,
DO NOT EDIT!!
See github.com/kotlin/dukat for details!
package org.w3c.fetch

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*

private val BODY_TAG = "body"

@kotlin.internal.InlineOnly
public inline fun RequestInit(method: String? = undefined,
headers: dynamic = undefined, body: dynamic = undefined, referrer: String? = undefined, referrerPolicy: dynamic = undefined,
mode: RequestMode? = undefined, credentials: RequestCredentials? = undefined, cache: RequestCache? = undefined,
redirect: RequestRedirect? = undefined, integrity: String? = undefined, keepalive: Boolean? = undefined,
window: Any? = undefined?): RequestInit {
    val o = js("({})")
    o["method"] = method
    o["headers"] = headers
    o["body"] = body
    o["referrer"] = referrer
    o["referrerPolicy"] = referrerPolicy
    o["mode"] = mode
    o["credentials"] = credentials
    o["cache"] = cache
    o["redirect"] = redirect
    o["integrity"] = integrity
    o["keepalive"] = keepalive
    o["window"] = window
    return o
}
Boolean
open val status: Short
open val ok: Boolean
open val statusText: String
open val bodyUsed: Boolean
fun clone(): Response
override fun arrayBuffer(): Promise<ArrayBuffer>
override fun blob(): Promise<Blob>
override fun formData(): Promise<FormData>
override fun json(): Promise<Any?>
override fun text(): Promise<String>

companion object {
fun error(): Response
fun redirect(url: String, status: Short = definedExternally): Response
}

public external interface ResponseInit {
var status: Short? /* = 200 */
get() = definedExternally
set(value) = definedExternally
var statusText: String? /* = "OK" */
get() = definedExternally
set(value) = definedExternally
var headers: dynamic
get() = definedExternally
set(value) = definedExternally
var headers:
get() = definedExterna
set(value) = definedExternally

fun @kotlin.internal.InlineOnly
public inline fun ResponseInit(status: Short? = 200,
statusText: String? = "OK", headers: dynamic = undefined): ResponseInit {
val o = js("({})")
o["status"] = status
o["statusText"] = statusText
o["headers"] = headers
return o
}" please, don't implement this interface! */

public external interface RequestType {
companion object
}

public inline val RequestType.Companion.EMPTY: RequestType get() = "".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.AUDIO: RequestType get() = "audio".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.FONT: RequestType get() = "font".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.IMAGE: RequestType get() = "image".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.SCRIPT: RequestType get() = "script".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.STYLE: RequestType get() = "style".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.TRACK: RequestType get() = "track".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.VIDEO: RequestType get() = "video".asDynamic().unsafeCast<RequestType>()

}" please, don't implement this interface!

public external interface RequestDestination {
companion object
}

public inline val RequestDestination.Companion.EMPTY: RequestDestination get() = "".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.DOCUMENT: RequestDestination get() = "document".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.EMBED: RequestDestination get() = "embed".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.FONT: RequestDestination get() = "font".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.IMAGE: RequestDestination get() = "image".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.MANIFEST: RequestDestination get() = "manifest".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.MEDIA: RequestDestination get() = "media".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.OBJECT: RequestDestination get() = "object".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.REPORT: RequestDestination get() = "report".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.SERVEWORKER: RequestDestination get() = "serviceworker".asDynamic().unsafeCast<RequestDestination>()
RequestDestination.Companion.SHAREDWORKER: RequestDestination get() =
"sharedworker".asDynamic().unsafeCast<RequestDestination>()

RequestDestination.Companion.STYLE: RequestDestination get() =
"style".asDynamic().unsafeCast<RequestDestination>()

RequestDestination.Companion.WORKER: RequestDestination get() =
"worker".asDynamic().unsafeCast<RequestDestination>()

RequestDestination.Companion.XSLT: RequestDestination get() =
"xslt".asDynamic().unsafeCast<RequestDestination>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestMode {

    companion object
}

public inline val RequestMode.Companion.NAVIGATE: RequestMode get() =
"navigate".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.SAME_ORIGIN: RequestMode get() = "same-origin".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.NO_CORS: RequestMode get() = "no-cors".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.CORS: RequestMode get() = "cors".asDynamic().unsafeCast<RequestMode>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestCredentials {

    companion object
}

public inline val RequestCredentials.Companion.OMIT: RequestCredentials get() = "omit".asDynamic().unsafeCast<RequestCredentials>()

public inline val RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = "same-origin".asDynamic().unsafeCast<RequestCredentials>()

public inline val RequestCredentials.Companion.INCLUDE: RequestCredentials get() = "include".asDynamic().unsafeCast<RequestCredentials>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestCache {

    companion object
}

public inline val RequestCache.Companion.DEFAULT: RequestCache get() = "default".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.NO_STORE: RequestCache get() = "no-store".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.RELOAD: RequestCache get() = "reload".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.NO_CACHE: RequestCache get() = "no-cache".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.FORCE_CACHE: RequestCache get() = "force-cache".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = "only-if-cached".asDynamic().unsafeCast<RequestCache>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestRedirect {

    companion object
}

public inline val RequestRedirect.Companion.FOLLOW: RequestRedirect get() = "follow".asDynamic().unsafeCast<RequestRedirect>()

public inline val RequestRedirect.Companion.ERROR: RequestRedirect get() = "error".asDynamic().unsafeCast<RequestRedirect>()

public inline val RequestRedirect.Companion.MANUAL: RequestRedirect get() = "manual".asDynamic().unsafeCast<RequestRedirect>()

/* please, don't implement this interface!
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ResponseType {

    companion object
}

public inline val ResponseType.Companion.BASIC: ResponseType get() = "basic".asDynamic().unsafeCast<ResponseType>()

public inline val ResponseType.Companion.CORS: ResponseType get() = "cors".asDynamic().unsafeCast<ResponseType>()

public inline val ResponseType.Companion.DEFAUL
"opaque".asDynamic().unsafeCast<ResponseType>()

public inline val ResponseType.Companion.OPAQUEREDIRECT: ResponseType get() =
"opaqueredirect".asDynamic().unsafeCast<ResponseType>()
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeModel"] = resizeModel
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}

public external interface MediaTrackCapabilities {
    var width: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var height: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var facingMode: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
    var resizeMode: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
    var volume: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var sampleSize: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var echoCancellation: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var autoGainControl: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var noiseSuppression: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var latency: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: String?
        get() = definedExternally
        set(value) = definedExternally
    var groupId: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeModel"] = resizeModel
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}

/**
 * Exposes the JavaScript
[MediaTrackConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackConstraints) to Kotlin
 */
public external interface MediaTrackConstraints : MediaTrackConstraintSet {
    var advanced: Array<MediaTrackConstraintSet>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["advanced"] = advanced
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeModel"] = resizeModel
}
```javascript
import { O } from 'open-source-used-in-dnac-platform-antelope-8257'

// Example usage:
const mediaTrackConstraintSet = O.flexible
    .set(width: 800, height: 600)
    .set(aspectRatio: 16/9)
    .set(frameRate: 30)
    .set(facingMode: 'environmental')
    .set(resizeMode: 'stretch')
    .set(volume: 0.5)
    .set(sampleRate: 44100)
    .set(sampleSize: 16)
    .set(echoCancellation: true)
    .set(autoGainControl: true)
    .set(noiseSuppression: true)
    .set(latency: 50)
    .set(channelCount: 2)
    .set(deviceId: '1234567890')
    .set(groupId: 'abc123')

console.log(mediaTrackConstraintSet)
```

This code snippet demonstrates how to create a `MediaTrackConstraintSet` object in JavaScript. The `O` object is from the `open-source-used-in-dnac-platform-antelope-8257` library, which provides functions for creating and manipulating media track constraints.

The `MediaTrackConstraintSet` constructor takes several parameters, such as `width`, `height`, `aspectRatio`, `frameRate`, `facingMode`, `resizeMode`, `volume`, `sampleRate`, `sampleSize`, `echoCancellation`, `autoGainControl`, `noiseSuppression`, `latency`, `channelCount`, `deviceId`, and `groupId`. Each of these parameters can be set to a specific value using dot notation.

The constructor also supports chaining methods to create a fluent interface, making it easier to build complex constraints.

```javascript
const mediaTrackSettings = O.flexible
    .setWidth(800)
    .setHeight(600)
    .setAspectRatio(16/9)
    .setFrameRate(30)
    .setFacingMode('environmental')
    .setResizeMode('stretch')
    .setVolume(0.5)
    .setSampleRate(44100)
    .setSampleSize(16)
    .setEchoCancellation(true)
    .setAutoGainControl(true)
    .setNoiseSuppression(true)
    .setLatency(50)
    .setChannelCount(2)
    .deviceId('1234567890')
    .groupId('abc123')

console.log(mediaTrackSettings)
```

This code snippet is similar to the previous one but uses the `set` method for each property, providing a more concise way to build the constraint set.

These examples illustrate how to use the library to create and configure media track constraints in JavaScript.
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}
```kotlin
interface DoubleRange {
    var max: Double?
        get() = definedExternally
        set(value) = definedExternally
    var min: Double?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun DoubleRange(max: Double? = undefined, min: Double? = undefined): DoubleRange {
    val o = js("{}")
    o["max"] = max
    o["min"] = min
    return o
}

public external interface ConstrainDoubleRange : DoubleRange {
    var exact: Double?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Double?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    o["max"] = max
    o["min"] = min
    return o
}

public external interface ULongRange {
    var max: Int?
        get() = definedExternally
        set(value) = definedExternally
    var min: Int?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ULongRange(max: Int? = undefined, min: Int? = undefined): ULongRange {
    val o = js("{}")
    o["max"] = max
    o["min"] = min
    return o
}

public external interface ConstrainULongRange : ULongRange {
    var exact: Int?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Int?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    o["max"] = max
    o["min"] = min
    return o
}

/**
 * Exposes the JavaScript [ConstrainBooleanParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainBooleanParameters) to Kotlin
 */
public external interface ConstrainBooleanParameters {
    var exact: Boolean?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Boolean?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    return o
}

/**
 * Exposes the JavaScript [ConstrainDOMStringParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainDOMStringParameters) to Kotlin
 */
public external interface ConstrainDOMStringParameters {
    var exact: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var ideal: dynamic
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ConstrainDOMStringParameters(exact: dynamic = undefined, ideal: dynamic = undefined): ConstrainDOMStringParameters {
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    return o
}
```

---

Open Source Used In DNAC Platform Antelope 8259
false): PointerEventInit {\n    val o = js("({})")\n    o["pointerId"] = pointerId\n    o["width"] = width\n    o["height"] = height\n    o["pressure"] = pressure\n    o["tangentialPressure"] = tangentialPressure\n    o["tiltX"] = tiltX\n    o["tiltY"] = tiltY\n    o["twist"] = twist\n    o["pointerType"] = pointerType\n    o["isPrimary"] = isPrimary\n    o["screenX"] = screenX\n    o["screenY"] = screenY\n    o["clientX"] = clientX\n    o["clientY"] = clientY\n    o["button"] = button\n    o["buttons"] = buttons\n    o["relatedTarget"] = relatedTarget\n    o["region"] = region\n    o["ctrlKey"] = ctrlKey\n    o["shiftKey"] = shiftKey\n    o["altKey"] = altKey\n    o["metaKey"] = metaKey\n    o["modifierAltGraph"] = modifierAltGraph\n    o["modifierCapsLock"] = modifierCapsLock\n    o["modifierFn"] = modifierFn\n    o["modifierLock"] = modifierLock\n    o["modifierNumLock"] = modifierNumLock\n    o["modifierScrollLock"] = modifierScrollLock\n    o["modifierSuper"] = modifierSuper\n    o["modifierSymbol"] = modifierSymbol\n    o["view"] = view\n    o["detail"] = detail\n    o["bubbles"] = bubbles\n    o["cancelable"] = cancelable\n    o["composed"] = composed\n    return o
}

/**
 * Exposes the JavaScript
 * [PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent) to Kotlin\n */
public external open class PointerEvent(type: String, eventInitDict: PointerEventInit = definedExternally) : MouseEvent {
    open val pointerId: Int
    open val width: Double
    open val height: Double
    open val pressure: Float
    open val tangentialPressure: Float
    open val tiltX: Int
    open val tiltY: Int
    open val twist: Int
    open val pointerType: String
    open val isPrimary: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

/*
* Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*/

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n
package org.w3c.dom.svg

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript
 */
public external abstract class SVGElement : Element, ElementCSSInlineStyle, GlobalEventHandlers, SVGElementInstance {
    open val dataset: DOMStringMap
    open val ownerSVGElement: SVGSVGElement?
    open val viewportElement: SVGElement?
    open var tabIndex: Int
    fun focus()
    fun blur()

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
    }
}

@kotlin.internal.InlineOnly
    val o = js("({})")\n    o["fill"] = fill\n    o["stroke"] = stroke\n    o["markers"] = markers\n    o["clipped"] = clipped\n    return o
}

/**
 * Exposes the JavaScript
 */
public external abstract class SVGGraphicsElement : SVGElememt, SVGTests {
    open val transform: SVGAnimatedTransformList
    fun getBBox(options: SVGBoundingBoxOptions = definedExternally):
    DOMRect
    fun getCTM(): DOMMatrix?
    fun getScreenCTM(): DOMMatrix?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGGeometryElement : SVGGraphicsElement {
    open val pathLength: SVGAnimatedNumber
    fun isPointInFill(point: DOMPoint): Boolean
    fun isPointInStroke(point: DOMPoint): Boolean
    fun getTotalLength(): Float
    fun getPointAtLength(distance: Float): DOMPoint

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGNumber {
    open var value: Float
}

public external abstract class SVGLength {
    open val unitType: Short
    open var value: Float
    open var valueInSpecifiedUnits: Float
    open var valueAsString: String
    fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)
    fun convertToSpecifiedUnits(unitType: Short)

    companion object {
        val SVG_LENGTHTYPE_UNKNOWN: Short
        val SVG_LENGTHTYPE_NUMBER: Short
        val SVG_LENGTHTYPE_PERCENTAGE: Short
        val SVG_LENGTHTYPE_EMS: Short
        val SVG_LENGTHTYPE_EXS: Short
        val SVG_LENGTHTYPE_PX: Short
        val SVG_LENGTHTYPE_CM: Short
        val SVG_LENGTHTYPE_MM: Short
        val SVG_LENGTHTYPE_IN: Short
        val SVG_LENGTHTYPE_PT: Short
        val SVG_LENGTHTYPE_PC: Short
    }
}

public external abstract class SVGAngle {
    open val unitType: Short
    open var value: Float
    open var valueInSpecifiedUnits: Float
    open var valueAsString: String
    fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)
    fun convertToSpecifiedUnits(unitType: Short)

    companion object {
        val SVG_ANGLETYPE_UNKNOWN: Short
        val SVG_ANGLETYPE_UNSPECIFIED: Short
        val SVG_ANGLETYPE_DEG: Short
        val SVG_ANGLETYPE_RAD: Short
        val SVG_ANGLETYPE_GRAD: Short
        val SVG_ANGLETYPE_CM: Short
        val SVG_ANGLETYPE_MM: Short
        val SVG_ANGLETYPE_IN: Short
        val SVG_ANGLETYPE_PT: Short
        val SVG_ANGLETYPE_PC: Short
    }

    public external abstract class SVGNameList {
        open val length: Int
        open val numberOfItems: Int
        fun clear()
        fun initialize(newItem: dynamic): dynamic
        fun insertItemBefore(newItem: dynamic, index: Int): dynamic
        fun replaceItem(newItem: dynamic, index: Int): dynamic
        fun removeItem(index: Int): dynamic
        fun appendItem(newItem: dynamic): dynamic
        fun
getItem(index: Int): dynamic

getItem(index: Int): dynamic = asDynamic()[index]

Exposes the JavaScript [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGNumberList) to Kotlin

*public external abstract class SVGNumberList {

    open val length: Int
    open val numberOfItems: Int

    fun clear()
    fun initialize(newItem: SVGNumber)
    fun insertItemBefore(newItem: SVGNumber, index: Int)
    fun replaceItem(newItem: SVGNumber, index: Int)
    fun get(index: Int)
    fun get(index: Int): SVGNumber?
    fun set(index: Int, newItem: SVGNumber)
    fun set(index: Int, newItem: SVGNumber?)

    /**
     * Exposes the JavaScript [SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
     */
    public external abstract class SVGLengthList {

        open val length: Int
        open val numberOfItems: Int

        fun clear()
        fun initialize(newItem: SVGLength)
        fun insertItemBefore(newItem: SVGLength, index: Int)
        fun replaceItem(newItem: SVGLength, index: Int)
        fun get(index: Int)
        fun get(index: Int): SVGLength?
        fun set(index: Int, newItem: SVGLength)
        fun set(index: Int, newItem: SVGLength?)

        /**
         * Exposes the JavaScript [SVGAnimatedBoolean](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedBoolean) to Kotlin
         */
        public external abstract class SVGAnimatedBoolean {

            open var baseVal: Boolean
            open val animVal: Boolean

            /**
             * Exposes the JavaScript [SVGAnimatedEnumeration](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedEnumeration) to Kotlin
             */
            public external abstract class SVGAnimatedEnumeration {

                open var baseVal: Short
                open val animVal: Short

                /**
                 * Exposes the JavaScript [SVGAnimatedInteger](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedInteger) to Kotlin
                 */
                public external abstract class SVGAnimatedInteger {

                    open var baseVal: Int
                    open val animVal: Int

                    /**
                     * Exposes the JavaScript [SVGAnimatedNumber](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumber) to Kotlin
                     */
                    public external abstract class SVGAnimatedNumber {

                        open var baseVal: Float
                        open val animVal: Float

                        /**
                         * Exposes the JavaScript [SVGAnimatedAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAngle) to Kotlin
                         */
                        public external abstract class SVGAnimatedAngle {

                            open var baseVal: SVGAngle
                            open val animVal: SVGAngle

                            /**
                             * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin
                             */
                            public external abstract class SVGAnimatedNumberList {

                                open var baseVal: SVGNumberList
                                open val animVal: SVGNumberList

                                /**
                                 * Exposes the JavaScript [SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin
                                 */
                                public external abstract class SVGAnimatedLengthList {

                                    open var baseVal: SVGLengthList
                                    open val animVal: SVGLengthList

                                    /**
                                     * Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect) to Kotlin
                                     */
                                    public external abstract class SVGAnimatedRect {

                                        open var baseVal: DOMRect
                                        open val animVal: DOMRect

                                        /**
                                         * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin
                                         */
                                        public external abstract class SVGAnimatedNumberList {

                                            open var baseVal: SVGNumberList
                                            open val animVal: SVGNumberList

                                            /**
                                             * Exposes the JavaScript [SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin
                                             */
                                            public external abstract class SVGAnimatedLengthList {

                                                open var baseVal: SVGLengthList
                                                open val animVal: SVGLengthList

                                                /**
                                                 * Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect) to Kotlin
                                                 */
                                                public external abstract class SVGAnimatedRect {

                                                    open var baseVal: DOMRect
                                                    open val animVal: DOMRect

                                                    /**
                                                     * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin
                                                     */
                                                    public external abstract class SVGAnimatedNumberList {

                                                        open var baseVal: SVGNumberList
                                                        open val animVal: SVGNumberList

                                                        /**
                                                         * Exposes the JavaScript [SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin
                                                         */
                                                         public external abstract class SVGAnimatedLengthList {

                                                             open var baseVal: SVGLengthList
                                                             open val animVal: SVGLengthList

                                                             /**
                                                              * Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect) to Kotlin
                                                              */
                                                              public external abstract class SVGAnimatedRect {

                                                                  open var baseVal: DOMRect
                                                                  open val animVal: DOMRect

                                                                  /**
                                                                   * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin
                                                                   */
                                                                   public external abstract class SVGAnimatedNumberList {

                                                                       open var baseVal: SVGNumberList
                                                                       open val animVal: SVGNumberList

                                                                       /**
                                                                        * Exposes the JavaScript [SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin
                                                                        */
                                                                        public external abstract class SVGAnimatedLengthList {

                                                                            open var baseVal: SVGLengthList
                                                                            open val animVal: SVGLengthList

                                                                            /**
                                                                             * Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect) to Kotlin
                                                                             */
                                                                             public external abstract class SVGAnimatedRect {

                                                                                 open var baseVal: DOMRect
                                                                                 open val animVal: DOMRect

                                                                                 /**
                                                                                  * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin
                                                                                  */
                                                                                  public external abstract class SVGAnimatedNumberList {

                                                                                      open var baseVal: SVGNumberList
                                                                                      open val animVal: SVGNumberList

                                                                                      /**
                                                                                       * Exposes the JavaScript [SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin
                                                                                       */
                                                                                       public external abstract class SVGAnimatedLengthList {

                                                                                           open var baseVal: SVGLengthList
                                                                                           open val animVal: SVGLengthList

                                                                                           /**
                                                                                            * Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect) to Kotlin
                                                                                           */
                                                                                           public external abstract class SVGAnimatedRect {

                                                                                                           open var baseVal: DOMRect
                                                                                                           open val animVal: DOMRect

                                                                                                           /**
                                                                                                            * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin
                                                                                                            */
                                                                                                            public external abstract class SVGAnimatedNumberList {

                                                                                                                open var baseVal: SVGNumberList
                                                                                                                open val animVal: SVGNumberList
                                                                                                                **/
animVal: SVGLengthList\n
```kotlin
public external abstract class SVGStringList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun insertItemAtBefore(newItem: String, index: Int): String
    fun replaceItem(newItem: String, index: Int): String
    fun removeItem(index: Int): String
    fun appendItem(newItem: String): String
    fun getItem(index: Int): String
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline operator fun SVGStringList.get(index: Int): String? =
```

```kotlin
@kotlin.internal.InlineOnly
public inline operator fun SVGStringList.set(index: Int, newItem: String) {
    asDynamic()[index] = newItem
}
```

```kotlin
public external interface SVGUnitTypes {
    companion object {
        val SVG_UNIT_TYPE_UNKNOWN: Short
        val SVG_UNIT_TYPE_USERSPACEONUSE: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
    }
}
```

```kotlin
/**
 * Exposes the JavaScript
 */
public external interface SVGTests {
    val requiredExtensions: SVGStringList
    val systemLanguage: SVGStringList
}
```

```kotlin
/**
 * Exposes the JavaScript
 */
public external interface SVGFitToViewBox {
    val viewBox: SVGAnimatedRect
    val preserveAspectRatio: SVGAnimatedPreserveAspectRatio
}
```

```kotlin
/**
 * Exposes the JavaScript
 */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface SVGZoomAndPan {
    var zoomAndPan: Short

    companion object {
        val SVG_ZOOMANDPAN_UNKNOWN: Short
        val SVG_ZOOMANDPAN_DISABLE: Short
        val SVG_ZOOMANDPAN_MAGNIFY: Short
    }
}
```

```kotlin
/**
 * Exposes the JavaScript
 */
public external interface SVGURIReference {
    val href: SVGAnimatedString
}
```

```kotlin
/**
 * Exposes the JavaScript
 */
public external abstract class SVGSVGElement : SVGGraphicsElement, SVGFitToViewBox, SVGZoomAndPan,
```

```kotlin
open val x: SVGAnimatedLength
open val y: SVGAnimatedLength
open val width: SVGAnimatedLength
open val height: SVGAnimatedLength
open var currentScale: Float
open var currentTranslate: DOMPointReadOnly
open fun getIntersectionList(rect: DOMRectReadOnly, referenceElement: SVGElement?): NodeList
open fun getEnclosureList(rect: DOMRectReadOnly, referenceElement: SVGElement?): NodeList
open fun checkIntersection(rect: DOMRect): Boolean
open fun checkEnclosure(rect: DOMRect): Boolean
open fun deselectAll()
open fun createSVGNumber(): SVGNumber
open fun createSVGLength(): SVGLength
open fun createSVGAngle(): SVGAngle
open fun createSVGPoint(): DOMPoint
open fun createSVGMatrix(): DOMMatrix
open fun createSVGRect(): DOMRect
open fun createSVGTransform(): SVGTransform
open fun createSVGTransformFromMatrix(matrix: DOMMatrix): SVGTransform
open fun getElementById(elementId: String): Element
open fun suspendRedraw(maxWaitMilliseconds: Int): Int
open fun unsuspendRedraw(suspendHandleID: Int): Int
open fun unsuspendRedrawAll(): Int
open fun forceRedraw(): Int
open companion object {
    val SVG_ZOOMANDPAN_UNKNOWN: Short
    val SVG_ZOOMANDPAN_DISABLE: Short
    val SVG_ZOOMANDPAN_MAGNIFY: Short
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITYREFERENCE_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
```
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGGElement](https://developer.mozilla.org/en/docs/Web/API/SVGGElement) to Kotlin
 */
public external abstract class SVGGElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGUnknownElement](https://developer.mozilla.org/en/docs/Web/API/SVGUnknownElement) to Kotlin
 */
public external abstract class SVGUnknownElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGDefsElement](https://developer.mozilla.org/en/docs/Web/API/SVGDefsElement) to Kotlin
 */
public external abstract class SVGDefsElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGDescElement](https://developer.mozilla.org/en/docs/Web/API/SVGDescElement) to Kotlin
 */
public external abstract class SVGDescElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGMetadataElement](https://developer.mozilla.org/en/docs/Web/API/SVGMetadataElement) to Kotlin
 */
public external abstract class SVGMetadataElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
public external open class ShadowAnimation(source: dynamic, newTarget: dynamic) {
    open val sourceAnimation: dynamic
}

/**
 * Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement) to Kotlin
 */
public external abstract class SVGSwitchElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface GetSVGDocument {
    fun getSVGDocument(): Document
}

/**
 * Exposes the JavaScript [SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin
 */
public external abstract class SVGStyleElement : SVGElement, LinkStyle {
    open var type: String
    open var media: String
    open var title: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin
 */
public external abstract class SVGTransform {
    open val type: Short
    open val matrix: DOMMatrix
    open val angle: Float
    fun setMatrix(matrix: DOMMatrixReadOnly)
    fun setTranslate(tx: Float, ty: Float)
    fun setScale(sx: Float, sy: Float)
    fun setRotate(angle: Float, cx: Float, cy: Float)
    fun setSkewX(angle: Float)
    fun setSkewY(angle: Float)

    companion object {
        val SVG_TRANSFORM_UNKNOWN: Short
        val SVG_TRANSFORM_MATRIX: Short
        val SVG_TRANSFORM_TRANSLATE: Short
        val SVG_TRANSFORM_SCALE: Short
        val SVG_TRANSFORM_ROTATE: Short
        val SVG_TRANSFORM_SKEWX: Short
        val SVG_TRANSFORM_SKEWY: Short
    }
}

/**
 * Exposes the JavaScript [SVGTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGTransformList) to Kotlin
 */
public external abstract class SVGTransformList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGTransform): SVGTransform
    fun insertItemBefore(newItem: SVGTransform, index: Int): SVGTransform
    fun replaceItem(newItem: SVGTransform, index: Int): SVGTransform
    fun removeItem(index: Int): SVGTransform
    fun appendItem(newItem: SVGTransform): SVGTransform
    fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform
    fun consolidate(): SVGTransform?
    fun getItem(index: Int): SVGTransform
}
Open Source Used In DNAC Platform Antelope

SVGTransform::\n
\n
public inline operator fun SVGTransformList.get(index: Int): SVGTransform

\n
public inline operator fun SVGTransformList.set(index: Int, newItem: SVGTransform) { asDynamic()[index] = newItem }

\n
* Exposes the JavaScript

[SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to Kotlin

\n
* Exposes the JavaScript

[SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin

\n
* Exposes the JavaScript

[SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspectRatio) to Kotlin

\n
* Exposes the JavaScript

[SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin

\n
* Exposes the JavaScript

[SVGRectElement](https://developer.mozilla.org/en/docs/Web/API/SVGRectElement) to Kotlin

\n
* Exposes the JavaScript

[SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin

\n
* Exposes the JavaScript

[SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin
**Open Source Used in DNAC Platform Antelope**

```java
[SVGCircleElement](https://developer.mozilla.org/en/docs/Web/API/SVGCircleElement) to Kotlin

```*/

```java
public
external abstract class SVGCircleElement : SVGGeometryElement {
    open val cx: SVGAnimatedLength
    open val cy: SVGAnimatedLength
    open val r: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val ENTITY_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin
 */
public
external abstract class SVGEllipseElement : SVGGeometryElement {
    open val cx: SVGAnimatedLength
    open val cy: SVGAnimatedLength
    open val rx: SVGAnimatedLength
    open val ry: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val ENTITY_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin
 */
public
external abstract class SVGLineElement : SVGGeometryElement {
    open val x1: SVGAnimatedLength
    open val y1: SVGAnimatedLength
    open val x2: SVGAnimatedLength
    open val y2: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val ENTITY_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin
 */
public
external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val ENTITY_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGAnimatedPoints](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPoints) to Kotlin
 */
public
```
external interface SVGAnimatedPoints { \n    val points: SVGPointList \n    val animatedPoints: SVGPointList
}

public external abstract class SVGPointList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: DOMPoint): DOMPoint
    fun insertItemAtBefore(newItem: DOMPoint, index: Int): DOMPoint
    fun removeItemAt(index: Int): DOMPoint
    fun appendItemAt(newItem: DOMPoint): DOMPoint
    fun getItem(index: Int): DOMPoint
}

@kotlin.internal.InlineOnly
public inline operator fun SVGPointList.get(index: Int): DOMPoint? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGPointList.set(index: Int, newItem: DOMPoint) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGPolylineElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolylineElement) to Kotlin
 */
public external abstract class SVGPolylineElement : SVGGeometryElement, SVGAnimatedPoints {

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
    }
}

/**
 * Exposes the JavaScript
 [SVGPolygonElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolygonElement) to Kotlin
 */
public external abstract class SVGPolygonElement : SVGGeometryElement, SVGAnimatedPoints {

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
    }
}

/**
 * Exposes the JavaScript
 [SVGTextContentElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextContentElement) to Kotlin
 */
public external abstract class SVGTextContentElement : SVGGraphicsElement {
    open val textLength: SVGAnimatedLength
    open val lengthAdjust: SVGAnimatedEnumeration
    fun getNumberOfChars(): Int
    fun getComputedTextLength(): Float
    fun getSubStringLength(charnum: Int, nchars: Int): Float
    fun getStartPositionOfChar(charnum: Int): DOMPoint
    fun getEndPositionOfChar(charnum: Int): DOMPoint
    fun getExtentOfChar(charnum: Int): DOMRect
    fun getRotationOfChar(charnum: Int): Float
    fun getCharNumAtPosition(point: DOMPoint): Int
    fun selectSubString(charnum: Int, nchars: Int): DOMPoint
    fun getExtentsOfChar(charnum: Int): DOMRect
    fun getRotationOfChar(charnum: Int): Float
    fun getCharNumAtPosition(point: DOMPoint): Int
    fun selectSubString(charnum: Int, nchars: Int): Int
    companion object {
        val LENGTHADJUST_UNKNOWN: Short
        val LENGTHADJUST_SPACING: Short
        val LENGTHADJUST_SPACINGANDGLYPHS: Short
    }
}
val TEXTPATH_SPACINGTYPE_AUTO: Short
val TEXTPATH_SPACINGTYPE_EXACT: Short
val LENGTHADJUST_UNKNOWN: Short
val LENGTHADJUST_SPACING: Short
val LENGTHADJUST_SPACINGANDGLYPHS: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGImageElement](https://developer.mozilla.org/en/docs/Web/API/SVGImageElement) to Kotlin
 */
public external abstract class SVGImageElement : SVGGraphicsElement, SVGURIReference, HTMLOrSVGImageElement {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open var crossOrigin: String?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGForeignObjectElement](https://developer.mozilla.org/en/docs/Web/API/SVGForeignObjectElement) to Kotlin
 */
public external abstract class SVGForeignObjectElement : SVGGraphicsElement {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGMarkerElement : SVGElement, SVGFitToViewBox {
    open val refX: SVGAnimatedLength
    open val refY: SVGAnimatedLength
    open val markerUnits: SVGAnimatedEnumeration
    open val markerWidth: SVGAnimatedLength
    open val markerHeight: SVGAnimatedLength
    open val orientType: SVGAnimatedEnumeration
    open val orientAngle: SVGAnimatedAngle
    open var orient: String
    fun setOrientToAuto()
    fun setOrientToAngle(angle: SVGAngle)

    companion object {
        val SVG_MARKERUNITS_UNKNOWN: Short
        val SVG_MARKERUNITS_USERSPACEONUSE: Short
        val SVG_MARKERUNITS_STROKEWIDTH: Short
        val SVG_MARKER.ORIENT_UNKNOWN: Short
        val SVG_MARKER.ORIENT_AUTO: Short
        val SVG_MARKER.ORIENT_ANGLE: Short
    }
}
/**
 * Exposes the JavaScript [SVGSolidcolorElement](https://developer.mozilla.org/en/docs/Web/API/SVGSolidcolorElement) to Kotlin
 */
public external abstract class SVGSolidcolorElement : SVGElement {

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }

    /**
     * Exposes the JavaScript [SVGGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGGradientElement) to Kotlin
     */
    public external abstract class SVGGradientElement : SVGElement, SVGURIReference, SVGUnitTypes {
        open val gradientUnits: SVGAnimatedEnumeration
        open val gradientTransform: SVGAnimatedTransformList
        open val spreadMethod: SVGAnimatedEnumeration

        companion object {
            val SVG_SPREADMETHOD_UNKNOWN: Short
            val SVG_SPREADMETHOD_PAD: Short
            val SVG_SPREADMETHOD_REFLECT: Short
            val SVG_SPREADMETHOD_REPEAT: Short
            val SVG_UNIT_TYPE_UNKNOWN: Short
            val SVG_UNIT_TYPE_USERSPACEONUSE: Short
            val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val NOTATION_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
        }
    }

    /**
     * Exposes the JavaScript [SVGLinearGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGLinearGradientElement) to Kotlin
     */
    public external abstract class SVGLinearGradientElement : SVGGradientElement {
        open val x1: SVGAnimatedLength
        open val y1: SVGAnimatedLength
        open val x2: SVGAnimatedLength
        open val y2: SVGAnimatedLength

        companion object {
            val SVG_SPREADMETHOD_UNKNOWN: Short
            val SVG_SPREADMETHOD_PAD: Short
            val SVG_SPREADMETHOD_REFLECT: Short
            val SVG_SPREADMETHOD_REPEAT: Short
            val SVG_UNIT_TYPE_UNKNOWN: Short
            val SVG_UNIT_TYPE_USERSPACEONUSE: Short
            val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val NOTATION_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
        }
    }
/**
 * Exposes the JavaScript [SVGStopElement](https://developer.mozilla.org/en/docs/Web/API/SVGStopElement) to Kotlin
 */
public external abstract class SVGStopElement : SVGElement {
    open val offset: SVGAnimatedNumber

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGPatternElement](https://developer.mozilla.org/en/docs/Web/API/SVGPatternElement) to Kotlin
 */
public external abstract class SVGPatternElement : SVGElement, SVGFitToViewBox, SVGURIReference, SVGUnitTypes {
    open val patternUnits: SVGAnimatedEnumeration
    open val patternContentUnits: SVGAnimatedEnumeration
    open val patternTransform: SVGAnimatedTransformList
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength

    companion object {
        val SVG_UNIT_TYPE_UNKNOWN: Short
        val SVG_UNIT_TYPE_USERSPACEONUSE: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGHatchElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGHatchpathElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
Open Source Used In DNAC Platform Antelope 8276

DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGCursorElement](https://developer.mozilla.org/en/docs/Web/API/SVGCursorElement) to Kotlin
 */
public external abstract class SVGCursorElement : SVGElement, SVGURIReference {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGScriptElement](https://developer.mozilla.org/en/docs/Web/API/SVGScriptElement) to Kotlin
 */
public external abstract class SVGScriptElement : SVGElement, SVGURIReference, HTMLOrSVGScriptElement {
    open var type: String
    open var crossOrigin: String?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGAElement](https://developer.mozilla.org/en/docs/Web/API/SVGAElement) to Kotlin
 */
public external abstract class SVGAElement : SVGGraphicsElement, SVGURIReference {
    open val target: SVGAnimatedString
    open val download: SVGAnimatedString
    open val rel: SVGAnimatedString
    open val relList: SVGAnimatedString
    open val hreflang: SVGAnimatedString
    open val type: SVGAnimatedString

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGViewElement](https://developer.mozilla.org/en/docs/Web/API/SVGViewElement) to Kotlin
 */
public external abstract class SVGViewElement : SVGElement, SVGFitToViewBox, SVGZoomAndPan {
    companion object {
        val SVG_ZOOMANDPAN_UNKNOWN: Short
        val SVG_ZOOMANDPAN_DISABLE: Short
        val SVG_ZOOMANDPAN_MAGNIFY: Short
        val ENTITY_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
ATTRIBUTE_NODE, TEXT_NODE, CDATA_SECTION_NODE, ENTITY_REFERENCE_NODE, PROCESSING_INSTRUCTION_NODE, COMMENT_NODE, DOCUMENT_NODE, DOCUMENT_TYPE_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_PRECEDING, DOCUMENT_POSITION_FOLLOWING, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See https://github.com/kotlin/dukat for details

package org.w3c.files

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript
 * [Blob](https://developer.mozilla.org/en/docs/Web/API/Blob) to Kotlin
 */
public external open class Blob(blobParts: Array<dynamic> = definedExternally, options: BlobPropertyBag = definedExternally) :
    ImageBitmapSource {
    open val size: Number
    open val type: String
    open val isClosed: Boolean
    fun slice(start: Int = definedExternally, end: Int = definedExternally, contentType: String = definedExternally): Blob
    fun close()
}

public external interface BlobPropertyBag {
    var type: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun BlobPropertyBag(type: String? = ""): BlobPropertyBag {
    val o = js("({})")
    o["type"] = type
    return o
}

/**
 * Exposes the JavaScript
 * [File](https://developer.mozilla.org/en/docs/Web/API/File) to Kotlin
 */
public external open class File(fileBits: Array<dynamic>, fileName: String, options: FilePropertyBag = definedExternally) : Blob {
    open val name: String
    open val lastModified: Int
}

public external interface FilePropertyBag : BlobPropertyBag {
    var lastModified: Int?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun FilePropertyBag(lastModified: Int? = undefined, type: String? = ""): FilePropertyBag {
    val o = js("({})")
    o["lastModified"] = lastModified
    o["type"] = type
    return o
}

/**
 * Exposes the JavaScript
 * [FileList](https://developer.mozilla.org/en/docs/Web/API/FileList) to Kotlin
 */
public external abstract class FileList : ItemArrayLike<File> {
    override fun item(index: Int): File?
}

@kotlin.internal.InlineOnly
public operator fun FileList.get(index: Int): File? = asDynamic()[index]

/**
 * Exposes the JavaScript
 * [FileReader](https://developer.mozilla.org/en/docs/Web/API/ FileReader) to Kotlin
 */
public external open class FileReader : EventTarget {
    open val readyState: Short
    open val result: dynamic
    open val error: dynamic
    var onloadstart: ((ProgressEvent) -> dynamic)?
    var onprogress: ((ProgressEvent) -> dynamic)?
    var onload: ((Event) -> dynamic)?
    var onabort: ((Event) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    fun readAsArrayBuffer(blob: Blob): ArrayBuffer
    fun readAsBinaryString(blob: Blob): String
    fun readAsText(blob: Blob, label: String = definedExternally): String
    fun readAsDataURL(blob: Blob): String
    fun abort(): Unit
    companion object {
        val EMPTY: Short
        val LOADING: Short
        val DONE: Short
    }
}

/**
 * Exposes the JavaScript
 * [FileReaderSync](https://developer.mozilla.org/en/docs/Web/API/FileReaderSync) to Kotlin
 */
public external open class FileReaderSync {
    fun readAsArrayBuffer(blob: Blob): ArrayBuffer
    fun readAsBinaryString(blob: Blob): String
    fun readAsText(blob: Blob, label: String = definedExternally): String
    fun readAsDataURL(blob: Blob): String
}
package org.w3c.notifications

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.device.*
import org.w3c.dom.events.*
import org.w3c.dom.html.*
import org.w3c.dom.svg.*
import org.w3c.files.*
import org.w3c.performance.*
import org.w3c.workers.*

/**
 * Exposes the JavaScript [Notification](https://developer.mozilla.org/en/docs/Web/API/Notification) to Kotlin
 */
public external open class Notification(title: String, options: NotificationOptions = definedExternally) : EventTarget {
    var onclick: ((MouseEvent) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    open val title: String
    open val dir: NotificationDirection
    open val lang: String
    open val body: String
    open val tag: String
    open val image: String
    open val icon: String
    open val badge: String
    open val sound: String
    open val vibrate: Array<out Int>
    open val timestamp: Number
    open val renotify: Boolean
    open val silent: Boolean
    open val noscreen: Boolean
    open val requireInteraction: Boolean
    open val sticky: Boolean
    open val data: Any
    open val actions: Array<out NotificationAction>
    fun close()

    companion object {
        val permission: NotificationPermission
        val maxActions: Int
        fun requestPermission(deprecatedCallback: (NotificationPermission) -> Unit = definedExternally): Promise<NotificationPermission>
    }
}

public external interface NotificationOptions {
    var dir: NotificationDirection? /* = NotificationDirection.AUTO */
    var lang: String? /* = "" */
    var body: String? /* = "" */
    var tag: String? /* = "" */
    var image: String?
    var icon: String?
    var badge: String?
    var sound: String?
    var vibrate: dynamic
    var timestamp: Number?
    var renotify: Boolean? /* = false */
    var silent: Boolean? /* = false */
    var noscreen: Boolean? /* = false */
    var requireInteraction: Boolean? /* = false */
    var sticky: Boolean? /* = false */
    var data: Any? /* = null */
    var actions: Array<NotificationAction>?
}

@kotlin.internal.InlineOnly

@kotlin.internal.InlineOnly
```kotlin
@kotlin.internal.InlineOnly
public inline fun GetNotificationOptions(tag: String? = ""):
    GetNotificationOptions {
        val o = js("({})")
        o["tag"] = tag
        return o
    }

/**
 * Exposes the JavaScript NotificationEvent([NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin
 */
public external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) :
    ExtendableEvent {
        open val notification: Notification
        open val action: String

        companion object {
            val NONE: Short
            val CAPTURING_PHASE: Short
            val AT_TARGET: Short
            val BUBBLING_PHASE: Short
        }

        public external interface NotificationEventInit : ExtendableEventInit {
            var notification: Notification?
                get() = definedExternally
                set(value) = definedExternally
            var action: String? /* = "" */
                get() = definedExternally
                set(value) = definedExternally
        }

        @kotlin.internal.InlineOnly
            val o = js("({})")
            o["notification"] = notification
            o["action"] = action
            o["bubbles"] = bubbles
            o["cancelable"] = cancelable
            o["composed"] = composed
            return o
        }

        /* please, don't implement this interface! */
        @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
        public external interface NotificationPermission {
        }

        public inline val NotificationPermission.Companion.DEFAULT: NotificationPermission get() = "default".asDynamic().unsafeCast<NotificationPermission>()
        public inline val NotificationPermission.Companion.DENIED: NotificationPermission get() = "denied".asDynamic().unsafeCast<NotificationPermission>()
        public inline val NotificationPermission.Companion.GRANTED: NotificationPermission get() = "granted".asDynamic().unsafeCast<NotificationPermission>()

        /* please, don't implement this interface! */
        @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
        public external interface NotificationDirection {
        }

        public inline val NotificationDirection.Companion.AUTO: NotificationDirection get() = "auto".asDynamic().unsafeCast<NotificationDirection>()
        public inline val NotificationDirection.Companion.LTR: NotificationDirection get() = "ltr".asDynamic().unsafeCast<NotificationDirection>()
        public inline val NotificationDirection.Companion.RTL: NotificationDirection get() = "rtl".asDynamic().unsafeCast<NotificationDirection>()
```

---

See github.com/kotlin/dukat for details
NotificationOptions = definedExternally): Promise<Unit>
fun getNotifications(filter: GetNotificationOptions = definedExternally): Promise<Array<Notification>>
fun methodName(): Promise<dynamic>

/**
 * Exposes the JavaScript
 */
public external abstract class ServiceWorkerContainer : EventTarget {

    val controller: ServiceWorker?
    val ready: Promise<ServiceWorkerRegistration>
    var oncontrollerchange: ((Event) -> dynamic)?
    var onmessage: ((MessageEvent) -> dynamic)?

    fun register(scriptURL: String, options: RegistrationOptions = definedExternally): Promise<ServiceWorkerRegistration>
    fun getRegistration(clientURL: String = definedExternally): Promise<ServiceWorkerRegistration?>
    fun startMessages(): Promise<Unit>

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    fun getRegistrations(): Promise<Array<ServiceWorkerRegistration>>
}

public external interface RegistrationOptions {
    var scope: String?
        get() = definedExternally
        set(value) = definedExternally
    var type: WorkerType? /* = WorkerType.CLASSIC */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun RegistrationOptions(scope: String? = undefined, type: WorkerType? = WorkerType.CLASSIC): RegistrationOptions {
    val o = js("({})")
    o["scope"] = scope
    o["type"] = type
    return o
}

/**
 * Exposes the JavaScript
 */
public external open class ServiceWorkerMessageEvent(type: String, eventInitDict: ServiceWorkerMessageEventInit = definedExternally) : Event {
    val data: Any?
    val origin: String
    val lastEventId: String
    val source: UnionMessagePortOrServiceWorker?
    val ports: Array<MessagePort>?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    fun getRegistration(clientURL: String = definedExternally): Promise<ServiceWorkerRegistration>
}

public external interface ServiceWorkerMessageEventInit : EventInit {
    var data: Any?
        get() = definedExternally
        set(value) = definedExternally
    var origin: String?
        get() = definedExternally
        set(value) = definedExternally
    var lastEventId: String?
        get() = definedExternally
        set(value) = definedExternally
    var source: UnionMessagePortOrServiceWorker?
        get() = definedExternally
        set(value) = definedExternally
    var ports: Array<MessagePort>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["data"] = data
    o["origin"] = origin
    o["lastEventId"] = lastEventId
    o["source"] = source
    o["ports"] = ports
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 */
public external abstract class ServiceWorkerGlobalScope : WorkerGlobalScope {
    val clients: Clients
    val registration: ServiceWorkerRegistration
    var oninstall: ((Event) -> dynamic)?
    var onactivate: ((Event) -> dynamic)?
    var onfetch: ((FetchEvent) -> dynamic)?
    var onforeignfetch: ((Event) -> dynamic)?
    var onmessage: ((MessageEvent) -> dynamic)?
    var onnotificationclick: ((NotificationEvent) -> dynamic)?
    var onnotificationclose: ((NotificationEvent) -> dynamic)?
    var onfunctionalevent: ((Event) -> dynamic)?
    fun skipWaiting(): Promise<Unit>
}

/**
 * Exposes the JavaScript
 */
public external abstract class Client : UnionClientOrMessagePortOrServiceWorker {
    val url: String
    val frameType: FrameType
    val id: String
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}

/**
 * Exposes the JavaScript
 */
public external abstract class WindowClient : Client {
    val visibilityState: dynamic
    var focused: Boolean
    fun focus(): Promise<WindowClient>
    fun navigate(url: String): Promise<WindowClient>
}

/**
 * Exposes the JavaScript
 */
public external abstract class DNACPlatformAntelope
Clients

```kotlin
public external abstract class Clients {
    fun get(id: String): Promise<Any?
    fun matchAll(options: ClientQueryOptions = definedExternally): Promise<Array<Client>>
    fun openWindow(url: String): Promise<WindowClient?

    fun get(id: String): Promise<Any?
    fun matchAll(options: ClientQueryOptions = definedExternally): Promise<Array<Client>>
    fun openWindow(url: String): Promise<WindowClient?
    fun claim(): Promise<Unit>
}

public external interface ClientQueryOptions {
    var includeUncontrolled: Boolean? /* = false */
    var type: ClientType? /* = ClientType.WINDOW */
}

@kotlin.internal.InlineOnly
public inline fun ClientQueryOptions(includeUncontrolled: Boolean? = false, type: ClientType? = ClientType.WINDOW): ClientQueryOptions {
    val o = js("({})")
    o["includeUncontrolled"] = includeUncontrolled
    o["type"] = type
    return o

    fun waitUntil(f: Promise<Any?>)

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface ExtendableEventInit : EventInit

    @kotlin.internal.InlineOnly
        val o = js("({})")
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }

    public external open class InstallEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : ExtendableEvent {
        fun registerForeignFetch(options: ForeignFetchOptions)
    }

    public external interface ForeignFetchOptions {
        var scopes: Array<String>?
        var origins: Array<String>?
    }

    @kotlin.internal.InlineOnly
    public inline fun ForeignFetchOptions(scopes: Array<String>?, origins: Array<String>?): ForeignFetchOptions {
        val o = js("({})")
        o["scopes"] = scopes
        o["origins"] = origins
        return o
    }

    public external open class ForeignFetchEvent(type: String, eventInitDict: ForeignFetchEventInit) : ExtendableEvent {
        fun respondWith(r: Promise<ForeignFetchResponse>)
    }

    public external interface ForeignFetchEventInit : ExtendableEventInit {
        var request: Request?
        var origin: String
    }

    @kotlin.internal.InlineOnly
        val o = js("({})")
        o["request"] = request
        o["origin"] = origin
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }
```

Open Source Used In DNAC Platform Antelope 8281
```kotlin
public external interface ForeignFetchResponse {
    var response: Response?
        get() = definedExternally
    var origin: String?
        get() = definedExternally
    var headers: Array<String>?
        get() = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["response"] = response
    o["origin"] = origin
    o["headers"] = headers
    return o
}

/**
 * Exposes the JavaScript
 * [ExtendableMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableMessageEvent) to Kotlin
 */
public external open class ExtendableMessageEvent(type: String, eventInitDict: ExtendableMessageEventInit = definedExternally) : ExtendableEvent {
    open val data: Any?
    open val origin: String
    open val lastEventId: String
    open val source: UnionClientOrMessagePortOrServiceWorker?
    open val ports: Array<out MessagePort>?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface ExtendableMessageEventInit : ExtendableEventInit {
    var data: Any?
        get() = definedExternally
    var origin: String?
        get() = definedExternally
        set(value) = definedExternally
    var lastEventId: String?
        get() = definedExternally
        set(value) = definedExternally
    var source: UnionClientOrMessagePortOrServiceWorker?
        get() = definedExternally
    var ports: Array<MessagePort>?
        get() = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["data"] = data
    o["origin"] = origin
    o["lastEventId"] = lastEventId
    o["source"] = source
    o["ports"] = ports
    o["bubbles"] = bubbles
    o["Cancelable"] = cancelable
    return o
}

/**
 * Exposes the JavaScript
 * [Cache](https://developer.mozilla.org/en/docs/Web/API/Cache) to Kotlin
 */
public external abstract class Cache {
    fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?>
    fun matchAll(request: dynamic = undefined, options: CacheQueryOptions = definedExternally): Promise<Array<Response>>
    fun add(request: dynamic): Promise<Unit>
    fun addAll(requests: Array<dynamic>): Promise<Unit>
    fun put(request: dynamic, response: Response): Promise<Unit>
    fun delete(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Boolean>
    fun keys(request: dynamic = undefined, options: CacheQueryOptions = definedExternally): Promise<Array<Request>>
}

public external interface CacheQueryOptions {
    var ignoreSearch: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var ignoreMethod: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var ignoreVary: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var cacheName: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["ignoreSearch"] = ignoreSearch
    o["ignoreMethod"] = ignoreMethod
    o["ignoreVary"] = ignoreVary
    o["cacheName"] = cacheName
    return o
}

public external interface CacheBatchOperation {
    var type: String?
        get() = definedExternally
        set(value) = definedExternally
    var request: Request?
        get() = definedExternally
        set(value) = definedExternally
    var response: Response?
        get() = definedExternally
        set(value) = definedExternally
    var options: CacheQueryOptions?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["type"] = type
    o["request"] = request
    o["response"] = response
    o["options"] = options
    return o
}
```

---

Open Source Used In DNAC Platform Antelope 8282
[CacheStorage](https://developer.mozilla.org/en/docs/Web/API/CacheStorage) to Kotlin

```kotlin
abstract class CacheStorage {
    fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?>
    fun has(cacheName: String): Promise<Boolean>
    fun open(cacheName: String): Promise<Cache>
    fun delete(cacheName: String): Promise<Boolean>
    fun keys(): Promise<Array<String>>
}

open class FunctionalEvent : ExtendableEvent { companion object {
    val NONE: Short
    val CAPTURING_PHASE: Short
    val AT_TARGET: Short
    val BUBBLING_PHASE: Short
}

open interface UnionMessagePortOrServiceWorker

external interface UnionClientOrMessagePortOrServiceWorker

@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
external interface ServiceWorkerState {
    companion object
}

inline val ServiceWorkerState.Companion.INSTALLING: ServiceWorkerState get() = "installing".asDynamic().unsafeCast<ServiceWorkerState>()
inline val ServiceWorkerState.Companion.INSTALLED: ServiceWorkerState get() = "installed".asDynamic().unsafeCast<ServiceWorkerState>()
inline val ServiceWorkerState.Companion.ACTIVATING: ServiceWorkerState get() = "activating".asDynamic().unsafeCast<ServiceWorkerState>()
inline val ServiceWorkerState.Companion.ACTIVATED: ServiceWorkerState get() = "activated".asDynamic().unsafeCast<ServiceWorkerState>()
inline val ServiceWorkerState.Companion.REDUNDANT: ServiceWorkerState get() = "redundant".asDynamic().unsafeCast<ServiceWorkerState>()

@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
external interface FrameType {
}

inline val FrameType.Companion.AUXILIARY: FrameType get() = "auxiliary".asDynamic().unsafeCast<FrameType>()
inline val FrameType.Companion.TOP_LEVEL: FrameType get() = "top-level".asDynamic().unsafeCast<FrameType>()
inline val FrameType.Companion.NESTED: FrameType get() = "nested".asDynamic().unsafeCast<FrameType>()
inline val FrameType.Companion.NONE: FrameType get() = "none".asDynamic().unsafeCast<FrameType>()

@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
external interface ClientType {
}

inline val ClientType.Companion.WINDOW: ClientType get() = "window".asDynamic().unsafeCast<ClientType>()
inline val ClientType.Companion.WORKER: ClientType get() = "worker".asDynamic().unsafeCast<ClientType>()
inline val ClientType.Companion.SHAREDWORKER: ClientType get() = "sharedworker".asDynamic().unsafeCast<ClientType>()
inline val ClientType.Companion.ALL: ClientType get() = "all".asDynamic().unsafeCast<ClientType>()

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.xhr

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*

/**
 * Exposes the
 * [XMLHttpRequestEventTarget](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequestEventTarget) to
 * Kotlin
 */
abstract class XMLHttpRequestEventTarget : EventTarget {
    var onloadstart: ((ProgressEvent) -> dynamic)?
    var onprogress: ((ProgressEvent) -> dynamic)?
    var onabort: ((Event) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    var onload: ((Event) -> dynamic)?
    var ontimeout: ((Event) -> dynamic)?
    var onloadend: ((Event) -> dynamic)?
}
```

Open Source Used In DNAC Platform Antelope 8283
Open Source Used In DNAC Platform Antelope

[XMLHttpRequest](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequest) to Kotlin

```kotlin
public external open class XMLHttpRequest : XMLHttpRequestEventTarget {
    var onreadystatechange: ((Event) -> dynamic)?
    open val readyState: Short
    var timeout: Int
    var withCredentials: Boolean
    open val upload: XMLHttpRequestUpload
    open val responseURL: String
    open val status: Short
    open val statusText: String
    var responseType: XMLHttpRequestResponseType
    open val response: Any?
    open val responseText: String
    open val responseXML: Document?

    fun open(method: String, url: String)
    fun setRequestHeader(name: String, value: String)
    fun send(body: dynamic = definedExternally)
    fun abort()
    fun getResponseHeader(name: String): String?
    fun getAllResponseHeaders(): String
    fun overrideMimeType(mime: String)

    companion object {
        val UNSENT: Short
        val OPENED: Short
        val HEADERS_RECEIVED: Short
    }
}
```

[FormData](https://developer.mozilla.org/en/docs/Web/API/FormData) to Kotlin

```kotlin
public external open class FormData(form: HTMLFormElement = definedExternally) {
    fun append(name: String, value: String)
    fun append(name: String, value: Blob, filename: String = definedExternally)
    fun delete(name: String)
    fun get(name: String): dynamic
    fun getAll(name: String): Array<dynamic>
    fun has(name: String): Boolean
    fun set(name: String, value: String)
    fun set(name: String, value: Blob, filename: String = definedExternally)
}
```

[ProgressEvent](https://developer.mozilla.org/en/docs/Web/API/ProgressEvent) to Kotlin

```kotlin
public external open class ProgressEvent(type: String, eventInitDict: ProgressEventInit = definedExternally) : Event {
    open val lengthComputable: Boolean
    open val loaded: Number
    open val total: Number

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}
```

```kotlin
public external interface ProgressEventInit : EventInit {
    var lengthComputable: Boolean? /* = false */
    var loaded: Number? /* = 0 */
    var total: Number? /* = 0 */
}
```

```kotlin
@kotlin.internal.InlineOnly
    val o = js("({})")
    o["lengthComputable"] = lengthComputable
    o["loaded"] = loaded
    o["total"] = total
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}
```

[XMLHttpRequestResponseType](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequestResponseType) to Kotlin

```kotlin
public inline val XMLHttpRequestResponseType.Companion.EMPTY: XMLHttpRequestResponseType get() = "".asDynamic().unsafeCast<XMLHttpRequestResponseType>()
public inline val XMLHttpRequestResponseType.Companion.ARRAYBUFFER: XMLHttpRequestResponseType get() = "arraybuffer".asDynamic().unsafeCast<XMLHttpRequestResponseType>()
public inline val XMLHttpRequestResponseType.Companion.BLOB: XMLHttpRequestResponseType get() = "blob".asDynamic().unsafeCast<XMLHttpRequestResponseType>()
public inline val XMLHttpRequestResponseType.Companion.DOCUMENT: XMLHttpRequestResponseType get() = "document".asDynamic().unsafeCast<XMLHttpRequestResponseType>()
public inline val XMLHttpRequestResponseType.Companion.JSON: XMLHttpRequestResponseType get() = "json".asDynamic().unsafeCast<XMLHttpRequestResponseType>()
public inline val XMLHttpRequestResponseType.Companion.TEXT: XMLHttpRequestResponseType get() = "text".asDynamic().unsafeCast<XMLHttpRequestResponseType>()
```

*/

---

Open Source Used In DNAC Platform Antelope 8284

[FormData](https://developer.mozilla.org/en/docs/Web/API/FormData) to Kotlin

```kotlin
[FormData](https://developer.mozilla.org/en/docs/Web/API/FormData) to Kotlin

```
kotlin.annotation.AnnotationTarget.*

import kotlin.internal.RequireKotlin

/**
 * Signals that the annotated annotation class is a marker of an experimental API.
 * Any declaration annotated with that marker is considered an experimental declaration
 * and its call sites should accept the experimental aspect of it either by using
 * [UseExperimental], or by being annotated with that marker themselves, effectively causing further
 * propagation of that experimental aspect.
 * This class is deprecated in favor of a more general approach provided by [RequiresOptIn]/[OptIn].
 */
@Target(ANNOTATION_CLASS)
@Retention(BINARY)
@SinceKotlin("1.2")
@RequireKotlinVersionKind.COMPLIER_VERSION
@Deprecated("Please use RequiresOptIn instead.", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)
@Experimental("Please use $OptIn instead.", ReplaceWith("$OptIn(*markerClass)", "kotlin.OptIn"))
public annotation class Experimental(
    vararg val markerClass: KClass<out Annotation>
)

@Target(CLASS, PROPERTY, CONSTRUCTOR, FUNCTION,
    PROPERTY_GETTER, PROPERTY_SETTER, EXPRESSION, FILE,
    TYPEALIAS)
@Retention(SOURCE)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@MustBeDocumented
@Retention(AnnotationRetention.BINARY)
@Target(Target
    CLASS,
    ANNOTATION_CLASS,
    PROPERTY,
    FIELD,
    LOCAL_VARIABLE,
    VALUE_PARAMETER,
    CONSTRUCTOR,
    FUNCTION,
    PROPERTY_GETTER,
    PROPERTY_SETTER,
    TYPEALIAS)
@MustBeDocumented
@Retention(AnnotationRetention.BINARY)
@Target(Target
    CLASS,
    ANNOTATION_CLASS,
    PROPERTY,
    FIELD,
    LOCAL_VARIABLE,
    VALUE_PARAMETER,
    CONSTRUCTOR,
    PROPERTY_GETTER,
    PROPERTY_SETTER,
    TYPEALIAS)

Open Source Used In DNAC Platform Antelope 8285
with `@BuilderInference`.

Example: we declare
```kotlin
fun <T> sequence(@BuilderInference block: suspend SequenceScope<T>(). -> Unit): Sequence<T>
```
and use it like
```kotlin
val result = sequence {
  yield("result")
}
```
Here the type argument of the resulting sequence is inferred to `String` from the argument of the `SequenceScope.yield` function, that is called inside the lambda passed to `sequence`.

Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.

```kotlin
@Target(VALUE_PARAMETER, FUNCTION, PROPERTY)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.3")
@ExperimentalTypeInference
public annotation class BuilderInference
```

Any usage of a declaration annotated with `@ExperimentalMultiplatform` must be accepted either by
```kotlin
@OptIn(ExperimentalMultiplatform::class)
```
or by using the compiler argument `-Xopt-in=kotlin.ExperimentalMultiplatform`.

```kotlin
@Suppress("DEPRECATION")
@Experimental
@RequiresOptIn
@MustBeDocumented
@Target(
  CLASS,
  ANNOTATION_CLASS,
  PROPERTY,
  FIELD,
  LOCAL_VARIABLE,
  VALUE_PARAMETER,
  CONSTRUCTOR,
  FUNCTION,
  PROPERTY_GETTER,
  PROPERTY_SETTER,
  TYPEALIAS
)
@Retention(AnnotationRetention.BINARY)
@RequireKotlin("1.2.50", versionKind =
  RequireKotlinVersionKind.COMPILER_VERSION)
public annotation class ExperimentalMultiplatform
```

```kotlin
@Target(ANNOTATION_CLASS)
@Retention(BINARY)
@SinceKotlin("1.3")
@RequireKotlin("1.3.70", versionKind =
  RequireKotlinVersionKind.COMPILER_VERSION)
public annotation class RequiresOptIn(

  val message: String = "",
  val level: Level = Level.ERROR
) {
  /**
   * Severity of the diagnostic that should be reported on usages of API without an explicit opt-in, or empty string for the default message.
   * The default message is: "This declaration is experimental and its usage should be marked with 'Marker' or '@OptIn(Marker::class)'", where 'Marker' is the opt-in requirement marker.
   * @property level specifies how usages of API without an explicit opt-in are reported in code.
   */

  @Override
  @Retention(AnnotationRetention.BINARY)
  @Target(ANNOTATION_CLASS)
  @Suppress("DEPRECATION")
  public annotation class OptionalExpectation
```
annotated with the corresponding marker annotation.

```java
public enum class Level {
    WARNING, ERROR
}
```

```java
@Target(
    CLASS, PROPERTY,
    LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER,
    PROPERTY_SETTER, EXPRESSION, FILE,
    TYPEALIAS
)
@Retention(SOURCE)
@SinceKotlin("1.1.1")
@RequireKotlinVersionKind.COMPILE_VERSION
public annotation class OptIn(
    vararg val markerClass: KClass<out Annotation>
)
```

```java
@SinceKotlin("1.1")
public abstract class AbstractCollection<out E> protected constructor() : Collection<E> {
    abstract override val size: Int
    abstract override fun iterator(): Iterator<E>

    override fun contains(element: @UnsafeVariance E): Boolean = any { it == element }

    override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean =
        elements.all { contains(it) }

    override fun toString(): String = joinToString(", ", "[", "]") {
        if (it === this) "(this Collection)" else it.toString()
    }

    /**
     * Returns new array of type `Array<Any?>` with
     * the elements of this collection.
     */
    @JsName("toArray")
    protected open fun toArray(): Array<Any?> = copyToArrayImpl(this)

    /**
     * Fills the provided array or creates new array of the same type
     * and
     * fills it with the elements of this collection.
     * @param array the array to fill
     * @param newElement the new element
     * @return the filled array
     */
    protected open fun <T> toArray(array: Array<T>, newElement: T): Array<T> =
        copyToArrayImpl(this, array)
}
```

```java
protected enum class State {
    Ready, NotReady, Done, Failed
}
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```

```java
@SinceKotlin("1.3.70")
@RequireKotlin("1.3.70")
private var markerClass: KClass<out Annotation>? = null
```
fun containsValue(value: @UnsafeVariance V): Boolean = entries.any { it.value == value }

internal fun

override fun containsKey(key: K): Boolean {
    return implFindEntry(key) != null
}

override

@SinceKotlin("1.1")
public abstract class AbstractMap<K, out V> protected constructor() : Map<K, V>

map is invariant on its key type.
 * @param V the type of map values. The map is covariant on its value type.

[entries] property, which should return read-only set of map entries.
 *
 * @param K the type of map keys. The
definition is: In

abstract override fun get(index: Int): E

override fun iterator(): Iterator<E> = IteratorImpl()

override fun subList(fromIndex: Int, toIndex: Int): List<E> = SubList(this, fromIndex, toIndex)

RandomAccess

private var _size: Int = 0

init {

checkRangeIndexes(fromIndex, toIndex, list.size)

this._size = toIndex - fromIndex

override fun get(index: Int): E {

val otherIterator = other.iterator()

for (elem in c) {

val elemOther = otherIterator.next()

if (elem != elemOther) {
                return false
            }
        }
        return true
    }

internal fun

fun orderedHashCode(c: Collection<*>) { Int {

hashCode = 31 * hashCode + (e?.hashCode() ?: 0)

}

internal fun checkElementIndex(index: Int, size: Int) {

if (index < 0 || index >= size) {
                throw IndexOutOfBoundsException("index: $index, size: $size")

}

checkPositionIndex(index: Int, size: Int) {

if (index < 0 || index >= size) {
                throw IndexOutOfBoundsException("index: $index, size: $size")

}

checkRangeIndexes(fromIndex: Int, toIndex: Int, size: Int) {

if (fromIndex < 0 || toIndex > size) {
                throw IndexOutOfBoundsException("fromIndex: $fromIndex, toIndex: $toIndex, size: $size")

}

checkBoundsIndexes(startIndex: Int, endIndex: Int, size: Int) {

if (startIndex < 0 || endIndex > size) {
                throw IndexOutOfBoundsException("startIndex: $startIndex, endIndex: $endIndex, size: $size")

}

checkPositionIndex(fromIndex: Int, toIndex: Int, size: Int) {

if (fromIndex > toIndex) {
                throw IllegalArgumentException("fromIndex: $fromIndex > toIndex: $toIndex")

}

checkRangeIndexes(fromIndex: Int, toIndex: Int, size: Int) {

if (startIndex < 0 || endIndex > size) {
                throw IndexOutOfBoundsException("startIndex: $startIndex, endIndex: $endIndex")

}

internal fun orderedHashCode(c: Collection<*>) {

if (!hasNext()) throw NoSuchElementException()
            return get(index++)
        }

override fun previousIndex(): Int = index - 1

override fun previous(): E {
            if (!hasPrevious()) throw NoSuchElementException()
            return get(--index)
        }

override fun nextIndex(): Int = index

override fun hasPrevious(): Boolean = index > 0

override fun next(): E {
            if (!hasNext()) throw NoSuchElementException()
            return get(index++)
        }

override fun hasNext(): Boolean = index < size

override fun listIterator(): ListIterator<E> = ListIteratorImpl(0)

override fun listIterator(index: Int): ListIterator<E> = ListIteratorImpl(index)

@UnsafeVariance E): Int = indexOfLast { it == element }

override fun lastIndexOf(element: @UnsafeVariance E): Int = indexOfFirst { it == element }

override fun orderInMap(): Boolean {

return orderedEquals(this, other)
    }

/**
 * Returns the hash code value for
 * @return true, if 

containsValue(Any?): Boolean {

if (other !== this) return false

return orderedEquals(this, other) &&

* * *

* Returns the hash code value for
this list.

* * *

* * *

* * *


private open inner class IteratorImpl {

/** the index of the item that will be returned on the next call to [next]`()` */

protected

* * *

* * *

* * *

Implementation of [AbstractMap]

private open inner class ListIteratorImpl(index: Int) :

ListIterator<E> {

*/

private val kotlin.collections.

open class AbstractMap<K, V> : Map<K, V>

abstract override val size: Int

abstract override

protected constructor() : AbstractCollection<E>(), List<E> {
    abstract override val size: Int
    abstract override

override

module

object {
    internal fun unorderedHashCode(c: Collection<*>) : Int {
        var hashcode = 0
        for (element in c) {
            hashcode += (element?.hashCode() ?: 0)
        }
        return hashcode
    }

    internal fun setEquals(c: Set<*>, other: Set<*>) : Boolean {
        if (c.size != other.size) return false
        return c.containsAll(other) || c.size == other.size
    }
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found
 * in the license/LICENSE.txt file. */

package kotlin.collections

import kotlin.native.concurrent.SharedImmutable

@SharedImmutable
private val emptyElementData = emptyArray<Any?>()
private const val maxArraySize = Int.MAX_VALUE - 8
private const val defaultMinCapacity = 10

/**
 * Resizable-array implementation of the deque data structure.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public class ArrayDeque<E> : AbstractMutableList<E> {
    private var head: Int = 0
    private var elementData: Array<Any?>

    override var size: Int = 0
        private set

    /**
     * Constructs an empty deque with specified [initialCapacity], or throws [IllegalArgumentException] if [initialCapacity] is negative.
     */
    public constructor(initialCapacity: Int) {
        if (initialCapacity < 0) throw IllegalArgumentException("Illegal Capacity: $initialCapacity")
        if (initialCapacity > maxArraySize)
            throw IllegalArgumentException("ArrayDeque is too big.")
        elementData = arrayOfNulls(initialCapacity.coerceAtLeast(defaultMinCapacity))
    }

    /**
     * Constructs a deque that contains the same elements as the specified [elements] collection in the same order.
     */
    public constructor(elements: Collection<E>) {
        elementData = elements.toTypedArray()
        size = elementData.size
        if (elementData.isEmpty())
            elementData = emptyElementData
    }

    /**
     * Ensures that the capacity of this deque is at least equal to the specified [minCapacity].
     * If the current capacity is less than the [minCapacity], a new backing storage is allocated with greater capacity.
     * Otherwise, this method takes no action and simply returns.
     */
    private fun ensureCapacity(minCapacity: Int) {
        if (minCapacity < 0) throw IllegalStateException("Deque is too big.")
        if (minCapacity <= elementData.size)
            return
        if (elementData === emptyElementData) {
            elementData = arrayOfNulls(minCapacity.coerceAtLeast(defaultMinCapacity))
            return
        }

        val newCapacity = newCapacity(elementData.size, minCapacity)
        copyElements(newCapacity)
    }

    // made internal for testing
    internal fun newCapacity(oldCapacity: Int, minCapacity: Int): Int {
        var newCapacity = oldCapacity + (oldCapacity shr 1)
        if (newCapacity - minCapacity < 0)
            newCapacity = minCapacity
        if (newCapacity - maxArraySize > 0)
            newCapacity = if (minCapacity > maxArraySize) Int.MAX_VALUE else maxArraySize
        return newCapacity
    }

    /**
     * Creates a new array with the specified [newCapacity] size and copies elements in the [elementData] array to it.
     */
    private fun copyElements(newCapacity: Int) {
        val newElements = arrayOfNulls<Any?>(newCapacity)
        elementData.copyInto(newElements, 0, head, elementData.size)
        elementData.copyInto(newElements, elementData.size - head, 0, head)
        head = 0
        elementData = newElements
    }

    @kotlin.internal.InlineOnly
    private inline fun internalGet(internalIndex: Int): E {
        @Suppress("UNCHECKED_CAST")
        return elementData[internalIndex] as E
    }

    private fun positiveMod(index: Int): Int = if (index >= elementData.size) index - elementData.size else index
    private fun negativeMod(index: Int): Int = if (index < 0) index + elementData.size else index

    @kotlin.internal.InlineOnly
    private fun internalIndex(index: Int): Int = positiveMod(index) + negativeMod(index)

    @kotlin.internal.InlineOnly
    private fun positiveMod(head: Int) : Int = head + elementData.size

    @kotlin.internal.InlineOnly
    private fun negativeMod(index: Int): Int = index + elementData.size

    /**
     * Returns the first element, or throws [NoSuchElementException] if this deque is empty.
     */
    override fun first(): E = if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")

    /**
     * Returns the first element, or `null` if this deque is empty.
     */
    override fun firstOrNull(): E? = if (isEmpty()) null

    /**
     * Returns the last element, or throws [NoSuchElementException] if this deque is empty.
     */
    override fun last(): E = if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")

    /**
     * Returns the first element, or `null` if this deque is empty.
     */
    override fun lastOrNull(): E? = if (isEmpty()) null

    /**
     * Returns the last element, or throws [NoSuchElementException] if this deque is empty.
     */
    override fun lastOrNull(): E? = if (isEmpty()) null
    }

    /**
     * Ensures that the capacity of this deque is at least equal to the specified [minCapacity].
     * If the current capacity is less than the [minCapacity], a new backing storage is allocated with greater capacity.
     * Otherwise, this method takes no action and simply returns.
     */
    private fun ensureCapacity(minCapacity: Int) {
        if (minCapacity < 0) throw IllegalStateException("Deque is too big.")
        if (minCapacity <= elementData.size)
            return
        if (elementData === emptyElementData) {
            elementData = arrayOfNulls(minCapacity.coerceAtLeast(defaultMinCapacity))
            return
        }

        val newCapacity = newCapacity(elementData.size, minCapacity)
        copyElements(newCapacity)
    }

    /**
     * Ensures that the capacity of this deque is at least equal to the specified [minCapacity].
     * If the current capacity is less than the [minCapacity], a new backing storage is allocated with greater capacity.
     * Otherwise, this method takes no action and simply returns.
     */
    private fun ensureCapacity(minCapacity: Int) {
        if (minCapacity < 0) throw IllegalStateException("Deque is too big.")
        if (minCapacity <= elementData.size)
            return
        if (elementData === emptyElementData) {
            elementData = arrayOfNulls(minCapacity.coerceAtLeast(defaultMinCapacity))
            return
        }

        val newCapacity = newCapacity(elementData.size, minCapacity)
        copyElements(newCapacity)
    }

    @kotlin.internal.InlineOnly
    private inline fun internalGet(internalIndex: Int): E {
        @Suppress("UNCHECKED_CAST")
        return elementData[internalIndex] as E
    }

    private fun positiveMod(index: Int): Int = if (index >= elementData.size) index - elementData.size else index
    private fun negativeMod(index: Int): Int = if (index < 0) index + elementData.size else index

    @kotlin.internal.InlineOnly
    private inline fun internalIndex(index: Int): Int = positiveMod(head) + negativeMod(index)

    @kotlin.internal.InlineOnly
    private fun positiveMod(head: Int) : Int = head + elementData.size

    @kotlin.internal.InlineOnly
    private fun negativeMod(index: Int): Int = index + elementData.size

    /**
     * Returns the first element, or throws [NoSuchElementException] if this deque is empty.
     */
    override fun first(): E = if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")

    /**
     * Returns the first element, or `null` if this deque is empty.
     */
    override fun firstOrNull(): E? = if (isEmpty()) null

    /**
     * Returns the last element, or throws [NoSuchElementException] if this deque is empty.
     */
    override fun last(): E = if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")

    /**
     * Returns the last element, or throws [NoSuchElementException] if this deque is empty.
     */
    override fun lastOrNull(): E? = if (isEmpty()) null

    /**
     * Returns the last element, or throws [NoSuchElementException] if this deque is empty.
     */
    override fun lastOrNull(): E? = if (isEmpty()) null
}
import kotlin.collections.*

public fun <T> Array<out T>.flatten(): List<T> {
    val result = ArrayList<T>(sumBy { it.size })
    for (element in this) {
        result.addAll(element)
    }
    return result
}

public fun <T, R> Array<out Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {
    val listT = ArrayList<T>(size)
    val listR = ArrayList<R>(size)
    for (pair in this) {
        listT.add(pair.first)
        listR.add(pair.second)
    }
    return listT to listR
}

public inline fun Array<*>?.isNullOrEmpty(): Boolean {
    contract {
        returns(false) implies (this@isNullOrEmpty != null)
    }

    return this == null || this.isEmpty()
}

public inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Array<*>, C : R =
    if (isEmpty())
        defaultValue()
    else
        this

@UseExperimental(ExperimentalUnsignedTypes::class)
@SinceKotlin("1.3")
@PublishedApi
@kotlin.jvm.JvmName("contentDeepEquals")
@kotlin.js.JsName("contentDeepEqualsImpl")
internal fun <T> Array<out T>.contentDeepEqualsImpl(other: Array<out T>): Boolean {
    if (this === other) return true
    if (this.size != other.size) return false

    for (i in indices) {
        val v1 = this[i]
        val v2 = other[i]

        if (v1 === v2) {
            continue
        } else if (v1 == null || v2 == null) {
            return false
        }

        when {
            v1 is Array<*> && v2 is Array<*> -> if (!v1.contentEquals(v2)) return false
            v1 is ByteArray && v2 is ByteArray -> if (!v1.contentEquals(v2)) return false
            v1 is ShortArray && v2 is ShortArray -> if (!v1.contentEquals(v2)) return false
            v1 is IntArray && v2 is IntArray -> if (!v1.contentEquals(v2)) return false
            v1 is LongArray && v2 is LongArray -> if (!v1.contentEquals(v2)) return false
            v1 is FloatArray && v2 is FloatArray -> if (!v1.contentEquals(v2)) return false
            v1 is DoubleArray && v2 is DoubleArray -> if (!v1.contentEquals(v2)) return false
            v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false
            v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false
            v1 is UByteArray && v2 is UByteArray -> if (!v1.contentEquals(v2)) return false
            v1 is UShortArray && v2 is UShortArray -> if (!v1.contentEquals(v2)) return false
            v1 is UIntArray && v2 is UIntArray -> if (!v1.contentEquals(v2)) return false
            v1 is ULongArray && v2 is ULongArray -> if (!v1.contentEquals(v2)) return false
            else -> if (v1 != v2) return false
        }
    }
    return true
}
result.append(element.contentToString())

is BooleanArray -> result.append(element.contentToString())

is UByteArray -> result.append(element.contentToString())

is UShortArray ->
result.append(element.contentToString())

is UIntArray ->
result.append(element.contentToString())

is ULongArray ->
result.append(element.contentToString())

else ->
result.append(element.toString())
    }
}

result.append(']')

processed.removeAt(processed.lastIndex)
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.collections

/**
 * Data class representing a value from a collection or sequence, along with its index in that collection or sequence.
 * * @property value the underlying value.
 * * @property index the index of the value in the collection or sequence.
 * */
public data class IndexedValue<out T>(
    public val index: Int,
    public val value: T)

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmName("MapAccessorsKt")

package kotlin.collections

import kotlin.reflect.KProperty
import kotlin.internal.Exact

/**
 * Returns the value of the property for the given object from this read-only map.
 * * @param thisRef the object for which the value is requested (not used).
 * * @param property the metadata for the property, used to get the name of property and lookup the value corresponding to this name in the map.
 * * @return the property value.
 * * @throws NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit default (see [withDefault]).
 * */
@kotlin.internal.InlineOnly
public inline operator fun <V, V1 : V> Map<in String, @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =
    @Suppress("UNCHECKED_CAST") (getOrImplicitDefault(property.name) as V1)

/**
 * Returns the value of the property for the given object from this mutable map.
 * * @param thisRef the object for which the value is requested (not used).
 * * @param property the metadata for the property, used to get the name of property and lookup the value corresponding to this name in the map.
 * * @return the property value.
 * * @throws NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit default (see [withDefault]).
 * */
@kotlin.jvm.JvmName("getVar")
@kotlin.internal.InlineOnly
public inline operator fun <V, V1 : V> MutableMap<in String, out @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =
    @Suppress("UNCHECKED_CAST") (getOrImplicitDefault(property.name) as V1)

@Deprecated("Use getValue() with two type parameters instead", level = DeprecationLevel.ERROR)
@kotlin.jvm.JvmName("getVarContravariant")
@kotlin.internal.LowPriorityInOverloadResolution
@kotlin.jvm.JvmName("MapsKt")
@PublishedApi
internal fun <K, V> Map<in K, @Exact V>.getValue(thisRef: Any?, property: KProperty<*>, value: V): V =
    @Suppress("UNCHECKED_CAST") (this.get(thisRef, property, value) as V)

/**
 * Stores the value of the property for the given object in this mutable map.
 * * @param thisRef the object for which the value is requested (not used).
 * * @param property the metadata for the property, used to get the name of property and store the value associated with that name in the map.
 * * @param value the value to set.
 */
@kotlin.internal.InlineOnly
public inline operator fun <V> MutableMap<in String, in V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {
    this.put(property.name, value)
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("MapsKt")

package kotlin.collections

/**
 * Returns the value for the given key, or the implicit default value for this map.
 * By default no implicit value is provided for maps and a [NoSuchElementException] is thrown.
 * To create a map with implicit default value use [withDefault] method.
 * * @throws NoSuchElementException when the map doesn't contain a value for the specified key and no implicit default was provided for that map.
 * */
@kotlin.jvm.JvmName("getOrImplicitDefaultNullable")
@PublishedApi
internal fun <K, V> Map<K, V>.getOrImplicitDefaultNullable(key: K): V {
    if (this is MapWithDefault) return this.getOrImplicitDefault(key)
    return getOrElseNullable(key, { throw NoSuchElementException("Key.key: 

Open Source Used In DNAC Platform Antelope 8295
is missing in the map.\})\n\n\n
...
<@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.removeAll(elements: Collection<T>): Boolean =
    @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).removeAll(elements)  Add the specified [element] to this mutable collection.  Allows to
    
Retains only the elements in this collection that are contained in the specified collection.  If any element was removed from the collection, 'false' if the collection was not modified.

MutableCollection<out T>.retainAll(elements: Collection<T>): Boolean =
    @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).retainAll(elements)  Adds all elements of the given [elements] collection to this mutable collection.  Adds all elements of the given [elements] array to this mutable collection.

MutableCollection<out T>.plusAssign(elements: Array<T>): Boolean =
    @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).plusAssign(elements)  Adds all elements of the given [elements] sequence to this mutable collection.

MutableCollection<out T>.plusAssign(elements: Sequence<T>): Boolean =
    @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).plusAssign(elements)  Adds all elements of the given [elements] sequence to this mutable collection.

MutableCollection<out T>.minusAssign(element: T): Boolean =
    this.remove(element)  Removes all elements contained in the given [elements] collection to this mutable collection.

MutableCollection<out T>.minusAssign(elements: Array<T>): Boolean =
    elements.isNotEmpty() && removeAll(elements.toHashSet())  Removes all elements from this [MutableCollection] that are also contained in the given [elements] collection.

MutableCollection<out T>.minusAssign(elements: Sequence<T>): Boolean =
    elements.toHashSet().isEmpty() && removeAll(elements)  Removes all elements from this [MutableCollection] that are also contained in the given [elements] sequence.

MutableCollection<out T>.minusAssign(elements: Collection<T>): Boolean =
    removeAll(elements.convertToSetForSetOperationWith(this))  Removes all elements contained in the given [elements] collection to this mutable collection.

MutableCollection<out T>.addAll(elements: Array<out T>): Boolean =
    addAll(elements.asList())  Adds all elements of the given [elements] array to this mutable collection.

MutableCollection<out T>.addAll(elements: Sequence<T>): Boolean =
    addAll(elements)  Adds all elements of the given [elements] sequence to this mutable collection.

MutableCollection<out T>.addAll(elements: Collection<T>): Boolean =
    addAll(elements)  Adds all elements of the given [elements] collection to this mutable collection.

MutableCollection<out T>.retainAll(elements: Array<T>): Boolean =
    retainAll(elements)  Retains only elements of this [MutableCollection] that are contained in the given [elements] array and is not empty.

MutableCollection<out T>.retainAll(elements: Collection<T>): Boolean =
    retainAll(elements)  Retains only elements of this [MutableCollection] that are contained in the given [elements] collection.

MutableCollection<out T>.retainAll(elements: Sequence<T>): Boolean =
    retainAll(elements)  Retains only elements of this [MutableCollection] that are contained in the given [elements] sequence.
MutableCollection<in T>.retainAll(elements: Sequence<T>): Boolean {
    val set = elements.toHashSet()
    if (set.isNotEmpty())
        return retainAll(set)
    else
        return retainNothing()
}

private fun <T> MutableCollection<in T>.retainNothing(): Boolean {
    val result = isNotEmpty()
    clear()
    return result
}

MutableIterable<T>.removeAll(predicate: (T) -> Boolean): Boolean {
    return filterInPlace(predicate, true)
}

MutableIterable<T>.retainAll(predicate: (T) -> Boolean): Boolean {
    return filterInPlace(predicate, false)
}

MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean {
    return filterInPlace(predicate, true)
}

MutableList<T>.retainAll(predicate: (T) -> Boolean): Boolean {
    return filterInPlace(predicate, false)
}

private fun <T> MutableList<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {
    if (this !is RandomAccess)
        return (this as MutableList<T>).filterInPlace(predicate, predicateResultToRemove)

    var writeIndex: Int = 0
    for (readIndex in 0..lastIndex) {
        val element = this[readIndex]
        if (predicate(element) == predicateResultToRemove)
            continue

        if (writeIndex != readIndex)
            this[writeIndex] = element

        writeIndex++
    }
    if (writeIndex < size) {
        for (removeIndex in lastIndex downTo writeIndex)
            removeAt(removeIndex)

        return true
    } else {
        return false
    }
}

MutableList<T>.removeFirst(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else removeAt(0)

MutableList<T>.removeLast(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else removeAt(lastIndex)

MutableList<T>.removeFirstOrNull(): T? = if (isEmpty()) null else removeAt(0)

MutableList<T>.removeLastOrNull(): T? = if (isEmpty()) null else removeAt(lastIndex)

MutableList<T>.retainAll(predicate: (T) -> Boolean): Boolean {
    return filterInPlace(predicate, false)
}

private fun <T> MutableList<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {
    if (this !is RandomAccess)
        return (this as MutableList<T>).filterInPlace(predicate, predicateResultToRemove)

    var writeIndex: Int = 0
    for (readIndex in 0..lastIndex) {
        val element = this[readIndex]
        if (predicate(element) == predicateResultToRemove)
            continue

        if (writeIndex != readIndex)
            this[writeIndex] = element

        writeIndex++
    }
    if (writeIndex < size) {
        for (removeIndex in lastIndex downTo writeIndex)
            removeAt(removeIndex)

        return true
    } else {
        return false
    }
}

Randomly shuffles elements in this mutable list using the specified [random] instance as the source of randomness.
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun <T> MutableList<T>.shuffle(random: Random): List<T> = toMutableList().apply { shuffle(random) }
@SinceKotlin("1.3")
public fun <T> MutableList<T>.shuffle(random: Random): Unit {
    for (i in lastIndex
        downTo 1) {
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
    }
}

package
kotlin.collections

private open class ReversedListReadOnly<out T>(private val delegate: List<T>) :
    AbstractList<T>() {
    override val size: Int get() = delegate.size
    override fun get(index: Int): T =
        delegate[reverseElementIndex(index)]
}

private class ReversedList<T>(private val delegate: MutableList<T>) :
    AbstractMutableList<T>() {
    override val size: Int get() = delegate.size
    override fun get(index: Int): T =
        delegate[reverseElementIndex(index)]

    override fun clear() = delegate.clear()
    override fun removeAt(index: Int): T =
        delegate.removeAt(reverseElementIndex(index))
    override fun set(index: Int, element: T): T =
        delegate.set(reverseElementIndex(index), element)
    override fun add(index: Int, element: T) {
        delegate.add(reversePositionIndex(index), element)
    }
}

private fun List<*>.reverseElementIndex(index: Int) =
    if (index in 0..lastIndex) lastIndex - index else throw IndexOutOfBoundsException("Element index $index must be in range [${0..lastIndex}].")

private fun List<*>.reversePositionIndex(index: Int) =
    if (index in 0..size) size - index else throw IndexOutOfBoundsException("Position index $index must be in range [${0..size}].")

/**
 * Returns a reversed read-only view of the original List.
 * All changes made in the original list will be reflected in the reversed one.
 * @sample samples.collections.ReversedViews.asReversedList
 */
public fun <T> List<T>.asReversed(): List<T> = ReversedListReadOnly(this)

/**
 * Returns a reversed mutable view of the original mutable List.
 * All changes made in the original list will be reflected in the reversed one and vice versa.
 * @sample samples.collections.ReversedViews.asReversedMutableList
 */
@kotlin.jvm.JvmName("asReversedMutable")
public fun <T> MutableList<T>.asReversed(): MutableList<T> = ReversedList(this)

/**
 * Builds a [Sequence] lazily yielding values one by one.
 * @see kotlin.sequences.generateSequence
 * @sample samples.collections.Sequences.Building.buildSequenceYieldAll
 * @sample samples.collections.Sequences.buildFibonacciSequence
 */
@SinceKotlin("1.3")
public fun <T> sequence(@BuilderInference block: suspend SequenceScope<T>.() -> Unit): Sequence<T> = Sequence {
    iterator(block) }

@SinceKotlin("1.3")
@Deprecated("Use 'sequence { }' function instead.",
    ReplaceWith("sequence(builderAction)"), level = DeprecationLevel.ERROR)
@kotlin.internal.InlineOnly
public fun <T> sequence(@BuilderInference noinline builderAction: suspend SequenceScope<T>.() -> Unit): Sequence<T> =
    Sequence { iterator(builderAction) }

/**
 * The scope for yielding values of a [Sequence] or an [Iterator],
 * provides [yield] and [yieldAll] suspension functions.
 * @see Sequence
 * @see Iterator
 */
@SinceKotlin("1.3")
public inline fun <T> SequenceScope<T>.yieldOne(): Unit = yield(1)
@SinceKotlin("1.3")
public inline fun <T> SequenceScope<T>.yieldAll(): Unit = yieldAll()
public abstract class SequenceScope<T> {

    /**
     * Yields a value to the [Iterator] being built and suspends
     * until the next value is requested.
     */
    @sample
    public abstract suspend fun yield(value: T)

    /**
     * Yields all values from the `iterator` to the [Iterator] being built
     * and suspends until all these values are iterated and the next one is requested.
     */
    @sample
    public abstract suspend fun yieldAll(Iterator<T> iterator)

    /**
     * Yields potentially infinite sequence of values to the [Iterator] being built
     * and suspends until all these values are iterated and the next one is requested.
     */
    @sample
    public suspend fun yieldAll(elements: Iterable<T>) {
        if (elements is Collection && elements.isEmpty()) return
        return yieldAll(elements.iterator())
    }

    /**
     * Yields potentially infinite sequence of values to the [Iterator] being built
     * and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(sequence: Sequence<T>) =
        yieldAll(sequence.iterator())
}

@Deprecated("Use SequenceScope class instead.", ReplaceWith("SequenceScope<T>"), level = DeprecationLevel.ERROR)
public typealias SequenceBuilder<T> = SequenceScope<T>

private typealias State = Int

private const val State_NotReady: State = 0
private const val State_ManyNotReady: State = 1
private const val State_ManyReady: State = 2
private const val State_Ready: State = 3
private const val State_Done: State = 4
private const val State_Failed: State = 5

private class SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {
    private var state = State_NotReady
    private var nextValue: T? = null
    private var nextIterator: Iterator<T>? = null
    var nextStep: Continuation<Unit>? = null

    override fun hasNext(): Boolean {
        while (true) {
            when (state) {
                State_NotReady -> {}
                State_ManyNotReady ->
                    if (nextIterator!!.hasNext()) {
                        state = State_ManyReady
                        return true
                    } else {
                        nextIterator = null
                    }
                State_Done -> return false
                State_Ready, State_ManyReady -> return true
                else -> throw exceptionalState()
            }

            state = State_Failed
            val step = nextStep!!
            nextStep = null
            step.resume(Unit)
        }
    }

    override fun next(): T {
        when (state) {
            State_NotReady, State_ManyNotReady -> return nextNotReady()
            State_ManyReady -> {
                state = State_ManyNotReady
                return nextIterator!!.next()
            }
            State_Ready -> {
                state = State_NotReady
                @Suppress("UNCHECKED_CAST")
                val result = nextValue as T
                if (result != null) return result
                else -> throw exceptionalState()
            }
            else -> throw exceptionalState()
        }
    }

    private fun nextNotReady(): T {
        if (nextIterator?.hasNext() != null) return nextIterator!!.next()
        else return null
    }

    private fun exceptionalState(): Throwable = when (state) {
        State_Done -> NoSuchElementException()
        State_Failed ->
            IllegalStateException("Iterator has failed.")
        else ->
            IllegalStateException("Unexpected state of the iterator: $state")
    }

    override suspend fun yield(value: T) {
        nextValue = value
        state = State_Ready
        return suspendCoroutineUninterceptedOrReturn { c ->
            c.resume(Unit)
        }
    }

    override suspend fun yieldAll(iterator: Iterator<T>) {
        if (!iterator.hasNext()) return
        nextIterator = iterator
        state = State_ManyReady
        return suspendCoroutineUninterceptedOrReturn { c ->
            c.resume(Unit)
        }
    }

    // Completion continuation implementation
    override fun resumeWith(result: Result<Unit>) {
        if (result.isSuccess) return
        @Suppress("UNCHECKED_CAST")
        val result = result.cause as Throwable
        if (result is NoSuchElementException) {
            state = State_Done
            return suspendCoroutineUninterceptedOrReturn { c ->
                c.resume(Unit)
            }
        } else {
            state = State_Failed
            return suspendCoroutineUninterceptedOrReturn { c ->
                c.resume(Unit)
            }
        }
    }
}

fun removeFirst(n: Int) {
    require(n >= 0) { "n shouldn't be negative but it is $n" }
    require(n <= size) { "n shouldn't be greater than the buffer size: n = $n, size = $size" }

    val start = startIndex
    val end = start.forward(n)

    if (start > end) {
        buffer.fill(null, start, capacity)
        buffer.fill(null, 0, end)
    } else {
        buffer.fill(null, start, end)
    }
}

@Suppress("NOTHING_TO_INLINE")
private inline fun Int.forward(n: Int): Int = (this + n) % capacity

package kotlin.collections

// UByteArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(array: UByteArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]

    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(array: UByteArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// UShortArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(array: UShortArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]

    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(array: UShortArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// UIntArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(array: UIntArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]

    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(array: UIntArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// ULongArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(array: ULongArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]

    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(array: ULongArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// Interfaces
=============================================================================
/**
 * Sorts the given array using qsort algorithm.
 */
@ExperimentalUnsignedTypes
internal fun sortArray(array: UByteArray) = quickSort(array, 0, array.size - 1)
@ExperimentalUnsignedTypes
internal fun sortArray(array: UShortArray) = quickSort(array, 0, array.size - 1)
@ExperimentalUnsignedTypes
internal fun sortArray(array: UIntArray) = quickSort(array, 0, array.size - 1)
@ExperimentalUnsignedTypes
internal fun sortArray(array: ULongArray) = quickSort(array, 0, array.size - 1)
import kotlin.internal.InlineOnly

/**
 * This marker distinguishes the experimental contract declaration API and is used to opt-in for that feature when declaring contracts of user functions. Any usage of a declaration annotated with `@ExperimentalContracts` must be accepted either by `optIn` annotating that usage with the `[OptIn]` annotation, e.g. `@OptIn(ExperimentalContracts::class)` or by using the compiler argument `-Xopt-in=kotlin.contracts.ExperimentalContracts`.

@Suppress("DEPRECATION")
@Retention(AnnotationRetention.BINARY)
@Experimental
@RequiresOptIn
@MustBeDocumented
public annotation class ExperimentalContracts

/**
 * Provides a scope, where the functions of the contract DSL, such as `[returns]`, `[callsInPlace]`, etc., can be used to describe the contract of a function. This type is used as a receiver type of the lambda function passed to the `[contract]` function.

@ContractsDsl
@ExperimentalContracts
@SinceKotlin("1.3")
public interface ContractBuilder {

    /**
     * Describes a situation when a function returns normally, without any exceptions thrown.
     * Uses [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
     */
    @ContractsDsl public fun returns(): Returns

    /**
     * Describes a situation when a function returns normally with the specified return value.
     * The possible values of [value] are limited to `true`, `false` or `null`. Uses [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
     */
    // @sample samples.contracts.returnsTrueContract
    // @sample samples.contracts.returnsFalseContract
    // @sample samples.contracts.returnsNullContract
    @ContractsDsl public fun returns(value: Any?): Returns

    /**
     * Describes a situation when a function returns normally with any value that is not `null`.
     * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
     */
    // @sample samples.contracts.returnsNotNullContract
    @ContractsDsl public fun returnsNotNull(): ReturnsNotNull

    /**
     * Specifies how many times a function invokes its function parameter in place.
     *
     * See [ContractBuilder.callsInPlace] for the details of the call-in-place function contract.
     */
    @ContractsDsl
    @ExperimentalContracts
    @SinceKotlin("1.3")
    public enum class InvocationKind {
        AT_MOST_ONCE,
        AT_LEAST_ONCE,
        EXACTLY_ONCE,
        UNKNOWN
    }

    /**
     * Specifies the contract of a function.
     *
     * The contract description must be at the beginning of a function and have at least one effect.
     * Only the top-level functions can have a contract for now.
     * @param builder the lambda where the contract of a function is described with the help of the [ContractBuilder] members.
     */
    @ContractsDsl
    @ExperimentalContracts
    @SinceKotlin("1.3")
    public interface public fun contract (builder: ContractBuilder)

    @ContractsDsl
    @ExperimentalContracts
    @SinceKotlin("1.3")
    @public interface ContractBuilder
    {
        /**
         * Describes a situation when a function returns normally, without any exceptions thrown.
         * Uses [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
         */
        @ContractsDsl public fun returns(): Returns

        /**
         * Describes a situation when a function returns normally with the specified return value.
         * The possible values of [value] are limited to `true`, `false` or `null`. Uses [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
         */
        // @sample samples.contracts.returnsTrueContract
        // @sample samples.contracts.returnsFalseContract
        // @sample samples.contracts.returnsNullContract
        @ContractsDsl public fun returns(value: Any?): Returns

        /**
         * Describes a situation when a function returns normally with any value that is not `null`.
         * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
         */
        // @sample samples.contracts.returnsNotNullContract
        @ContractsDsl public fun returnsNotNull(): ReturnsNotNull

        /**
         * Specifies how many times a function invokes its function parameter in place.
         *
         * See [ContractBuilder.callsInPlace] for the details of the call-in-place function contract.
         */
        @ContractsDsl
        @ExperimentalContracts
        @SinceKotlin("1.3")
        public enum class InvocationKind {
            AT_MOST_ONCE,
            AT_LEAST_ONCE,
            EXACTLY_ONCE,
            UNKNOWN
        }

        /**
         * Specifies the contract of a function.
         *
         * The contract description must be at the beginning of a function and have at least one effect.
         * Only the top-level functions can have a contract for now.
         * @param builder the lambda where the contract of a function is described with the help of the [ContractBuilder] members.
         */
        @ContractsDsl
        @ExperimentalContracts
        @SinceKotlin("1.3")
        public interface public fun contract (builder: ContractBuilder)
samples.contracts.returnsNotNullContract
* @sample samples.contracts.callsInPlaceAtMostOnceContract
* @sample samples.contracts.callsInPlaceAtLeastOnceContract
* @sample samples.contracts.callsInPlaceExactlyOnceContract
* @sample samples.contracts.callsInPlaceUnknownContract
*
@ContractsDsl
@ExperimentalContracts
@InlineOnly
@SinceKotlin("1.3")
@Suppress("UNUSED_PARAMETER")
public inline fun contract(builder:
ContractBuilder.() -> Unit) { }

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.coroutines

/**
 * Marks coroutine context element that intercepts coroutine continuations.
 * The coroutines framework uses [ContinuationInterceptor.Key] to retrieve the interceptor and
 * intercepts all coroutine continuations with [interceptContinuation] invocations.
 * [ContinuationInterceptor] behaves like a [polymorphic element][AbstractCoroutineContextKey], meaning that
 * its implementation delegates [get][CoroutineContext.Element.get] and
 * [minusKey][CoroutineContext.Element.minusKey] to [getPolymorphicElement] and [minusPolymorphicKey] respectively.
 * [ContinuationInterceptor] subtypes can be extracted from the coroutine context using either
 * [ContinuationInterceptor.Key] or subtype key if it extends [AbstractCoroutineContextKey].
 */
@SinceKotlin("1.3")
public interface ContinuationInterceptor : CoroutineContext.Element {

    /**
     * The key that defines *the* context interceptor.
     */
    companion object Key :
        CoroutineContext.Key<ContinuationInterceptor>

    /**
     * Returns continuation that wraps the original [continuation], thus intercepting all resumptions.
     * This function is invoked by coroutines framework when needed and the resulting continuations are
     * cached internally per each instance of the original [continuation].
     */
    public fun <T>
        interceptContinuation(continuation: Continuation<T>): Continuation<T>

    /**
     * Invoked for the continuation instance returned by [interceptContinuation] when the original
     * continuation completes and will not be used anymore. This function is invoked only if [interceptContinuation]
     * had returned a different continuation instance from the one it was invoked with.
     * Default implementation does nothing.
     */
    public fun releaseInterceptedContinuation(continuation: Continuation<*>) {
        /* do nothing by default */
    }

        // getPolymorphicKey specialized for ContinuationInterceptor key
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CAST")
            return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null
        } else {
            return if (ContinuationInterceptor === key) this as E else null
        }
    }

    public override fun minusKey(key: CoroutineContext.Key<*>): CoroutineContext {
        // minusPolymorphicKey specialized for ContinuationInterceptor key
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CAST")
            return if (key.isSubKey(this.key)) EmptyCoroutineContext else this
        } else {
            return if (ContinuationInterceptor === key) EmptyCoroutineContext else this
        }
    }

    public override fun get(key: CoroutineContext.Key<*>): CoroutineContext {
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CAST")
            return if (key.isSubKey(this.key)) EmptyCoroutineContext else this
        } else {
            return if (ContinuationInterceptor === key) EmptyCoroutineContext else this
        }
    }

    public override fun get(key: CoroutineContext.Key<*>): CoroutineContext {
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CAST")
            return if (key.isSubKey(this.key)) EmptyCoroutineContext else this
        } else {
            return if (ContinuationInterceptor === key) EmptyCoroutineContext else this
        }
    }

    public override fun get(key: CoroutineContext.Key<*>): CoroutineContext {
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CAST")
            return if (key.isSubKey(this.key)) EmptyCoroutineContext else this
        } else {
            return if (ContinuationInterceptor === key) EmptyCoroutineContext else this
        }
    }

    public override fun get(key: CoroutineContext.Key<*>): CoroutineContext {
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CAST")
            return if (key.isSubKey(this.key)) EmptyCoroutineContext else this
        } else {
            return if (ContinuationInterceptor === key) EmptyCoroutineContext else this
        }
    }

    public override fun get(key: CoroutineContext.Key<*>): CoroutineContext {
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CAST")
            return if (key.isSubKey(this.key)) EmptyCoroutineContext else this
        } else {
            return if (ContinuationInterceptor === key) EmptyCoroutineContext else this
        }
    }

    public override fun get(key: CoroutineContext.Key<*>): CoroutineContext {
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CAST")
            return if (key.isSubKey(this.key)) EmptyCoroutineContext else this
        } else {
            return if (ContinuationInterceptor === key) EmptyCoroutineContext else this
        }
    }

    public override fun get(key: CoroutineContext.Key<*>): CoroutineContext {
        @OptIn(ExperimentalStdlibApi::class) @sample if (key is AbstractCoroutineContextKey<*>) {
            @Suppress("UNCHECKED_CA
Returns a context containing elements from this context and elements from other [context].

* The elements from this context with the same key as in the other one are dropped.

```kotlin
operator fun plus(context: CoroutineContext): CoroutineContext =
    if (context === EmptyCoroutineContext) this else
    context.fold(this) { acc, element ->
        val removed =
            acc.minusKey(element.key)
        if (removed === EmptyCoroutineContext) element else {
            // make sure interceptor is always last in the context (and thus is fast to get when present)
            val interceptor = removed[ContinuationInterceptor]
            if (interceptor == null) CombinedContext(removed, element) else {
                val left = removed.minusKey(ContinuationInterceptor)
                if (left === EmptyCoroutineContext) CombinedContext(element, interceptor) else
                    CombinedContext(CombinedContext(left, element), interceptor)
            }
        }
    }
```

* Returns a context containing elements from this context, but without an element with
  the specified [key].

```kotlin
public fun minusKey(key: Key<*>): CoroutineContext
```

* An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.

```kotlin
public interface Key<E : Element> {
    public val key: Key<*>

    public override operator fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)

    public override fun minusKey(key: Key<*>): CoroutineContext =
        if (this.key == key) EmptyCoroutineContext else this
```

* The elements from this context with the same key as in the other one are dropped.

```kotlin
CoroutineContext =
    if (context === EmptyCoroutineContext) element else
    context.fold(this) { acc, element ->
        val removed =
            acc.minusKey(element.key)
        if (removed === EmptyCoroutineContext) element else {
            // make sure interceptor is always last in the context (and thus is fast to get when present)
            val interceptor = removed[ContinuationInterceptor]
            if (interceptor == null) CombinedContext(removed, element) else {
                val left = removed.minusKey(ContinuationInterceptor)
                if (left === EmptyCoroutineContext) CombinedContext(element, interceptor) else
                    CombinedContext(CombinedContext(left, element), interceptor)
            }
        }
    }
```

* An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.

```kotlin
public interface Element : CoroutineContext {
    /**
     * A key of this coroutine context element.
     */
    public val key: Key<*>

    public override operator fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)

    public override fun minusKey(key: Key<*>): CoroutineContext =
        if (this.key == key) EmptyCoroutineContext else this
```

* A key of this coroutine context element.

```kotlin
public val key: Key<*>
```

* public override fun minusKey(key: Key<*>): CoroutineContext

* An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.

```kotlin
public interface Element : CoroutineContext {
    /**
     * A key of this coroutine context element.
     */
    public val key: Key<*>

    public override operator fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)

    public override fun minusKey(key: Key<*>): CoroutineContext =
        if (this.key == key) EmptyCoroutineContext else this
```

* A key of this coroutine context element.

```kotlin
public val key: Key<*>
```

* public override fun minusKey(key: Key<*>): CoroutineContext

* An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.

```kotlin
public interface Element : CoroutineContext {
    /**
     * A key of this coroutine context element.
     */
    public val key: Key<*>

    public override operator fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)

    public override fun minusKey(key: Key<*>): CoroutineContext =
        if (this.key == key) EmptyCoroutineContext else this
```

* A key of this coroutine context element.

```kotlin
public val key: Key<*>
```

* public override fun minusKey(key: Key<*>): CoroutineContext

* An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.

```kotlin
public interface Element : CoroutineContext {
    /**
     * A key of this coroutine context element.
     */
    public val key: Key<*>

    public override operator fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)

    public override fun minusKey(key: Key<*>): CoroutineContext =
        if (this.key == key) EmptyCoroutineContext else this
```

* A key of this coroutine context element.

```kotlin
public val key: Key<*>
```

* public override fun minusKey(key: Key<*>): CoroutineContext

* An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.

```kotlin
public interface Element : CoroutineContext {
    /**
     * A key of this coroutine context element.
     */
    public val key: Key<*>

    public override operator fun <E : Element> get(key: Key<E>): E? =
        if (this.key == key) this as E else null

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(initial, this)

    public override fun minusKey(key: Key<*>): CoroutineContext =
        if (this.key == key) EmptyCoroutineContext else this
```

* A key of this coroutine context element.

```kotlin
public val key: Key<*>
```

* public override fun minusKey(key: Key<*>): CoroutineContext

* An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.
the [key] is associated\n with [Element.key] via [AbstractCoroutineContextKey].\n * See
 [AbstractCoroutineContextKey] for the example of usage.
*/
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun <E : Element>
Element.getPolymorphicElement(key: Key<E>): E? {
    if (key is AbstractCoroutineContextKey<*>, *) {
        @Suppress("UNCHECKED_CAST")
        return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null
    }
    @Suppress("UNCHECKED_CAST")
    return if (this.key === key) this as E else null
}

/**
 * Returns empty coroutine context if the element is associated with the given [key] in a polymorphic manner \n * or \n * `null` otherwise.\n * This method returns empty context if either [Element.key] is equal to the given [key] or if the \n * [key] is associated\n * with [Element.key] via [AbstractCoroutineContextKey].\n */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun Element.minusPolymorphicKey(key: Key<*>): CoroutineContext {
    if (key is AbstractCoroutineContextKey<*>, *) {
        return if (key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this
    }
    return if (this.key === key) EmptyCoroutineContext else this
}

/**
 * An empty coroutine context.
 */
@SinceKotlin("1.3")
public object EmptyCoroutineContext : CoroutineContext,Serializable {
    private const val serialVersionUID: Long = 0
    private fun readResolve(): Any = EmptyCoroutineContext

    public override fun <E : Element> get(key: Key<E>): E? = null
    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R = initial
    public override fun plus(context: CoroutineContext): CoroutineContext = context
    public override fun minusKey(key: Key<*>): CoroutineContext = this
    public override fun hashCode(): Int = 0
    public override fun toString(): String = "EmptyCoroutineContext"
}

//--------------------- internal impl ---------------------

class CombinedContext(
    private val left: CoroutineContext,
    private val element: Element
) : CoroutineContext, Serializable {

    override fun <E : Element> get(key: Key<E>): E? {
        var cur = this
        while (true) {
            cur.element[key]?.let { return it }
            val next = cur.left
            if (next is CombinedContext) {
                cur = next
            } else {
                return next[key]
            }
        }
    }

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =
        operation(left.fold(initial, operation), element)

    public override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }
        val newLeft = left.minusKey(key)
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element
            else ->
                CombinedContext(newLeft, element)
        }
    }

    override fun equals(other: Any?): Boolean =
        this === other || other is CombinedContext && other.size() == size() &&
        other.containsAll(this)

    override fun hashCode(): Int = left.hashCode() + element.hashCode()

    override fun toString(): String =
        "[" + fold("") { acc, element ->
            if (acc.isEmpty()) element.toString() else
                "$acc, $element"
        } + "]"
}


---
current continuation instance inside suspend functions and either suspends
* currently running coroutine or returns
result immediately without suspension.
* If the [block] returns the special [COROUTINE_SUSPENDED]
value, it means that suspend function did suspend the execution and will
* not return any result immediately. In
this case, the [Continuation] provided to the [block] shall be
* resumed by invoking [Continuation.resumeWith] at
some moment in the
* future when the result becomes available to resume the computation.
* Otherwise, the
return value of the [block] must have a type assignable to [T] and represents the result of this suspend function.
* It means that the execution was not suspended and the [Continuation] provided to the [block] shall not be invoked.
* As the result type of the [block] is declared as `Any?` and cannot be correctly type-checked,
* its proper return
* type remains on the conscience of the suspend function's author.
* Invocation of [Continuation.resumeWith] resumes coroutine directly in the invoker's thread without going through the
* [ContinuationInterceptor] that might be present in the coroutine's [CoroutineContext].
* It is the invoker's responsibility to ensure that a proper invocation context is established.
* [Continuation.intercepted] can be used to acquire the intercepted continuation.
* Note that it is not recommended to call either [Continuation.resume] nor
* [Continuation.resumeWithException] functions synchronously
* in the same stackframe where suspension function is run. Use [suspendCoroutine] as a safer way to obtain current continuation instance.

suspendCoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T =
    throw
NotImplementedError("Implementation of suspendCoroutineUninterceptedOrReturn is intrinsic")

/**
 * This value is used as a return value of [suspendCoroutineUninterceptedOrReturn] `block` argument to state that
 * the execution was suspended and will not return any result immediately.
 * **Note: this value should not be used in general code.** Using it outside of the context of [suspendCoroutineUninterceptedOrReturn] `block` argument to state that
 * the execution was suspended and will not return any result immediately.
 * [suspendCoroutine] can be used to specify the desired behavior of the code.
 */
public val COROUTINE_SUSPENDED: Any get() = CoroutineSingletons.COROUTINE_SUSPENDED

// Using enum
// 1. It makes SafeContinuation serializable with all kinds of serialization frameworks (since all of them natively support enums)
// 2. It improves debugging experience, since you clearly see toString() value of those objects and what package they come from
@SinceKotlin("1.3")
internal enum class CoroutineSingletons { COROUTINE_SUSPENDED, UNDECIDED, RESUMED }

package kotlin.experimental

/** Performs a bitwise AND operation between the two values. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.and(other: Byte): Byte =
    (this.toInt() and other.toInt()).toByte()

/** Performs a bitwise OR operation between the two values. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.or(other: Byte): Byte =
    (this.toInt() or other.toInt()).toByte()

/** Performs a bitwise XOR operation between the two values. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.xor(other: Byte): Byte =
    (this.toInt() xor other.toInt()).toByte()

/** Inverts the bits in this value. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun Byte.inv(): Byte =
    (this.toInt().inv()).toByte()
Since 1.2, the Kotlin compiler supports

* [RequireKotlinVersionKind.LANGUAGE_VERSION],
* [RequireKotlinVersionKind.COMPILE_VERSION] and [RequireKotlinVersionKind.API_VERSION].

If the actual value of [versionKind] is something different (e.g. a new version kind, added in future versions of Kotlin), Kotlin 1.2 is going to ignore this [RequireKotlin] altogether, where as Kotlin before 1.2 is going to treat this as a requirement on the language version. This annotation is erased at compile time; its arguments are stored in a more compact form in the Kotlin metadata.

@Target(AnnotationTarget.CLASS,
            AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY,
            AnnotationTarget.CONSTRUCTOR,
            AnnotationTarget.TYPEALIAS)
@Retention(AnnotationRetention.SOURCE)
@Repeatable
@SinceKotlin("1.2")
internal annotation class RequireKotlin(version: String,
                                          message: String = "",
                                          level: DeprecationLevel = DeprecationLevel.ERROR,
                                          versionKind: RequireKotlinVersionKind = RequireKotlinVersionKind.LANGUAGE_VERSION,
                                          errorCode: Int = -1)

/**
 * The kind of the version that is required by [RequireKotlin].
 */
@SinceKotlin("1.2")
internal enum class RequireKotlinVersionKind {
    LANGUAGE_VERSION,
    COMPILER_VERSION,
    API_VERSION,
}

/**
 * Specifies that this declaration is a part of special DSL, used for constructing
 * function's contract.
 */
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
internal annotation class ContractsDsl

private class NotNullVar<T : Any>() : ReadWriteProperty<Any?, T> {
    private var value: T? = null

    public override fun getValue(thisRef: Any?, property: KProperty<*>): T {
        return value
    }

    public override fun setValue(thisRef: Any?, property: KProperty<*>, value: T) {
        this.value = value
    }

}
property delegate for a read/write property that calls callback functions when changed.

```kotlin
@throws IllegalArgumentException if [from] is greater than or equal to [until].
```
override fun nextDouble(until: Double): Double = defaultRandom.nextDouble(until)
override fun nextDouble(from: Double, until: Double): Double = defaultRandom.nextDouble(from, until)
override fun nextFloat(): Float = defaultRandom.nextFloat()
override fun nextBytes(array: ByteArray): ByteArray = defaultRandom.nextBytes(array)
override fun nextBytes(size: Int): ByteArray = defaultRandom.nextBytes(size)
override fun nextBytes(array: ByteArray, fromIndex: Int, toIndex: Int): ByteArray = defaultRandom.nextBytes(array, fromIndex, toIndex)

@Deprecated("Use Default companion object instead", level = DeprecationLevel.HIDDEN)
@Suppress("DEPRECATION_ERROR")
@kotlin.jvm.JvmField

@Deprecated("Use Default companion object instead", level = DeprecationLevel.HIDDEN)
public object Companion : Random() {
    override fun nextBits(bitCount: Int): Int = Default.nextBits(bitCount)
}

/**
 * Returns a repeatable random number generator seeded with the given [seed] `Int` value.
 * Two generators with the same seed produce the same sequence of values within the same version of Kotlin runtime.
 * *Note:* Future versions of Kotlin may change the algorithm of this seeded number generator so that it will return a sequence of values different from the current one for a given seed.
 * *@sample samples.random.Randoms.seededRandom
 */
@SinceKotlin("1.3")
public fun Random(seed: Int): Random = XorWowRandom(seed, seed.shr(31))

/**
 * Returns a repeatable random number generator seeded with the given [seed] `Long` value.
 * Two generators with the same seed produce the same sequence of values within the same version of Kotlin runtime.
 * *Note:* Future versions of Kotlin may change the algorithm of this seeded number generator so that it will return a sequence of values different from the current one for a given seed.
 * *@sample samples.random.Randoms.seededRandom
 */
@SinceKotlin("1.3")
public fun Random(seed: Long): Random = XorWowRandom(seed.toInt(), seed.shr(32).toInt())

/**
 * Gets the next random `Int` from the random number generator in the specified [range].
 * Generates an `Int` random value uniformly distributed in the specified [range]:
 * from `range.start` inclusive to `range.endInclusive` inclusive.
 * *Note:* Throws `IllegalArgumentException` if [range] is empty.
 * *@sample samples.random.Randoms.nextIntRange
 */
@SinceKotlin("1.3")
public fun Random.nextInt(range: IntRange): Int = when {
    range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range: $range")
    range.last < Int.MAX_VALUE -> nextInt(range.first, range.last + 1)
    range.first > Int.MIN_VALUE -> nextInt(range.first - 1, range.last) + 1
    else -> nextInt()
}

/**
 * Gets the next random `Long` from the random number generator in the specified [range].
 * Generates a `Long` random value uniformly distributed in the specified [range]:
 * from `range.start` inclusive to `range.endInclusive` inclusive.
 * *Note:* Throws `IllegalArgumentException` if [range] is empty.
 * *@sample samples.random.Randoms.nextLongRange
 */
@SinceKotlin("1.3")
public fun Random.nextLong(range: LongRange): Long = when {
    range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range: $range")
    range.last < Long.MAX_VALUE -> nextLong(range.first, range.last + 1)
    range.first > Long.MIN_VALUE -> nextLong(range.first - 1, range.last) + 1
    else -> nextLong()
}

internal expect fun defaultPlatformRandom(): Random
internal expect fun doubleFromParts(hi26: Int, low27: Int): Double

@UseExperimental(ExperimentalStdlibApi::class)
internal fun fastLog2(value: Int): Int = 31 - value.countLeadingZeroBits()

/** Takes upper [bitCount] bits (0..32) from this number. */
internal fun Int.takeUpperBits(bitCount: Int): Int =
    this.ushr(32 - bitCount) and (-bitCount).shr(31)

internal fun checkRangeBounds(from: Int, until: Int) = require(until > from) {
    boundsErrorMessage(from, until)
}

internal fun checkRangeBounds(from: Long, until: Long) = require(until > from) {
    boundsErrorMessage(from, until)
}

internal fun checkRangeBounds(from: Double, until: Double) = require(until > from) {
    boundsErrorMessage(from, until)
}

package kotlin.random

/**
 * Gets the next random `UInt` from the random number generator.
 * Generates a `UInt` random value uniformly distributed between `UInt.MIN_VALUE` and `UInt.MAX_VALUE` (inclusive).
 * *@sample samples.random.Randoms.nextInt
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUInt(): UInt = nextInt().toUInt()
Generates a `UInt` random value uniformly distributed between `0` (inclusive) and the specified `until` bound (exclusive).

@throws IllegalArgumentException if `until` is zero.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUInt(until: UInt): UInt = nextUInt(0u, until)

Generates a `UInt` random value uniformly distributed between the specified `from` (inclusive) and `until` (exclusive) bounds.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUInt(from: UInt, until: UInt): UInt {
    checkUIntRangeBounds(from, until)
    val signedFrom = from.toInt() xor Int.MIN_VALUE
    val signedUntil = until.toInt() xor Int.MIN_VALUE
    val signedResult = nextInt(signedFrom, signedUntil) xor Int.MIN_VALUE
    return signedResult.toUInt()
}

Generates a `UInt` random value uniformly distributed in the specified range: `from` (inclusive) to `until` (exclusive).

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUInt(range: UIntRange): UInt = when {
    range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range: $range")
    range.last < UInt.MAX_VALUE ->
        nextUInt(range.first, range.last + 1u)
    range.first >= UInt.MIN_VALUE ->
        nextUInt(range.first - 1u, range.last) + 1u
    else ->
        nextUInt(rng.nextInt(rng.getMin()) < 0L, rng.getMax())
    else ->
        nextUInt()
}

Fills the specified unsigned byte array with random bytes and returns it.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUBytes(array: UByteArray): UByteArray {
    nextBytes(array.asByteArray())
    return array
}

Fills a subrange of the specified `UByte` array starting from `fromIndex` inclusive and ending `toIndex` exclusive with random UBytes.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUBytes(array: UByteArray, fromIndex: Int = 0, toIndex: Int = array.size): UByteArray {
    nextBytes(array.asByteArray(), fromIndex, toIndex)
    return
@ExperimentalUnsignedTypes
internal fun checkUIntRangeBounds(from: UInt, until: UInt) = require(until > from) { boundsErrorMessage(from, until) }

@ExperimentalUnsignedTypes
internal fun checkULongRangeBounds(from: ULong, until: ULong) = require(until > from) { boundsErrorMessage(from, until) }

import kotlin.random

/**
 * Random number generator, using Marsaglia's "xorwow" algorithm
 */
internal class XorWowRandom
    internal constructor(
        x: Int, y: Int, z: Int, w: Int, v: Int, addend: Int
    ) : Random() {

    init {
        require((x or y or z or w or v) != 0) { "Initial state must have at least one non-zero element." }

        // some trivial seeds can produce several values with zeroes in upper bits, so we discard first 64
        repeat(64) { nextInt() }
    }

    override fun nextInt(): Int {
        // Equivalent to the xorxow algorithm
        var t = x
        t = t xor (t ushr 2)
        x = y
        y = z
        z = w
        val v0 = v
        w = v0
        t = (t xor (t shl 1)) xor v0 xor (v0 shl 4)
        v = t
        addend += 362437
        return t + addend
    }

    override fun nextBits(bitCount: Int): Int =
        nextInt().takeUpperBits(bitCount)
}

import kotlin.ranges

/**
 * Represents a range of \[Comparable\] values.
 */
private open class ComparableRange<T : Comparable<T>>(
    start: T,
    endInclusive: T
) : ClosedRange<T> {

    override fun equals(other: Any?): Boolean {
        return other is ComparableRange<*> && (isEmpty() && other.isEmpty() ||
                start == other.start && endInclusive == other.endInclusive)
    }

    override fun hashCode(): Int = 31 * start.hashCode() + endInclusive.hashCode()

    override fun toString(): String = "$start..$endInclusive"
}

/**
 * Creates a range from this \[Comparable\] value to the specified \[that\] value.
 * This value needs to be smaller than \[that\] value, otherwise the returned range will be empty.
 */
public operator fun <T : Comparable<T>> T.rangeTo(that: T): ClosedRange<T> = ComparableRange(this, that)

/**
 * Represents a range of floating point numbers.
 * Extends \[ClosedRange\] interface providing custom operation \[lessThanOrEquals\] for comparing values of range domain type.
 * This interface is implemented by floating point ranges returned by \[Float.rangeTo\] and \[Double.rangeTo\] operators to achieve IEEE-754 comparison order instead of total order of floating point numbers.
 */
@SinceKotlin("1.1")
public interface ClosedFloatingPointRange<T : Comparable<T>> : ClosedRange<T> {
    override fun contains(value: T): Boolean = lessThanOrEquals(start, value) && lessThanOrEquals(value, endInclusive)
    override fun isEmpty(): Boolean = !lessThanOrEquals(start, endInclusive)

    /**
     * Compares two values of range domain type and returns true if first is less than or equal to second.
     */
    fun lessThanOrEquals(a: T, b: T): Boolean
}

/**
 * A closed range of values of type `Double`.
 */
private class ClosedDoubleRange(
    start: Double,
    endInclusive: Double
) :

    ClosedFloatingPointRange<Double> {
        override val start: Double get() = _start
        override val endInclusive: Double get() = _endInclusive

        override fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b
        override fun contains(value: Double): Boolean = value >= _start && value <= _endInclusive

        override fun equals(other: Any?): Boolean {
            return other is ClosedDoubleRange &&
                _start == other._start && _endInclusive == other._endInclusive
        }

        override fun hashCode(): Int = 31 * start.hashCode() + endInclusive.hashCode()

        override fun nextInt(): Int {
            // Some trivial seeds can produce several values with zeroes in upper bits, so we discard first 64
            repeat(64) { nextInt() }
        }

        override fun toString(): String = "$_start..$_endInclusive"
value to the specified [that] value.

* Numbers are compared with the ends of this range according to IEEE-754.

* @sample samples.ranges.Ranges.rangeFromDouble

@SinceKotlin("1.1")
public operator fun Double.rangeTo(that: Double): ClosedDoubleRange = ClosedDoubleRange(this, that)


/**
 * A closed range of values of type `Float`.
 * Numbers are compared with the ends of this range according to IEEE-754.
 * @sample samples.ranges.Ranges.rangeFromFloat
 */
@SinceKotlin("1.1")
public operator fun Float.rangeTo(that: Float): ClosedFloatRange = ClosedFloatRange(this, that)


/**
 * Creates a range from this [Float] value to the specified [that] value.
 * Numbers are compared with the ends of this range according to IEEE-754.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@LowPriorityInOverloadResolution
    element != null && contains(element)


internal fun checkStepIsPositive(isPositive: Boolean, step: Number) {
    if (!isPositive)
        throw IllegalArgumentException("Step must be positive, was: $step."

@file:kotlin.jvm.JvmName("KClasses")
@file:Suppress("UNCHECKED_CAST")

package kotlin.reflect

/**
 * Casts the given [value] to the class represented by this [KClass] object.
 * Throws an exception if the value is `null` or if it is not an instance of this class.
 * This is an experimental function that behaves as a similar function from kotlin.reflect.full on JVM.
 * @see [KClass.isInstance]
 * @see [KClass.safeCast]
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@LowPriorityInOverloadResolution
public inline fun <reified T> typeOf(): KType =
    throw UnsupportedOperationException("This function is implemented as an intrinsic on all supported platforms.")

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

/**
 * An object to which char sequences and values can be appended.
 */
expect interface Appendable {
    /**
     * Appends the specified character [value] to this Appendable and returns this instance.
     * @param value the character to append.
     */
    fun append(value: Char): Appendable

    /**
     * Returns a runtime representation of the given reified type [T] as an instance of [KType].
     * Returns a runtime representation of the given reified type [T] as an instance of [KType].
     * @param value the character to append.
     */
    fun <T> typeOf(): KType =
        throw UnsupportedOperationException("This function is implemented as an intrinsic on all supported platforms.")

    */

* *
replaceIndentByMargin(\"\", marginPrefix)\n\n/**
 * Detects indent by [marginPrefix] as it does [trimMargin] and replace it with [newIndent].\n * @param marginPrefix non-blank string, which is used as a margin delimiter.
 * Default is \'|\' (pipe character).\n * @param newIndent new indent string, which is used as a new indent delimeter.
 * Default is \"\" (empty string).
 */
public fun String.replaceIndentByMargin(newIndent: String = \"\", marginPrefix: String = \"\": String { require(marginPrefix.isNotBlank()) { \"marginPrefix must be non-blank string.\" }\n     val lines = lines().\n     return lines.reindent(length + newIndent.length * lines.size, getIndentFunction(newIndent), \n     { line -> line.firstNonWhitespaceIndex == -1 -> null\n     line.startsWith(marginPrefix, firstNonWhitespaceIndex) -> line.substring(firstNonWhitespaceIndex + marginPrefix.length)\n     else -> null\n     })\n }

/**
 * Detects a common minimal indent of all the input lines, removes it from every line and also removes the first and the last line if they are blank (notice difference blank vs empty).\n * @param indent common indent level in lines.\n * @param inCase if there are non-blank lines with no leading whitespace characters (no indent at all) then this common indent is 0, and therefore this function doesn't change the indentation.\n * @param doesn't preserve the original line endings.\n * @param sample\n * @see trimMargin\n * @see kotlin.text.isBlank\n */
public fun String.trimIndent(): String = replaceIndent("")

/**
 * Detects a common minimal indent like it does \[trimIndent\] and replaces it with the specified [newIndent].\n * @param newIndent\n * @see trimMargin\n * @see kotlin.text.isBlank\n */
public fun String.replaceIndent(newIndent: String = ": String {\n     val lines = lines()\n
     val minCommonIndent = lines.filter(String::isNotBlank).map(String::indentWidth).min() ?: 0\n     return lines.reindent(length + newIndent.length * lines.size, getIndentFunction(newIndent), \n     { line: String -> line.drop(minCommonIndent) }\n     }

/**
 * Prepends \[indent\] to every line of the original string.\n * @param indent\n * @param sample\n */
public fun String.prependIndent(indent: String = "    "): String =
    lineSequence()\n    .map {\n        when {
            it.isBlank() -> {
                when {
                    it.length < indent.length -> indent\n                    else -> it
                }
            }\n            else -> indent + it
        }
    }\n    .joinTo(StringBuilder(), \"\n\")

private fun String.indentWidth(): Int = indexOfFirst { !it.isWhitespace() }.let { if (it == -1) length else it }

private fun getIndentFunction(indent: String) = when {
    indent.isEmpty() -> { inline: String -> inline }\n    else -> { inline: String -> indent + inline }\n}

private inline fun List<String>.reindent(
    resultSizeEstimate: Int,\n    indentAddFunction: (String) -> String?,\n    indentCutFunction: (String) -> String?\n): String {
    val lastIndex = lastIndex\n    return\n    mapIndexedNotNull { index, value ->
        if ((index == 0 || index == lastIndex) && value.isBlank())
        null\n        else
            indentCutFunction(value)?.let(indentAddFunction) ?: value
    }
    .joinTo(StringBuilder(resultSizeEstimate), \"\n\")
    .toString()

@package:org.jetbrains.kotlin

public object Typography {
    /** The character "\" », quotation mark */
    public const val quote: Char = '\u0022'
    /** The character \$ », dollar sign */
    public const val dollar: Char = '\u0024'
    /** The character & », ampersand */
    public const val amp: Char = '\u0026'
    /** The character < », less-than sign */
    public const val less: Char = '\u003C'
    /** The character > », greater-than sign */
    public const val greater: Char = '\u003E'
    /** The non-breaking space character */
    public const val nbsp: Char = '\u00A0'
    /** The character × », times */
    public const val times: Char = '\u00D7'
    /** The character ¢ », cent */
    public const val cent: Char = '\u00A2'
    /** The character £ », pound */
    public const val pound: Char = '\u00A3'
    /** The character § », section */
    public const val section: Char = '\u00A7'
    /** The character © », copyright */
    public const val copyright: Char = '\u00A9'
    /** The character « », leftGuillemete */
    public const val leftGuillemete: Char = '\u00AB'
    /** The character » », rightGuillemete */
    public const val rightGuillemete: Char = '\u00BB'
    /** The character ® », registered */
    public const val registered: Char = '\u00AE'
    /** The character ° », degree */
    public const val degree: Char = '\u00B0'
    /** The character ± », plusMinus */
    public const val plusMinus: Char = '\u00B1'
    /** The character · », middleDot */
    public const val middleDot: Char = '\u00B7'
    /** The character ½ », half */
    public const val half: Char = '\u00BD'
    /** The character \"\" », doublequote */
    public const val doublequote: Char = '\u201C'
    /** The character \"\" », singlequote */
    public const val singlequote: Char = '\u2018'}
character &x214; /* An public const val mdash: Char = \u2014
 /** The character &x218; * An public const val leftSingleQuote: Char = \u2018
 /** The character &x21A; * An public const val lowSingleQuote: Char = \u201A
 /** The character &x21C; * An public const val leftDoubleQuote: Char = \u201C
 /** The character &x21D; * An public const val rightDoubleQuote: Char = \u201D
 /** The character &x21E; * An public const val lowDoubleQuote: Char = \u201E
 /** The character &x220; * An public const val dagger: Char = \u2020
 /** The character &x221; * An public const val doubleDagger: Char = \u2021
 /** The character &x222; * An public const val bullet: Char = \u2022
 /** The character &x223; * An public const val ellipsis: Char = \u2026
 /** The character &x224; * An public const val prime: Char = \u2032
 /** The character &x225; * An public const val doublePrime: Char = \u2033
 /** The character &x226; * An public const val euro: Char = \u20AC
 /** The character &x227; * An public const val tm: Char = \u2122
 /** The character &x228; * An public const val almostEqual: Char = \u2248
 /** The character &x229; * An public const val notEqual: Char = \u2260
 /** The character &x22A; * An public const val lessOrEqual: Char = \u2264
 /** The character &x22B; * An public const val greaterOrEqual: Char = \u2265
} /*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 *
@package kotlin.text

/**
 * Represents a collection of captured groups in a single match of a regular expression.
 */
public interface MatchGroupCollection : Collection<MatchGroup?> {

    /** Returns a group with the specified index.
     *
     * @return An instance of [MatchGroup] if the group with the specified index was matched or `null` otherwise.
     * @throws IllegalArgumentException if there is no group with the specified name defined in the regex pattern.
     * @throws UnsupportedOperationException if getting named groups isn’t supported on the current platform.
     */
    public operator fun get(index: Int): MatchGroup?
}

/**
 * Extends [MatchGroupCollection] by introducing a way to get matched groups by name, when regex supports it.
 */
@SinceKotlin("1.1")
public interface MatchNamedGroupCollection : MatchGroupCollection {
    /**
     * Returns a named group with the specified name.
     * @return An instance of [MatchGroup] if the group with the specified name was matched or `null` otherwise.
     * @throws IllegalFormatException if the named group isn’t supported on the current platform.
     */
    public operator fun get(name: String): MatchGroup?
}

/**
 * Represents the results from a single regular expression match.
 */
public interface MatchResult {
    /** The range of indices in the original string where match was captured.
     */
    public val range: IntRange
    /** The substring from the input string captured by this match.
     */
    public val value: String
    /** A collection of groups matched by the regular expression.
     */
    public val groups: MatchGroupCollection
    /** A list of matched indexed group values.
     */
    public val groupValues: List<String>

    /** Returns a new [MatchResult] with the results for the next match, starting at the position
     * at which the last match ended (at the character after the last
matched character).

 * /n public fun next(): MatchResult?\n
 /**
 * Provides components for destructuring assignment of group values.\n * [component1] corresponds to the value of the first group,\n * [component2] \n * If the group in the regular expression is optional and there were no match captured by that group,\n * corresponding component value is an empty string.\n
 * @sample samples.text.Regexps.matchDestructuringToGroupValues
 */
    public fun next(): MatchResult?

    /**
     * Provides components for destructuring assignment of group values.
     *
     * [component1] corresponds to the value of the first group,\n     * [component2] \n     * If the group in the regular expression is optional and there were no match captured by that group,\n     * corresponding component value is an empty string.\n
     * @sample samples.text.Regexps.matchDestructuringToGroupValues
     */
    public class Destructured internal
        constructor(public val match: MatchResult) {
            @kotlin.internal.InlineOnly
            public operator inline fun
                component1():  String = match.groupValues[1]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component2():  String = match.groupValues[2]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component3():  String = match.groupValues[3]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component4():  String = match.groupValues[4]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component5():  String = match.groupValues[5]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component6():  String = match.groupValues[6]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component7():  String = match.groupValues[7]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component8():  String = match.groupValues[8]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component9():  String = match.groupValues[9]
            @kotlin.internal.InlineOnly
            public operator inline fun
                component10(): String = match.groupValues[10]

                /**
                 * Returns destructured group values as a list of strings.\n                 * First value in the returned list corresponds to the value of the first group, and so on.\n
                 * @sample samples.text.Regexps.matchDestructuringToGroupValues
                 */
            public fun toList(): List<String> = match.groupValues.subList(1, match.groupValues.size)
        }

        /**
         * Returns destructured group values as a list of strings.\n         * First value in the returned list corresponds to the value of the first group, and so on.\n
         * @sample samples.text.Regexps.matchDestructuringToGroupValues
         */
        public fun toList(): List<String> = match.groupValues.subList(1, match.groupValues.size)

        /**
         * Returns destructured group values as a list of strings.\n         * First value in the returned list corresponds to the value of the first group, and so on.\n
         * @sample samples.text.Regexps.matchDestructuringToGroupValues
         */
        public fun toList(): List<String> = match.groupValues.subList(1, match.groupValues.size)

        /**
         * Returns destructured group values as a list of strings.\n         * First value in the returned list corresponds to the value of the first group, and so on.\n
         * @sample samples.text.Regexps.matchDestructuringToGroupValues
         */
        public fun toList(): List<String> = match.groupValues.subList(1, match.groupValues.size)

        /**
         * Returns destructured group values as a list of strings.\n         * First value in the returned list corresponds to the value of the first group, and so on.\n
         * @sample samples.text.Regexps.matchDestructuringToGroupValues
         */
        public fun toList(): List<String> = match.groupValues.subList(1, match.groupValues.size)
divided by the given [scale] number. */

    public operator fun div(scale: Double): Duration = Duration(value / scale)

    /** Returns a number that is the ratio of this and [other] duration values. */
    public operator fun div(other: Duration): Double = this.value / other.value

    /** Returns true, if the duration value is less than zero. */
    public fun isNegative(): Boolean = value < 0

    /** Returns true, if the duration value is greater than zero. */
    public fun isPositive(): Boolean = value > 0

    /** Returns true, if the duration value is infinite. */
    public fun isInfinite(): Boolean = value.isInfinite()

    /** Returns true, if the duration value is finite. */
    public fun isFinite(): Boolean = value.isFinite()

    /** Returns the absolute value of this value. The returned value is always non-negative. */
    public val absoluteValue: Duration get() = if (isNegative()) -this else this

    override fun compareTo(other: Duration): Int = this.value.compareTo(other.value)

    // splitting to components

    /**
     * Splits this duration into days, hours, minutes, seconds, and nanoseconds and executes the given [action]
     * with these components.
     *
     * - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 1_000_000_000;
     * - `days` represents the whole number of days in this duration.
     *
     * If the value doesn't fit in [Int] range, i.e. it's greater than [Long.MAX_VALUE] or less than [Long.MIN_VALUE],
     * it is coerced into that range.
     */
    public inline fun <T> toComponents(action: (days: Int, hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T =
        action(days.toInt(), hoursComponent, minutesComponent, secondsComponent, nanosecondsComponent)

    /**
     * Splits this duration into hours, minutes, and nanoseconds and executes the given [action] with these components.
     *
     * - `seconds` represents the whole number of seconds in this duration.
     * - `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;
     * - `hours` represents the whole number of hours in this duration, and its absolute value is less than 24;
     *
     * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
     * it is coerced into that range.
     */
    public inline fun <T> toComponents(action: (hours: Int, minutes: Int, seconds: Int) -> T): T =
        action(hours.toInt(), minutesComponent, secondsComponent)

    /**
     * Splits this duration into minutes, seconds, and nanoseconds and executes the given [action] with these components.
     *
     * - `seconds` represents the whole number of seconds in this duration.
     * - `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;
     * - `hours` represents the whole number of hours in this duration.
     *
     * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
     * it is coerced into that range.
     */
    public inline fun <T> toComponents(action: (minutes: Int, seconds: Int, nanoseconds: Int) -> T): T =
        action(minutes.toInt(), secondsComponent, nanosecondsComponent)

    /**
     * Splits this duration into seconds, and nanoseconds and executes the given [action] with these components.
     *
     * - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 1_000_000_000;
     * - `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;
     *
     * If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE],
     * it is coerced into that range.
     */
    public inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Long) -> T): T =
        action(seconds.toLong(), nanosecondsComponent)

    /**
     * Returns the value of this duration expressed as a [Double] number of the specified [unit]. */
    public fun toDouble(unit: DurationUnit): Double = convertDurationUnit(value, storageUnit, unit)
/**
 * Returns the value of this duration expressed as a [Long] number of the specified [unit].
 */
fun toLong(unit: DurationUnit): Long = toDouble(unit).toLong()

/**
 * Returns the value of this duration expressed as an [Int] number of the specified [unit].
 */
fun toInt(unit: DurationUnit): Int = toDouble(unit).toInt()

/** The value of this duration expressed as a [Double] number of days. */
fun toLongNanoseconds(): Long = toLong(DurationUnit.NANOSECONDS)

/** The value of this duration expressed as a [Double] number of milliseconds. */
fun toLongMilliseconds(): Long = toLong(DurationUnit.MILLISECONDS)

/** The value of this duration expressed as a [Double] number of nanoseconds. */
fun toDoubleMilliseconds(): Double = toLong(DurationUnit.MILLISECONDS)

/** The value of this duration expressed as a [Double] number of seconds. */
fun toSeconds(): Double = toDouble(DurationUnit.SECONDS)

/** The value of this duration expressed as a [Double] number of minutes. */
fun toMinutes(): Double = toDouble(DurationUnit.MINUTES)

/** The value of this duration expressed as a [Double] number of hours. */
fun toHours(): Double = toDouble(DurationUnit.HOURS)

/** The value of this duration expressed as a [Double] number of days. */
fun toDays(): Double = toDouble(DurationUnit.DAYS)

/** The value of this duration expressed as a [Double] number of months. */
fun toMonths(): Double = toDouble(DurationUnit.MONTHS)

/** The value of this duration expressed as a [Double] number of years. */
fun toYears(): Double = toDouble(DurationUnit.YEARS)
Open Source Used In DNAC Platform Antelope 8322

```kotlin
@sample samples.time.Durations.toStringDecimals
    /**
     * public fun toString(unit: DurationUnit, decimals: Int = 0): String {
         require(decimals >= 0) { "decimals must be not negative, but was $decimals" }
         if (isInfinite()) return value.toString() + "Inf" *n +
         val number = toDouble(unit) *n +
         return when {
             abs(number) < 1e14 -> formatToExactDecimals(number, decimals.coerceAtMost(12))
             else -> formatScientific(number)
         } + unit.shortName(*) *n +
     */
    public fun toString(unit: DurationUnit, decimals: Int = 0): String {
        require(decimals >= 0) { "decimals must be not negative, but was $decimals" }
        if (isInfinite()) return value.toString() + "Inf" *n +
        val number = toDouble(unit) *n +
        return when {
            abs(number) < 1e14 -> formatToExactDecimals(number, decimals.coerceAtMost(12))
            else -> formatScientific(number)
        } + unit.shortName(*) *n +
    }

    /**
     * Returns an ISO-8601 based string representation of this duration.
     * The returned value is presented in the format 'PThMMs.fS', where 'T', 'h', 'm', 's' are the integer components of this duration (see [toComponents]) and 'f' is a fractional part of second. Depending on the roundness of the value the fractional part can be formatted with either 0, 3, 6, or 9 decimal digits.
     * If the hours component absolute value of this duration is greater than [Int.MAX_VALUE], it is replaced with [Int.MAX_VALUE].
     * so the infinite duration is formatted as \"P\"T2147483647H\".
     * Negative durations are indicated with the sign '-' in the beginning of the returned string, for example, \"-P\"T5M30S\".
     */
    @UseExperimental(ExperimentalStdlibApi::class)
    public fun toIsoString(): String = buildString {
        if (isNegative()) append('-') *n +
        append("PT") *n +
        absoluteValue.toComponents { hours, minutes, seconds, nanoseconds ->
            val hasHours = hours != 0
            val hasMinutes = minutes != 0 || hasHours
            val hasSeconds = seconds != 0 || (hasMinutes && hasHours)
            if (hasHours) {
                append(hours).append('H') *n +
            }
            if (hasMinutes) {
                append(minutes).append('M') *n +
            }
            if (hasSeconds || (!hasHours && !hasMinutes)) {
                append(seconds) *n +
                if (nanoseconds != 0) {
                    val nss = nanoseconds.toString().padStart(9, '0') *n +
                    when {
                        nanoseconds % 1_000_000 == 0 -> appendRange(nss, 0, 3) *n +
                        nanoseconds % 1_000 == 0 -> appendRange(nss, 0, 6) *n +
                        else -> append(nss) *n +
                    }
                }
                append('S') *n +
            } *n +
        } *n +
    }

    // constructing from number of units
    // extension functions

    /**
     * Returns a [Duration] equal to this [Int] number of the specified [unit].
     */
    @SinceKotlin("1.3")
    @ExperimentalTime
    public fun Int.toDuration(unit: DurationUnit): Duration =
        convertDurationUnit(this, unit, StorageUnit.NANOSECONDS) *n +
    }

    /**
     * Returns a [Duration] equal to this [Long] number of nanoseconds.
     */
    @SinceKotlin("1.3")
    @ExperimentalTime
    public fun Long.nanoseconds get() =
        toDuration(DurationUnit.NANOSECONDS) *n +
    }

    /**
     * Returns a [Duration] equal to this [Double] number of nanoseconds.
     */
    @SinceKotlin("1.3")
    @ExperimentalTime
    public fun Double.nanoseconds get() =
        toDuration(DurationUnit.MICROSECONDS) *n +
    }

    /**
     * Returns a [Duration] equal to this [Int] number of milliseconds.
     */
    @SinceKotlin("1.3")
    @ExperimentalTime
    public fun Int.milliseconds get() =
        toDuration(DurationUnit.MILLISECONDS) *n +
    }

    /**
     * Returns a [Duration] equal to this [Long] number of milliseconds.
     */
    @SinceKotlin("1.3")
    @ExperimentalTime
    public fun Long.milliseconds get() =
        toDuration(DurationUnit.MILLISECONDS) *n +
    }

    /**
     * Returns a [Duration] equal to this [Double] number of milliseconds.
     */
    @SinceKotlin("1.3")
    @ExperimentalTime
    public fun Double.milliseconds get() =
        toDuration(DurationUnit.MILLISECONDS) *n +
    }

    /**
     * Returns a [Duration] equal to this [Int] number of seconds.
     */
    @SinceKotlin("1.3")
    @ExperimentalTime
    public fun Int.seconds get() =
        toDuration(DurationUnit.SECONDS) *n +
    }
```
Open Source Used In DNAC Platform Antelope

Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.

package kotlin.time

@SinceKotlin("1.3")
@ExperimentalTime
internal expect object MonotonicTimeSource : TimeSource

/**
 * An abstract class used to implement time sources that return their readings as [Long] values in the specified [unit].
 * The unit in which this time source's readings are expressed.
 * *An@ExperimentalTime
 */
@SinceKotlin("1.3")
@ExperimentalTime
public abstract class AbstractLongTimeSource(protected val unit: DurationUnit) : TimeSource {
    /**
     * This protected method should be overridden to return the current reading of the time source expressed as a [Long] number in the unit specified by the [unit] property.
     * An protected abstract fun read(): Long
     */
    protected abstract fun read(): Long

    private class LongTimeMark(private val startedAt: Long, private val timeSource: AbstractLongTimeSource, private val offset: Duration) : TimeMark() {
        override fun elapsedNow(): Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset
        override fun plus(duration: Duration): TimeMark = LongTimeMark(startedAt, timeSource, offset + duration)
    }

    override fun markNow(): TimeMark = LongTimeMark(read(), this, Duration.ZERO)
}

/**
 * A time source that has programmatically updatable readings. It is useful as a predictable source of time in tests.
 * The current reading value can be advanced by the specified duration amount with the operator [plusAssign]:
 * ```
 * val timeSource = TestTimeSource()
 * timeSource += 10.seconds
 * ```
 * The current reading value is stored as a [Long] number of nanoseconds, so it's capable to represent a time range of approximately ±292 years. Should the reading value overflow as the result of [plusAssign] operation, an [IllegalStateException] is thrown.
 * A time source that has programmatically updateable readings. It is useful as a predictable source of time in tests.
 */
@SinceKotlin("1.3")
@ExperimentalTime
public class TestTimeSource : AbstractLongTimeSource(unit = DurationUnit.NANOSECONDS) {
    private var reading: Long = 0L

    override fun read(): Long = reading

    /**
     * Advances the current reading value of this time source by the specified [duration].
     * * An * [duration] value is rounded down towards zero when converting it to a [Long] number of nanoseconds.
     * * For example, if the duration being added is 0.6.nanoseconds, the reading doesn't
advance because
* the duration value is rounded to zero nanoseconds.
* @throws
IllegalStateException when the reading value overflows as the result of this operation.
* public operator fun
plusAssign(duration: Duration) { val delta = duration.toDouble(unit) val longDelta = delta.toLong()
reading = if (longDelta != Long.MIN_VALUE && longDelta != Long.MAX_VALUE) { if when delta
fits in long, add it as long
val newReading = reading + longDelta
if (reading xor longDelta != 0 && reading xor newReading < 0) overflow(duration)
newReading
} else { if when delta is
greater than long, add it as double
val newReading = reading + delta
if (newReading >
Long.MAX_VALUE || newReading < Long.MIN_VALUE) overflow(duration)
newReading.toLong()
}
}

private fun overflow(duration: Duration) {
throw IllegalStateException("TestTimeSource will
overflow if its reading ${reading}ns is advanced by $duration.
")
}

@SinceKotlin("1.3")
@ExperimentalTime
@Deprecated("Use TimeSource.Monotonic instead.",
ReplaceWith("TimeSource.Monotonic", "kotlin.time.TimeSource"))
public typealias MonoClock =
TimeSource.Monotonic

@SinceKotlin("1.3")
@ExperimentalTime
@Deprecated("Use
AbstractLongTimeSource instead.",
ReplaceWith("AbstractLongTimeSource",
"kotlin.time.AbstractLongTimeSource"))
public typealias AbstractLongClock =
AbstractLongTimeSource

@SinceKotlin("1.3")
@ExperimentalTime
@Deprecated("Use
AbstractDoubleTimeSource instead.",
ReplaceWith("AbstractDoubleTimeSource",
"kotlin.time.AbstractDoubleTimeSource"))
public typealias AbstractDoubleClock =
AbstractDoubleTimeSource

@SinceKotlin("1.3")
@ExperimentalTime
@Deprecated("Use
TestTimeSource instead.",
ReplaceWith("TestTimeSource", "kotlin.time.TestTimeSource"))
public typealias TestClock =
TestTimeSource

public data class TimedValue<T>(val value: T, val duration:
Duration)

public inline fun measureTimedValue(block: () -> T): TimedValue<T> {
contract {
callsInPlace(block, InvocationKind.EXACTLY_ONCE)
}
return
TimeSource.Monotonic.measureTimedValue(block)
}

public inline fun <T> TimeSource.measureTimedValue(block: () -> T): TimedValue<T> {
contract {
callsInPlace(block, InvocationKind.EXACTLY_ONCE)
}
val mark = markNow()
val result = block()
return TimedValue(result, mark.elapsedNow())
}


is `null`.

```kotlin
class KotlinVersion(val major: Int, val minor: Int, val patch: Int) :
    Comparable<KotlinVersion> {
    public constructor(major: Int, minor: Int) :
        this(major, minor, 0)

    private fun versionOf(major: Int, minor: Int, patch: Int): Int {
        require(major in 0..MAX_COMPONENT_VALUE && minor in 0..MAX_COMPONENT_VALUE &&
                patch in 0..MAX_COMPONENT_VALUE) {
            "Version components are out of range: $major.$minor.$patch"
        }
        return major.shl(16) + minor.shl(8) + patch
    }

    public fun atLeast(major: Int, minor: Int, patch: Int): Boolean = // this.version >= versionOf(major, minor, patch)
        this.major > major || (this.major == major &&
                this.minor > minor || this.minor == minor &&
                        this.patch >= patch)

    companion object {
        public const val MAX_COMPONENT_VALUE = 255

        public val CURRENT: KotlinVersion = KotlinVersion(1, 3, 72) // value is written here
            automatically during build
    }
}
```

```kotlin
package kotlin

import kotlin.reflect.KProperty

/**
 * Represents a value with lazy initialization.
 *
 * To create an instance of [Lazy] use the [lazy] function.
 */
public interface Lazy<out T> {
    /**
     * Gets the lazily initialized value of the current Lazy instance.
     * Once the value was initialized it must not change during the rest of lifetime of this Lazy instance.
     */
    public val value: T

    /**
     * Returns `true` if a value for this Lazy instance has been already initialized, and `false` otherwise.
     */
    public fun isInitialized(): Boolean

    companion object {
        public fun of(value: T): Lazy<T> =
            InitializedLazyImpl(value)
    }
}
```

```kotlin
package kotlin

import kotlin.reflect.KProperty

/**
 * Represents a version of the Kotlin standard library.
 * @sinceKotlin("1.1")
 * @see Any.hashCode
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun Any?.hashCode(): Int = this?.hashCode() ?: 0
```

```kotlin
@file:kotlin.jvm.JvmName("LateinitKt")
@file:Suppress("unused")

package kotlin

import kotlin.reflect.KProperty0

/**
 * Returns `true` if this lateinit property has been assigned a value, and `false` otherwise.
 */
@SinceKotlin("1.2")
@InlineOnly
inline val @receiver:AccessibleLateinitPropertyLiteral KProperty0<*>.isInitialized: Boolean
    get() = throw NotImplementedError("Implementation is intrinsic")
```

```kotlin
@file:kotlin.jvm.JvmName("LazyKt")
@file:kotlin.jvm.JvmMultifileClass

package kotlin

import kotlin.reflect.KProperty

/**
 * Creates a version from all three components.
 */
@SinceKotlin("1.1")
public class KotlinVersion(val major: Int, val minor: Int, val patch: Int) :
    Comparable<KotlinVersion> {
    public constructor(major: Int, minor: Int) :
        this(major, minor, 0)

    private fun versionOf(major: Int, minor: Int, patch: Int): Int {
        require(major in 0..MAX_COMPONENT_VALUE && minor in 0..MAX_COMPONENT_VALUE &&
                patch in 0..MAX_COMPONENT_VALUE) {
            "Version components are out of range: $major.$minor.$patch"
        }
        return major.shl(16) + minor.shl(8) + patch
    }

    public fun atLeast(major: Int, minor: Int, patch: Int): Boolean = // this.version >= versionOf(major, minor, patch)
        this.major > major || (this.major == major &&
                this.minor > minor || this.minor == minor &&
                        this.patch >= patch)

    companion object {
        public const val MAX_COMPONENT_VALUE = 255

        public val CURRENT: KotlinVersion = KotlinVersion(1, 3, 72) // value is written here
            automatically during build
    }
}
```

```kotlin
@file:kotlin.jvm.JvmName("LateinitKt")
@file:Suppress("unused")

import kotlin.reflect.KProperty0

/**
 * Represents a value with lazy initialization.
 */
public interface Lazy<out T> {
    /**
     * Gets the lazily initialized value of the current Lazy instance.
     * Once the value was initialized it must not change during the rest of lifetime of this Lazy instance.
     */
    public val value: T

    /**
     * Returns `true` if a value for this Lazy instance has been already initialized, and `false` otherwise.
     */
    public fun isInitialized(): Boolean

    companion object {
        public fun of(value: T): Lazy<T> =
            InitializedLazyImpl(value)
    }
}
```

```kotlin
@file:kotlin.jvm.JvmName("LateinitKt")
@file:Suppress("unused")

import kotlin.reflect.KProperty0

/**
 * Represents a version of the Kotlin standard library.
 * @sinceKotlin("1.3")
 * @see Any.hashCode
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun Any?.hashCode(): Int = this?.hashCode() ?: 0
```

```kotlin
@file:kotlin.jvm.JvmName("LateinitKt")
@file:Suppress("unused")

import kotlin.reflect.KProperty0

/**
 * Represents a value with lazy initialization.
 */
public interface Lazy<out T> {
    /**
     * Gets the lazily initialized value of the current Lazy instance.
     * Once the value was initialized it must not change during the rest of lifetime of this Lazy instance.
     */
    public val value: T

    /**
     * Returns `true` if a value for this Lazy instance has been already initialized, and `false` otherwise.
     */
    public fun isInitialized(): Boolean

    companion object {
        public fun of(value: T): Lazy<T> =
            InitializedLazyImpl(value)
    }
}
```
This extension allows to use instances of Lazy for property delegation:

```kotlin
val property: String by lazy { initializer }
```

Initializer function can be called several times on concurrent access to uninitialized Lazy instance value but only the first returned value will be used as the value of Lazy instance.

No locks are used to synchronize an access to the Lazy instance value; if the instance is accessed from multiple threads, its behavior is undefined.

This mode should not be used unless the Lazy instance is guaranteed never to be initialized from more than one thread.

```kotlin
internal object UNINITIALIZED_VALUE

internal class UnsafeLazyImpl<out T>(initializer: () -> T) : Lazy<T>, Serializable {
    private var initializer: (() -> T)? = initializer
    private var _value: Any? = UNINITIALIZED_VALUE

    override val value: T
        get() {
            if (_value === UNINITIALIZED_VALUE) {
                _value = initializer!!()
                initializer = null
            }
            return _value as T
        }

    override fun isInitialized(): Boolean = _value !== UNINITIALIZED_VALUE

    override fun toString(): String = if (isInitialized())
        value.toString() else "Lazy value not initialized yet."

    private fun writeReplace(): Any = InitializedLazyImpl(value)
}

internal class InitializedLazyImpl<out T>(override val value: T) : Lazy<T>, Serializable {
    override fun isInitialized(): Boolean = true
    override fun toString(): String = value.toString()
}
```

Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

Counts the number of set bits in the binary representation of this Int number.

Counts the number of consecutive most significant bits that are zero in the binary representation of this Int number.

Counts the number of consecutive least significant bits that are zero in the binary representation of this Int number.

Returns a number having a single bit set in the position of the most significant set bit of this Int number.

Rotates the binary representation of this Int number left by the specified bitCount number of bits.

Rotates the binary representation of this Int number right by the specified bitCount number of bits.

Counts the number of set bits in the binary representation of this Long number.

Counts the number of set bits in the binary representation of this DNAC Platform Antelope

Open Source Used In DNAC Platform Antelope 8328
Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long]
number.

`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic expect fun Long.countLeadingZeroBits():
Int\n/**\n*\n\nCounts the number of consecutive least significant bits that are zero in the binary representation of
this [Long]
number.\n\n`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic expect fun
Long.countTrailingZeroBits(): Int\n/**\n*\n\nCounts the number of consecutive least significant bits that are zero in the binary representation of
this [Long]
number.\n\n`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic expect fun
Long.takeHighestOneBit(): Long\n/**\n*\n\nReturns a number having a single bit set in the position of the most
significant set bit of this [Long] number.\n\n`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic expect fun
Long.takeLowestOneBit(): Long\n
/**\n*\n\nCounts the number of set bits in the binary representation of this
[Byte]
number.\n\n`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline
fun Byte.countOneBits(): Int = (toInt() and 0xFF).countOneBits()

/**\n*\n\nCounts the number of consecutive most
significant bits that are zero in the binary representation of this
[Byte]
number.\n\n`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline
fun Byte.countLeadingZeroBits(): Int = (toInt() and 0xFF).countLeadingZeroBits() - (Int.SIZE_BITS -
Byte.SIZE_BITS)

/**\n*\n\nReturns a number having a single bit set in the position of the least
significant set bit of this [Byte] number.\n\n`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline
fun Byte.takeLowestOneBit(): Byte = toInt().takeLowestOneBit().toByte()

/**\n*\n\nRotates the binary representation of this [Byte] number left by the
specified [bitCount] number of bits.\n\n`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline
fun Byte.rotateLeft(bitCount: Int): Byte =
(toInt().shl(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7))).toByte()

/**\n*\n\nRotates the binary representation of this [Byte] number right by the
specified [bitCount] number of bits.\n\n`*\n*/\n\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline
fun Byte.rotateRight(bitCount: Int): Byte =
(toInt() or 0xFF).rotateRight(bitCount).toDouble().toByte()
the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 8)`\n *\n
```kotlin
*\n
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\n@public fun Byte.rotateRight(bitCount: Int): Byte =\n    (toInt().shl(8 - (bitCount and 7)) or (toInt() and 0xFF).ushr(bitCount and 7)).toByte()\n```

* Counts the number of set bits in the binary representation of this [Short] number.\n
```kotlin
*\n
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\n@public fun Short.countOneBits(): Int = (toInt() and 0xFFF).countOneBits()\n*\n
```kotlin
* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Short] number.\n
```kotlin
*\n
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\n@public fun Short.countLeadingZeroBits(): Int =\n    (toInt() and 0xFFF).countLeadingZeroBits() - (Int.SIZE_BITS - Short.SIZE_BITS)\n```

* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Short] number.\n
```kotlin
*\n
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\n@public fun Short.countTrailingZeroBits(): Int = (toInt() or 0x10000).countTrailingZeroBits()\n```

* Returns a number having a single bit set in the position of the most significant set bit of this [Short] number.\n
```kotlin
*\n
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\n@public fun Short.takeHighestOneBit(): Short = (toInt() and 0xFFF).takeHighestOneBit().toShort()\n```

* Returns a number having a single bit set in the position of the least significant set bit of this [Short] number.\n
```kotlin
*\n
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\n@public fun Short.takeLowestOneBit(): Short = (toInt() or 0x10000).takeLowestOneBit().toShort()\n```

* Rotates the binary representation of this [Short] number left by the specified [bitCount] number of bits.\n
```kotlin
*\n
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\n@public fun Short.rotateLeft(bitCount: Int): Short =\n    (toInt().shl(bitCount and 15) or (toInt() and 0xFFF).ushr(16 - (bitCount and 15))).toShort()\n```

* Rotates the binary representation of this [Short] number right by the specified [bitCount] number of bits.\n
```kotlin
*\n
@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\n@public fun Short.rotateRight(bitCount: Int): Short =\n    (toInt().shl(16 - (bitCount and 15)) or (toInt() and 0xFFF).ushr(bitCount and 15)).toShort()\n```

* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
* @package kotlin\n
```
*\n
```
second: B(): Serializable
* Returns string representation of the [Pair] including its [first] and [second] values.
*public override fun toString(): String = 
"(
$first, $second
)"
* Creates a tuple of type [Pair] from this and [that].
* There is no meaning attached to values in this class, it can be used for any purpose.
* Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.
* An example of decomposing it into values:
* @sample samples.misc.Tuples.tripleDestructuring
* Represents a triad of values
* There is no meaning attached to values in this class, it can be used for any purpose.
* Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.
* An example of decomposing it into values:
* @sample samples.misc.Tuples.tripleDestructuring
*Annotations
* Represents a triad of values
* There is no meaning attached to values in this class, it can be used for any purpose.
* Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.
* An example of decomposing it into values:
* @sample samples.misc.Tuples.tripleDestructuring
*Annotations
* Represents a triad of values
* There is no meaning attached to values in this class, it can be used for any purpose.
* Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.
* An example of decomposing it into values:
* @sample samples.misc.Tuples.tripleDestructuring
*Annotations
* Represents a triad of values
* There is no meaning attached to values in this class, it can be used for any purpose.
* Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.
* An example of decomposing it into values:
* @sample samples.misc.Tuples.tripleDestructuring
*Annotations
private var hasNext: Boolean = if (step > 0) first <= last else first >= last
val private var step = step.toUInt() // use 2-complement math for negative steps
private var next = if (hasNext) first else finalElement
override fun hasNext(): Boolean = hasNext
override fun nextUInt(): UInt {
    val value = next
    if (value == finalElement) {
        if (!hasNext) throw kotlin.NoSuchElementException()
        hasNext = false
    } else {
        next += step
    }
    return value
}

/*
 * Copyright 2010-2019 JetBrains s.r.o.
 * Use of this source code is governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

/** An iterator over a sequence of values of type `UByte`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UByteIterator : Iterator<UByte> {
    override final fun next() = nextUByte()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextUByte(): UByte
}

/** An iterator over a sequence of values of type `UShort`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UShortIterator : Iterator<UShort> {
    override final fun next() = nextUShort()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextUShort(): UShort
}

/** An iterator over a sequence of values of type `UInt`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UIntIterator : Iterator<UInt> {
    override final fun next() = nextUInt()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextUInt(): UInt
}

/** An iterator over a sequence of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class ULongIterator : Iterator<ULong> {
    override final fun next() = nextULong()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextULong(): ULong
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// Auto-generated file. DO NOT EDIT!

package kotlin.ranges

import kotlin.internal.*

/**
 * A range of values of type `ULong`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public class ULongRange(start: ULong, endInclusive: ULong) : ULongProgression(start, endInclusive, 1), ClosedRange<ULong> {
    override val start: ULong get() = first
    override val endInclusive: ULong get() = last

    override fun contains(value: ULong): Boolean = first <= value && value <= last

    override fun isEmpty(): Boolean = if (step > 0) first > last else first < last

    override fun equals(other: Any?): Boolean =
        other is ULongRange && (isEmpty() && other.isEmpty() ||
                first == other.first && last == other.last &&
                overrun fun hashCodeAt: Int =
                        if (isEndInclusive) {1 shl 31} else value
                Override fun toString(): String = "$first..$last"

    companion object {
        /**
         * Creates ULongRange within the specified bounds of a closed range.
         * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the
specified [step]. In order to go backwards the [step] must be negative. * [step] must be greater than Long.MIN_VALUE and not equal to zero. */

public fun fromClosedRange(rangeStart: ULong, rangeEnd: ULong, step: Long): ULongProgression = ULongProgression(rangeStart, rangeEnd, step)

/**
 * An iterator over a progression of values of type `ULong`. * @property step the number by which the value is incremented on each step. * @property finalElement the last element. * private var hasNext: Boolean = if (step > 0) first <= last else first >= last
 */
private class ULongProgressionIterator(first: ULong, last: ULong, step: Long) : ULongIterator() {

    private val finalElement = last
    private var hasNext: Boolean = if (step > 0) first <= last else first >= last
    private val step = step.toULong() // use 2-complement math for negative steps
    private var next = if (hasNext) first else finalElement

    override fun hasNext(): Boolean = hasNext

    override fun nextULong(): ULong {
        val value = next
        if (value == finalElement) {
            if (!hasNext) throw kotlin.NoSuchElementException()
            hasNext = false
        } else {
            next += step
        }
        return value
    }
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.math

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun min(a: UInt, b: UInt): UInt {
    return minOf(a, b)
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun min(a: ULong, b: ULong): ULong {
    return minOf(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun max(a: UInt, b: UInt): UInt {
    return maxOf(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun max(a: ULong, b: ULong): ULong {
    return maxOf(a, b)
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmName("UNumbersKt")
package kotlin

/**
 * Counts the number of set bits in the binary representation of this [UInt] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.countOneBits(): Int = toInt().countOneBits()

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [UInt] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.countLeadingZeroBits(): Int = toInt().countLeadingZeroBits()

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UInt] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.countTrailingZeroBits(): Int = toInt().countTrailingZeroBits()

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [UInt] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.takeHighestOneBit(): UInt = toInt().takeHighestOneBit().toUInt()

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [UInt] number.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.takeLowestOneBit(): UInt = toInt().takeLowestOneBit().toUInt()

/**
 * Rotates the binary representation of this [UInt] number left by the specified [bitCount] number of bits.
 * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UInt.rotateLeft(bitCount: Int): UInt = toInt().rotateLeft(bitCount).toUInt()
the binary representation of this [UInt] number right by the specified [bitCount] number of bits.

`number.rotateRight(-n) == number.rotateLeft(n)`

Counts the number of set bits in the binary representation of this [ULong] number.

`number.rotateLeft(-n) == number.rotateRight(n)`

Counts the number of consecutive most significant bits that are zero in the binary representation of this [ULong] number.

`number.rotateLeft(-n) == number.rotateRight(n)`

Counts the number of consecutive least significant bits that are zero in the binary representation of this [ULong] number.

`number.rotateLeft(-n) == number.rotateRight(n)`

Counts the number of consecutive least significant bits that are zero in the binary representation of this [UByte] number.

`number.rotateLeft(-n) == number.rotateRight(n)`

Counts the number of set bits in the binary representation of this [UByte] number.

`number.rotateLeft(-n) == number.rotateRight(n)`

Counts the number of set bits in the binary representation of this [UByte] number.

Returns a number having a single bit set in the position of the most significant set bit of this [ULong] number.

Returns a number having a single bit set in the position of the least significant set bit of this [ULong] number.

Returns a number having a single bit set in the position of the most significant set bit of this [UByte] number.

Returns a number having a single bit set in the position of the least significant set bit of this [UByte] number.

this number is zero.

*\@SinceKotlin("1.3")\n*\@ExperimentalStdlibApi\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly
*\public inline fun UByte.takeHighestOneBit(): UByte = toInt().takeHighestOneBit().toUByte()\n*\n* \* Returns a number having a single bit set in the position of the least significant set bit of this [UByte] number;\n* \* or zero, if this number is zero.

*\@SinceKotlin("1.3")\n*\@ExperimentalStdlibApi\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly
*\public inline fun UByte.takeLowestOneBit(): UByte = toInt().takeLowestOneBit().toUByte()\n*\n* Rotates the binary representation of this [UByte] number left by the specified [bitCount] number of bits.\n* \* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n* \* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n* \`number.rotateLeft(-n) == number.rotateRight(n)`\n* \* Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally:\n* \`number.rotateLeft(n %= 8) == number.rotateLeft(n)`\n
*\@SinceKotlin("1.3")\n*\@ExperimentalStdlibApi\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly
*\public inline fun UByte.rotateLeft(bitCount: Int): UByte = toByte().rotateLeft(bitCount).toUByte()\n*\n* Rotates the binary representation of this [UByte] number right by the specified [bitCount] number of bits.\n* \* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n* \`number.rotateRight(-n) == number.rotateLeft(n)`\n* \* Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally:\n* \`number.rotateRight(n %= 8) == number.rotateRight(n)`\n
*\@SinceKotlin("1.3")\n*\@ExperimentalStdlibApi\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly
*\public inline fun UByte.rotateRight(bitCount: Int): UByte = toByte().rotateRight(bitCount).toUByte()\n*\n* Counts the number of set bits in the binary representation of this [UShort] number.\n
*\@SinceKotlin("1.3")\n*\@ExperimentalStdlibApi\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly
*\public inline fun UByte.rotateLeft(bitCount: Int): UByte = toByte().rotateLeft(bitCount).toUByte()\n*\n* Counts the number of consecutive most significant bits that are zero in the binary representation of this [UShort] number.\n
*\@SinceKotlin("1.3")\n*\@ExperimentalStdlibApi\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly
*\public inline fun UByte.rotateRight(bitCount: Int): UByte = toByte().rotateRight(bitCount).toUByte()\n*\n* Counts the number of consecutive least significant bits that are zero in the binary representation of this [UShort] number.\n
*\@SinceKotlin("1.3")\n*\@ExperimentalStdlibApi\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly
*\public inline fun UByte.rotateLeft(bitCount: Int): UByte = toByte().rotateLeft(bitCount).toUByte()\n*\n* Rotates the binary representation of this [UShort] number left by the specified [bitCount] number of bits.\n* \* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n* \* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n* \`number.rotateLeft(-n) == number.rotateRight(n)`\n* \* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally:\n* \`number.rotateLeft(n %= 16) == number.rotateLeft(n)`\n
*\@SinceKotlin("1.3")\n*\@ExperimentalStdlibApi\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly
*\public inline fun UByte.rotateRight(bitCount: Int): UByte = toByte().rotateRight(bitCount).toUByte()\n*\n* Rotates the binary representation of this [UShort] number right by the specified [bitCount] number of bits.\n* \* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n* \* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n* \`number.rotateRight(-n) == number.rotateLeft(n)`\n* \* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally:\n* \`number.rotateRight(n %= 16) == number.rotateRight(n)`
number.rotateRight(-n) == number.rotateLeft(n)

* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally

numberrotateRight(n) == number.rotateRight(n % 16)

SinceKotlin("1.3")@ExperimentalStdlibApi

ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

@SinceKotlin("1.3")
@ExperimentalStdlibApi

@ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

`number.rotateRight(-n) == number.rotateLeft(n)`

* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally

number.rotateRight(n) == number.rotateRight(n % 16)

SinceKotlin("1.3")@ExperimentalStdlibApi

ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

@SinceKotlin("1.3")
@ExperimentalStdlibApi

@ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

`number.rotateRight(-n) == number.rotateLeft(n)`

* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally

number.rotateRight(n) == number.rotateRight(n % 16)

SinceKotlin("1.3")@ExperimentalStdlibApi

ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

@SinceKotlin("1.3")
@ExperimentalStdlibApi

@ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

`number.rotateRight(-n) == number.rotateLeft(n)`

* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally

number.rotateRight(n) == number.rotateRight(n % 16)

SinceKotlin("1.3")@ExperimentalStdlibApi

ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

@SinceKotlin("1.3")
@ExperimentalStdlibApi

@ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

`number.rotateRight(-n) == number.rotateLeft(n)`

* Rotating by a multiple of [UShort.SIZEBITS] (16) returns the same number, or more generally

number.rotateRight(n) == number.rotateRight(n % 16)

SinceKotlin("1.3")@ExperimentalStdlibApi

ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

@SinceKotlin("1.3")
@ExperimentalStdlibApi

@ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly

public inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()

`number.rotateRight(-n) == number.rotateLeft(n)`

* Rotating by a multiple of [UShort.SIZEBITS] (16) returns the same number, or more generally

number.rotateRight(n) == number.rotateRight(n % 16)
as a signed [UByte] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUByte(): UByte = toUByteOrNull() ?: numberFormatError(this)\n\n/**
 * Parses the string as a signed [UByte] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentExcep
return null

val limit: UInt = UInt.MAX_VALUE
val start: Int
val firstChar: Char = this[0]
if (firstChar < '0') { if (length == 1 || firstChar != '+') return null
    start = 1
} else {
    if (start == 0)
        limit = limitForMaxRadix  // limit / 36
    var limitBeforeMul = limitForMaxRadix

    val uradix = radix.toUInt()
    var result = 0u
    for (i in start until length)
        val digit = digitOf(this[i], radix)
        if (digit < 0) return null
        if (result > limitBeforeMul) {
            if (limitBeforeMul == limitForMaxRadix) {
                limitBeforeMul = limit / uradix
                if (result > limitBeforeMul) {
                    return null
                }
            } else {
                return null
            }
        }
        result *= uradix
        val beforeAdding = result
        result += digit.toUInt()  // overflow has happened
        return result
    }

    // parses the string as an [ULong] number and returns the result
    // or `null` if the string is not a valid representation of a number.
    if (radix.toUInt() == 10)
        return result
    val limitForMaxRadix = 512409557603043100uL  // limit / 36
    var limitBeforeMul = limitForMaxRadix
    val uradix = radix.toULong()
    var result = 0uL
    for (i in start until length)
        val digit = digitOf(this[i], radix)
        if (digit < 0) return null
        if (result > limitBeforeMul) {
            if (limitBeforeMul == limitForMaxRadix) {
                limitBeforeMul = limit / uradix
                if (result > limitBeforeMul) {
                    return null
                }
            } else {
                return null
            }
        }
        result *= uradix
        val beforeAdding = result
        result += digit.toUInt()  // overflow has happened
        if (result < beforeAdding) return null
    }
    return result
}

/**
 * Parses the string as an [ULong] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toULongOrNull(): ULong? = toULongOrNull(radix = 10)

/**
 * Parses the string as an [ULong] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 *
 * @throws
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toULongOrNull(radix: Int): ULong? {
    checkRadix(radix)
    val length = this.length
    if (length == 0) return null

    val limit: ULong = ULong.MAX_VALUE
    val start: Int
    val firstChar = this[0]
    if (firstChar < '0') {
        if (length == 1 || firstChar != '+') return null
        start = 1
    } else {
        start = 0
    }

    val limitForMaxRadix = 119304647u  // limit / 36
    var limitBeforeMul = limitForMaxRadix
    val uradix = radix.toUInt()
    var result = 0u
    for (i in start until length)
        val digit = digitOf(this[i], radix)
        if (digit < 0) return null
        if (result > limitBeforeMul) {
            if (limitBeforeMul == limitForMaxRadix) {
                limitBeforeMul = limit / uradix
                if (result > limitBeforeMul) {
                    return null
                }
            } else {
                return null
            }
        }
        result *= uradix
        val beforeAdding = result
        result += digit.toUInt()  // overflow has happened
        if (result < beforeAdding) return null
    }
    return result
}

/**
 * Computes the sine of the angle [x] given in radians.
 */
@SinceKotlin("1.2")
@OptIn(ExperimentalUnsignedTypes::class)
public fun sin(x: UDouble): UDouble { return 0.0

    if (x < 0) return -sin(x)
    if (x >= PI) return sin(x - PI)
    if (x >= PI*2) return sin(x - PI*2)

    // optimize for multiples of PI
    var y = x
    var d = 0.0
    if (y >= PI) y -= PI
    if (y >= PI*2) y -= PI*2

    var n = 0
    while (y > 0) {
        if (y > PI/4) {
            d = d + (y - PI/4) / (y + PI/4)
        } else {
            d = d + (PI/4 - y) / (PI/4 + y)
        }
        n++

        y = y * y / 90.0
        if (n == 1) return x
        if (n == 2) return -sin(x)
        if (n == 3) return sin(x - PI)
        if (n == 4) return sin(x - PI*2)
    }
    return result
}

/**
 * Computes the sine of the angle [x] given in radians.
 */
@SinceKotlin("1.2")
@OptIn(ExperimentalUnsignedTypes::class)
public fun sin(x: UFloat): UFloat { return 0.0

    if (x < 0) return -sin(x)
    if (x >= PI) return sin(x - PI)
    if (x >= PI*2) return sin(x - PI*2)

    // optimize for multiples of PI
    var y = x
    var d = 0.0
    if (y >= PI) y -= PI
    if (y >= PI*2) y -= PI*2

    var n = 0
    while (y > 0) {
        if (y > PI/4) {
            d = d + (y - PI/4) / (y + PI/4)
        } else {
            d = d + (PI/4 - y) / (PI/4 + y)
        }
        n++

        y = y * y / 90.0
        if (n == 1) return x
        if (n == 2) return -sin(x)
        if (n == 3) return sin(x - PI)
        if (n == 4) return sin(x - PI*2)
    }
    return result
}

/**
 * Computes the cosine of the angle [x] given in radians.
 */
@SinceKotlin("1.2")
@OptIn(ExperimentalUnsignedTypes::class)
public fun cos(x: UDouble): UDouble { return 1.0

    if (x < 0) return cos(x)
    if (x >= PI) return cos(x - PI)
    if (x >= PI*2) return cos(x - PI*2)

    // optimize for multiples of PI
    var y = x
    var d = 0.0
    if (y >= PI) y -= PI
    if (y >= PI*2) y -= PI*2

    var n = 0
    while (y > 0) {
        if (y > PI/4) {
            d = d + (y - PI/4) / (y + PI/4)
        } else {
            d = d + (PI/4 - y) / (PI/4 + y)
        }
        n++

        y = y * y / 90.0
        if (n == 1) return x
        if (n == 2) return -sin(x)
        if (n == 3) return sin(x - PI)
        if (n == 4) return sin(x - PI*2)
    }
    return result
}

/**
 * Computes the cosine of the angle [x] given in radians.
 */
@SinceKotlin("1.2")
@OptIn(ExperimentalUnsignedTypes::class)
public fun cos(x: UFloat): UFloat { return 1.0

    if (x < 0) return cos(x)
    if (x >= PI) return cos(x - PI)
    if (x >= PI*2) return cos(x - PI*2)

    // optimize for multiples of PI
    var y = x
    var d = 0.0
    if (y >= PI) y -= PI
    if (y >= PI*2) y -= PI*2

    var n = 0
    while (y > 0) {
        if (y > PI/4) {
            d = d + (y - PI/4) / (y + PI/4)
        } else {
            d = d + (PI/4 - y) / (PI/4 + y)
        }
        n++

        y = y * y / 90.0
        if (n == 1) return x
        if (n == 2) return -sin(x)
        if (n == 3) return sin(x - PI)
        if (n == 4) return sin(x - PI*2)
    }
    return result
}
Open Source Used In DNAC Platform Antelope

Compute the logarithm of the value \([x]\) to the given \([base]\).

**Special cases:**
- \(\log(x, b)\) is `NaN` when \(x < 0\) or \(b \leq 0\) or \(b == 1.0\).
- \(\log(0.0, b)\) is `-Inf` for `b > 1` and `-Inf` for `b < 1`
- \(\log(b, 0.0)\) is `-Inf` for `b > 1` and `+Inf` for `b < 1`

`log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1`
- `log(0.0, b)` is `-Inf` for `b > 1` and
- `log(x, b)` is `NaN` when \(x < 0\) or \(b <= 0\) or \(b == 1.0\)
- `log(+Inf, +Inf)` is `-Inf` for `b > 1` and `+Inf` for `b < 1`
/*@SinceKotlin("1.2")*/public expect fun log(x: Double, base: Double): Double

/**
 * Computes the natural logarithm (base `E`) of the value [x].
 * Special cases:
 * - `ln(NaN)` is `NaN`
 * - `ln(0)` is `-Inf`
 */
@SinceKotlin("1.2")
public expect fun ln(x: Double): Double

/**
 * Computes the common logarithm (base 10) of the value [x].
 * Special cases:
 * - `log10(NaN)` is `NaN`
 * - `log10(0)` is `-Inf`
 */
@SinceKotlin("1.2")
public expect fun log10(x: Double): Double

/**
 * Computes the binary logarithm (base 2) of the value [x].
 * Special cases:
 * - `log2(NaN)` is `NaN`
 * - `log2(0)` is `-Inf`
 * - `log2(1)` is `0.0`
 * - `log2(Inf)` is `Inf`
 */
@SinceKotlin("1.2")
public expect fun log2(x: Double): Double

/**
 * Rounds the given value [x] to an integer towards positive infinity.
 */
@SinceKotlin("1.2")
public expect fun round(x: Double): Int

/**
 * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.
 */
@SinceKotlin("1.2")
public expect fun round(x: Double): Int

/**
 * Returns the absolute value of the given value [x].
 * Special case:
 * - `abs(NaN)` is `NaN`
 */
@SinceKotlin("1.2")
public expect val abs: Double

/**
 * Returns the sign of the given value:
 * - `-1.0` if the value is negative,
 * - `0.0` if the value is zero,
 * - `1.0` if the value is positive
 */
@SinceKotlin("1.2")
public expect val sign: Double

/**
 * Returns the ulp (unit in the last place) of this value.
 * Special cases:
 * - `ulp(NaN)` is `NaN`
 * - `ulp(0.0)` is `Double.MIN_VALUE`
 * - `ulp(Inf)` is `Inf`
 */
@SinceKotlin("1.2")
public expect val ulp: Double

/**
 * Returns the Double value nearest to
this value in direction of positive infinity.

* Returns the [Double] value nearest to this value in direction of negative infinity.

* Returns the [Double] value nearest to this value in direction from this value towards the value [to].

* Special cases:

  - `-x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`
  - `-x.nextTowards(x) == x`

* Rounds this [Double] value to the nearest integer and converts the result to [Int].

* Returns the nearest integer and converts the result to [Long].

* Ties are rounded towards positive infinity.

* Special cases:

  - `-x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`
  - `-x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`

* @throws IllegalArgumentException when this value is `NaN`

* @SinceKotlin("1.2")

* Rounds this [Double] value to the nearest integer and converts the result to [Long].

* Ties are rounded towards positive infinity.

* Special cases:

  - `-x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`
  - `-x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`

* @throws IllegalArgumentException when this value is `NaN`

* @SinceKotlin("1.2")

* Computes the sine of the angle [x] given in radians.

* Special cases:

  - `sin(NaN) == NaN`
  - `sin(+-Inf) == -Inf` when `abs(x) > 1`
  - `sin(0.0)` is `0.0`

* Computes the cosine of the angle [x] given in radians.

* Special cases:

  - `cos(NaN) == NaN`
  - `cos(+-Inf) == -Inf`

* Computes the tangent of the angle [x] given in radians.

* Special cases:

  - `tan(NaN) == NaN`
  - `tan(+-Inf) == -Inf`

* Computes the arc sine of the value [x]; the returned value is an angle in the range from `-PI/2` to `PI/2` radians.

* Special cases:

  - `asin(NaN) == NaN`
  - `asin(0.0) == 0.0`
  - `asin(+-Inf) == PI/2` when `abs(x) < 1`

* Computes the arc cosine of the value [x]; the returned value is an angle in the range from `0.0` to `PI` radians.

* Special cases:

  - `acos(NaN) == NaN`
  - `acos(0.0) == PI/2`
  - `acos(+-Inf) == 0.0`

* Computes the arc tangent of the value [x]; the returned value is an angle in the range from `-PI/2` to `PI/2` radians.

* Special cases:

  - `atan(NaN) == NaN`
  - `atan(0.0) == 0.0`
  - `atan(+-Inf) == PI/2` when `abs(x) > 1`

* Computes the hyperbolic sine of the value [x]; the returned value is an angle in the range from `-PI` to `PI` radians.

* Special cases:

  - `sinh(0.0) == 0.0`

* Computes the hyperbolic cosine of the value [x]; the returned value is an angle in the range from `-PI` to `PI` radians.

* Special cases:

  - `cosh(0.0) == 1.0`

* Computes the hyperbolic tangent of the value [x]; the returned value is an angle in the range from `-1.0` to `1.0` radians.

* Special cases:

  - `tanh(0.0) == 0.0`

* Computes the inverse hyperbolic sine of the value [x]; the returned value is positive `y` such that `sinh(y) == x` and `y` is positive.

* Special cases:

  - `asinh(0.0) == 0.0`

* Computes the inverse hyperbolic cosine of the value [x]; the returned value is positive `y` such that `cosh(y) == x` and `y` is positive.

* Special cases:

  - `acosh(0.0) == +Inf`

* Computes the inverse hyperbolic tangent of the value [x]; the returned value is an angle in the range from `-1.0` to `1.0` radians.

* Special cases:

  - `tanh(+-Inf) == +-Inf`

* Computes the inverse hyperbolic sine of the value [x]; the returned value is positive `y` such that `sin(y) == x`.
tangent of the value \[x\].\ln *\n* The returned value is ‘y’ such that \(\tan(h(y)) = x\).\ln *\n* Special cases:\n* - \(\tanh(NaNaN)\) is ‘NaNaN’\n* - \(\tan(h(x))\) is ‘NaNaN’ when \(x > 1\) or \(x < -1\)\n* - \(\tan(h(1.0))\) is ‘+Inf’\n* - \(\tan(h(-1.0))\) is ‘-Inf’\n* @SinceKotlin("1.2")\n
```kotlin
public fun atanh(x: Float): Float
```

```kotlin
/**
 * Computes \(\text{atanh}(x)\) without intermediate overflow or underflow.\ln *\n* Special cases:\n* - returns ‘+Inf’ if any of arguments is infinite\n* - returns ‘NaNaN’ if any of arguments is ‘NaNaN’ and the other is not infinite\n```
cases:
* - `b.pow(0.0)` is `1.0`
* - `b.pow(1.0)` is `b`
* - `b.pow(NaN)` is `NaN`
* - `NaN.pow(x)` is `NaN` for `x != 0.0`
* - `NaN.pow(x)` is `NaN` for `x < 0`
* - `NaN.pow(x)` is `NaN` for `x > 0` and `x` is finite and not an integer
* - `NaN` is `NaN` for `abs(b)` == `1.0` for `b < 0` and `x` is finite and not an integer

`@SinceKotlin("1.2")` public expect fun Float.pow(x: Float): Float

```kotlin
public expect fun Float.pow(n: Int): Float
```

```kotlin
public expect fun Float.pow(x: Float): Float

/**
 * Returns the absolute value of this value.
 * Special cases:
 * - `NaN.absoluteValue` is `NaN`
 * - `abs(b)` is `NaN` for `b < 0` and `x` is finite and not an integer
 */
@SinceKotlin("1.2")
public expect fun Float.pow(x: Float): Float

```

```
```
Open Source Used In DNAC Platform Antelope 8355
MAAA,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU,yB;M
MAAA,iB;MAAAM,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;MAAAM,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;MAAAM,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;MAAAM,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;MAAAM,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;MAAAM,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;MAAAM,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;MAAAM,0B;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
A,yB;MAAA,iB;QAOI,OAAO,MAAK,KAAM,CAAN,EAAiB,CAAjB,EA44B,CAAB5B,C;O;KAPhiB,C;mFAU
Open Source Used In DNAC Platform Antelope 8472
Open Source Used In DNAC Platform Antelope
1.2228 hbase-shaded-gson 3.4.1

1.2228.1 Available under license:

Apache HBase Relocated (Shaded) GSON Libs
Copyright 2017-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.2229 pinkie-promise 1.0.0 1.0.0

1.2229.1 Available under license:

The MIT License (MIT)

Copyright (c) Vsevolod Strukchinsky <floatdrop@gmail.com> (github.com/floatdrop)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2230 ip-regexp 1.0.3

1.2230.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2231 openssl 1.0.2m

1.2231.1 Available under license:

LICENSE ISSUES

===============

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License

-----------

/*==============================================================================
* Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.
* * Redistribution and use in source and binary forms, with or without
* * modification, are permitted provided that the following conditions
* * are met:
* * */
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
   "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:
   "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *       "This product includes cryptographic software written by
 *        Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *       "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 *
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA
02111-1307, USA.

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.
We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

Open Source Used In DNAC Platform Antelope 8532
3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program Gnomovision (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the
Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

   a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

   b) use the modified Package only within your corporation or organization.

   c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

   d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

   a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

   b) accompany the distribution with the machine-readable source of the Package with your modifications.

   c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together
with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
1.2232 @jsreport/mingo 2.4.1

1.2232.1 Available under license:

The MIT License (MIT)

Copyright (c) 2017 Francis Asante

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2233 fs-writefile-promise 1.0.3 1.0.3

1.2233.1 Available under license:

Copyright (c) 2015, Ahmad Nassri <ahmad@ahmadnassri.com>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.2234 openjpeg 1.6.35

1.2234.1 Available under license:

### MIT License

Copyright (c) 2017 by Jim Pattee <jimp03@email.com>.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2002-2014, Universite catholique de Louvain (UCL), Belgium
Copyright (c) 2002-2014, Professor Benoit Macq
Copyright (c) 2001-2003, David Janssens
Copyright (c) 2002-2003, Yannick Verschueren
Copyright (c) 2003-2007, Francois-Olivier Devaux and Antonin Descampe
Copyright (c) 2005, Herve Drolon, FreeImage Team
Copyright (c) 2007, Digital Signal Processing Laboratory, Universit degli studi di Perugia (UPG), Italy
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS `AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Little CMS
Copyright (c) 1998-2011 Marti Maria Saguer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

This code is released under the libpng license.

libpng versions 1.0.7, July 1, 2000 through 1.6.25, September 1, 2016 are
Copyright (c) 2000-2002, 2004, 2006-2016 Glenn Randers-Pehrson, are
derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Mans Rullgard
Cosmin Truta
Gilles Vollant
James Yu

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the
library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

Some files in the "scripts" directory have other copyright owners but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner
The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

END OF COPYRIGHT NOTICE, DISCLAIMER, and LICENSE.

TRADEMARK:

The name "libpng" has not been registered by the Copyright owner as a trademark in any jurisdiction. However, because libpng has been distributed and maintained world-wide, continually since 1995, the Copyright owner claims "common-law trademark protection" in any jurisdiction where common-law trademark is recognized.

OSI CERTIFICATION:

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative. OSI has not addressed the additional disclaimers inserted at version 1.0.7.

EXPORT CONTROL:

The Copyright owner believes that the Export Control Classification Number (ECCN) for libpng is EAR99, which means not subject to export controls or International Traffic in Arms Regulations (ITAR) because it is open source, publicly available software, that does not contain
any encryption software. See the EAR, paragraphs 734.3(b)(3) and
734.7(b).

Glenn Randers-Pehrson
glennrp at users.sourceforge.net
September 1, 2016
/*
 * The copyright in this software is being made available under the 2-clauses
 * BSD License, included below. This software may be subject to other third
 * party and contributor rights, including patent rights, and no such rights
 * are granted under this license.
 *
 * Copyright (c) 2002-2014, Universite catholique de Louvain (UCL), Belgium
 * Copyright (c) 2002-2014, Professor Benoit Macq
 * Copyright (c) 2003-2014, Antonin Descampe
 * Copyright (c) 2003-2009, Francois-Olivier Devaux
 * Copyright (c) 2005, Herve Drolon, FreeImage Team
 * Copyright (c) 2002-2003, Yannick Verschueren
 * Copyright (c) 2001-2003, David Janssens
 * Copyright (c) 2011-2012, Centre National d'Etudes Spatiales (CNES), France
 * Copyright (c) 2012, CS Systemes d'Information, France
 *
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS `AS IS'
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
1.2235 p-reduce 2.1.0
1.2235.1 Available under license:
   MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2236 kudu-client 1.14.0
1.2236.1 Available under license:
   Apache Kudu
   Copyright 2016 The Apache Software Foundation

   This product includes software developed at
   The Apache Software Foundation (http://www.apache.org/).

   Portions of this software were developed at
   Cloudera, Inc (http://www.cloudera.com/).

   This product includes software developed by the OpenSSL
   Project for use in the OpenSSL Toolkit (http://www.openssl.org/)

   This product includes cryptographic software written by Eric Young
   (eay@cryptsoft.com). This product includes software written by Tim
   Hudson (tjh@cryptsoft.com).

   Apache License
   Version 2.0, January 2004
   http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.

--------------------------------------------------------------------------------

src/kudu/gutil (some portions): Apache 2.0, and 3-clause BSD

Some portions of this module are derived from code in the Chromium project, 
copyright (c) Google inc and (c) The Chromium Authors and licensed under the 
Apache 2.0 License or the under the 3-clause BSD license:

Copyright (c) 2013 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without 
modification, are permitted provided that the following conditions are 
met:

  * Redistributions of source code must retain the above copyright 
    notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above 
    copyright notice, this list of conditions and the following disclaimer 
    in the documentation and/or other materials provided with the 
    distribution.
  * Neither the name of Google Inc. nor the names of its 
    contributors may be used to endorse or promote products derived from 
    this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT 
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR 
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT 
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, 
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT 
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, 
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY 
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT 
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE 
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

--------------------------------------------------------------------------------

src/kudu/gutil/utf: licensed under the following terms:

UTF-8 Library

The authors of this software are Rob Pike and Ken Thompson.
Copyright (c) 1998-2002 by Lucent Technologies.
Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR LUCENT TECHNOLOGIES MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

src/kudu/util (some portions): 3-clause BSD license

Some portions of this module are derived from code from LevelDB
( https://github.com/google/leveldb ):

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

src/kudu/util/array_view.h: 3-clause BSD license with patent grant
Copyright (c) 2015, The WebRTC project authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional IP Rights Grant (Patents)
------------------------------------

"This implementation" means the copyrightable works distributed by Google as part of the WebRTC code package.

Google hereby grants to you a perpetual, worldwide, non-exclusive, no-charge, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, transfer, and otherwise run, modify and propagate the contents of this implementation of the WebRTC code package, where such license applies only to those patent claims, both currently owned by Google and acquired in the future, licensable by Google that are necessarily infringed by this implementation of the WebRTC code package. This grant does not include claims that would be infringed only as a consequence of further modification of this implementation. If you or
your agent or exclusive licensee institute or order or agree to the
institution of patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that this
implementation of the WebRTC code package or any code incorporated
within this implementation of the WebRTC code package constitutes
direct or contributory patent infringement, or inducement of patent
infringement, then any patent rights granted to you under this License
for this implementation of the WebRTC code package shall terminate as
of the date such litigation is filed.

--------------------------------------------------------------------------------

src/kudu/common/zp7.cc: MIT license

ZP7 (Zach's Peppy Parallel-Prefix-Popcountin' PEXT/PDEP Polyfill)

Copyright (c) 2020 Zach Wegner

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

--------------------------------------------------------------------------------

www/d3.v2.js: BSD 3-clause license

Copyright (c) 2012, Michael Bostock
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* The name Michael Bostock may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL MICHAEL BOSTOCK BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

--------------------------------------------------------------------------------

www/epoch.*: MIT license

Copyright (c) 2014 Fastly, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

--------------------------------------------------------------------------------

www/jquery*.js: MIT license

Copyright 2005, 2014 jQuery Foundation and other contributors,
This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

java/kudu-client/: Some classes are derived from the AsyncHBase project under the following 3-clause BSD license:

Copyright (C) 2010-2012 The Async HBase Authors. All rights reserved. This file is part of Async HBase.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the StumbleUpon nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====================================================================
LICENSE ISSUES
================

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts.

OpenSSL License
------------

Copyright (c) 1998-2016 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
   "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written
permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:
   "This product includes software developed by the OpenSSL Project
   for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY
EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.

====================================================================
This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com). This product includes software written by Tim
Hudson (tjh@cryptsoft.com).

Original SSLeay License
-----------------------
Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an SSL implementation written
by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as
the following conditions are aheared to. The following conditions
apply to all code found in this distribution, be it the RC4, RSA,
lhash, DES, etc., code; not just the SSL code. The SSL documentation
included with this distribution is covered by the same copyright terms
except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in
the code are not to be removed.
If this package is used in a product, Eric Young should be given attribution
as the author of the parts of the library used.
This can be in the form of a textual message at program startup or
in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"
   The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:
   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

================================================================================

The following dependencies or pieces of incorporated source code have licenses under one of the following categories:

(a) do not require their license text to be re-distributed with binary distributions (e.g. Boost license)

(b) have no requirements about re-distributing their license text in either source or binary distributions (e.g. public domain)
(c) require inclusion of the license text in both source and binary distributions, but are used only at build-time and thus not incorporated in binary artifacts (e.g. BSD- or MIT-licensed build tooling)

(d) are the same Apache 2.0 license reproduced in its entirety above.

Therefore, we do not reproduce their licenses in their entirety in this file.

--------------------------------------------------------------------------------

.ycm_extra_conf.py: public domain
cmake_modules/FindGMock.cmake: MIT license, build-only
cmake_modules/FindProtobuf.cmake: BSD 3-clause license, build-only
src/kudu/util (HdrHistogram-related classes): public domain
src/kudu/gutil/valgrind.h: Hybrid BSD (half BSD, half zlib)
  - See the file headers for full license text.
build-support/iwyu (some portions): LLVM Release License.
  - See thirdparty/LICENSE.txt for full license text.

src/kudu/server/url-coding.cc (some portions): Boost software license

www/bootstrap: Apache 2.0 license
java/kudu-client/src/main/java/org/apache/kudu/util/{Slice,Slices}.java: Apache 2.0 license
java/gradlew: Apache 2.0 license
  - See above for full text.

================================================================================

Note on thirdparty dependencies downloaded during the build process

------------------------------------------------------------

The Kudu build process downloads many thirdparty dependencies automatically. Many of these dependencies are statically linked into the resulting Kudu binaries. Please refer to thirdparty/LICENSE.txt for relevant information.

**1.2237 domelementtype 1.3.1**

**1.2237.1 Available under license:**

Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
1.2238 semver-diff 3.1.1
1.2238.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2239 guava 30.1-jre
1.2239.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2010 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/ContiguousSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collection/SortedLists.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Strings.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Atomics.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Equivalence.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/Beta.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/UncaughtExceptionHandler.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

```
* Copyright (C) 2011 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
* Copyright (C) 2016 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*/
```

```
* Copyright (C) 2011 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
* Copyright (C) 2011 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/MoreCollectors.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/ImmutableValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/ElementOrder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/ValueGraphBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/Comparators.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/GraphConstants.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/GraphBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/UndirectedMultiNetworkConnections.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/graph/ForwardingValueGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
***/

* Holder for web specializations of methods of {@code Doubles}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permit/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/primitives/DoublesMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
***/

* Holder for web specializations of methods of {@code Ints}. Intended to be empty for regular
* version.
*/
Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/IntsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the
 * License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * either
 * express or implied. See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSortedAsList.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CollectSpliterators.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableBiMapFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/package-info.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Sets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Multimaps.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/BiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Multisets.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingConcurrentMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ReverseNaturalOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/NullsFirstOrdering.java
Open Source Used In DNAC Platform Antelope 8569
jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/MutableClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMapBasedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/UsingToStringOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Iterables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Lists.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ExplicitOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/LinkedHashSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ArrayListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ByFunctionOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ReverseOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/SetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/LexicographicalOrdering.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultiset.java
jar/com/google/common/collect/ForwardingCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Maps.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/ForwardingObject.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/MapDifference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/EnumBiMap.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/Resources.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Predicates.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Function.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Throwables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/Primitives.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/FinalizableWeakReference.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ExecutionList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2017 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/ImmutableDoubleArray.java
/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/ToDoubleRounder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/BigDecimalMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Java8Compatibility.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/ToDoubleRounder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/BigDecimalMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/Java8Compatibility.java

No license file was found, but licenses were detected in source scan.
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/Streams.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Internal.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
/*

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/LittleEndianByteArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/FarmHashFingerprint64.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/MacHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AggregateFutureState.java

No license file was found, but licenses were detected in source scan.

/*

* Copyright (C) 2018 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
* */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/JdkBackedImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/ImmutableSupplier.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultisetSupplier.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultisetIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultisetSpliterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultisetSpliteratorImpl.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultisetSpliteratorOverlapManager.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashMultisetSpliteratorOverlapManagerImpl.java
*/
jar/com/google/common/util/concurrent/ExecutionSequencer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/IgnoreJRERequirement.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

/*
* This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
* and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
*
* As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
* java.util.HashMap class.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

---

Open Source Used In DNAC Platform Antelope  8577
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/AbstractScheduledService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/cache/ForwardingLoadingCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/base/Optional.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/Ticker.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/Present.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/primitives/UnsignedLong.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/Types.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ForwardingExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/PairwiseEquivalence.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/TreeRangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/Absent.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/HashCode.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/net/MediaType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/Hasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/HashCode.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/net/MediaType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
Open Source Used In DNAC Platform Antelope 8580

jar/com/google/common/hash/AbstractStreamingHasher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/Uninterruptibles.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/Murmur3_128HashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/ Enums.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/math/MathPreconditions.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/net/HttpHeaders.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/RemovalCause.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/math/package-info.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/AsyncFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/Cache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ExecutionError.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/math/IntMath.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/EmptyContiguousSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the
 * License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 * express or implied. See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/**
 * Returns an array containing all of the elements in the specified collection. This method
 * returns the elements in the order they are returned by the collection's iterator. The returned
 * array is "safe" in that no references to it are maintained by the collection. The caller is
 * thus free to modify the returned array.
 *
 * <p>This method assumes that the collection size doesn't change while the method is running.
 *
 * <p>TODO(kevinb): support concurrently modified collections?
 *
 * @param c the collection for which to return an array of elements
 */
Found in path(s):
  */opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ObjectArrays.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 This method was rewritten in Java from an intermediate step of the Murmur hash function in
 http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */

Found in path(s):
  */opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2014 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
Open Source Used In DNAC Platform Antelope 8584

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/MoreObjects.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenerCallQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/math/Quantiles.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ListenerCallQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/TrustedListenableFuture.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ServiceManagerBridge.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Java8Usage.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2006 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeToken.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/CaseFormat.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/annotations/VisibleForTesting.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2013 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/HashingInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/VerifyException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/Runnables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Verify.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/io/CharSequenceReader.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/HashingInputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/web/Function.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/NetServerSocketFactory.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/transientobject/TransientObject.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/reflection/ParameterizedTypeWithActual.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ToIntFunction.java
*/
jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 * */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/JdkBackedImmutableBimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/BaseImmutableMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/JdkBackedImmutableBiMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/IndexedImmutableSet.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 * */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/RangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/TreeRangeMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/DescendingImmutableSortedSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/AbstractNavigableMap.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors.
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/hash/package-info.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 The Guava Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of { @code Objects} only in web. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
* This following method is a modified version of one found in
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
* which contained the following notice:
* 
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
* the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
* 
* Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/MoreExecutors.java
No license file was found, but licenses were detected in source scan.
/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableMapEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/MultimapBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/io/MoreFiles.java

No license file was found, but licenses were detected in source scan.

/*/  
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/Tables.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Serialization.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/StandardTable.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableListMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/PeekingIterator.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/RegularImmutableMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableMultiset.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/SingletonImmutableBiMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/RegularImmutableBiMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableBiMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableSortedSet.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableMapValues.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/Collections2.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableMapEntrySet.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableCollection.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/Table.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/EmptyImmutableListMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/CollectPreconditions.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/HashBasedTable.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableMultimap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/Platform.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/TreeBasedTable.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/StandardRowSortedTable.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/ImmutableMap.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
   jar/com/google/common/collect/Range.java
 * /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/UnmodifiableIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/ImmutableMapKeySet.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/**
 * Holder for web specializations of methods of [ @code Shorts]. Intended to be empty for regular
 * version.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/ImmutableRangeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/math/LinearTransformation.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/CharSink.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/LongAddable.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/SmoothRateLimiter.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/FilteredKeyMultimap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/Parameter.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/ByteSink.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/Closer.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/math/Stats.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/html/package-info.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/math/PairedStatsAccumulator.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/Element.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/CharSource.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/package-info.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/math/StatsAccumulator.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/escape/package-info.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/MutableTypeToInstanceMap.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/SipHashFunction.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/TypeCapture.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/LongAddable.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/ByteSource.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/ChecksumHashFunction.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/Invokable.java

* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/RateLimiter.java
*/
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/StandardSystemProperty.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/AbstractByteHasher.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/AbstractInvocationHandler.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/TypeToInstanceMap.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/math/PairedStats.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/LongAddables.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/hash/LongAddables.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/CartesianList.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/FileWriteMode.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect/ClassPath.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/xml/package-info.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/ImmutableRangeMap.java
*/opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ServiceManager.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/AbstractBaseGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/Traverser.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/ClosingFuture.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2019 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/graph/IncidentEdgeSet.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/collect/CompactHashing.java
No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2008 The Guava Authors
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
*/
* This method was rewritten in Java from an intermediate step of the Murmur hash function in
* http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
* following header:
*
* MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
* hereby disclaims copyright to this source code.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/Hashing.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
/*
**
* Outer class that exists solely to let us write [@code Partially.GwtIncompatible] instead of plain
* [@code GwtIncompatible]. This is more accurate for [@link Futures#catching], which is available
* under GWT but with a slightly different signature.
*<p>We can't use [@code PartiallyGwtIncompatible] because then the GWT compiler wouldn't recognize
* it as a [@code GwtIncompatible] annotation. And for [@code Futures.catching], we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT
* version.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/ImmutableGraph.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/SuccessorsFunction.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/ImmutableNetwork.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/io/InsecureRecursiveDeleteException.java
No license file was found, but licenses were detected in source scan.

/*
 * Written by Doug Lea with assistance from members of JCP JSR-166
 * Expert Group and released to the public domain, as explained at
 * http://creativecommons.org/publicdomain/zero/1.0/
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/util/concurrent/AtomicDoubleArray.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/hash/Striped64.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-

jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Doubles.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Joiner.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Chars.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/base/Converter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/common/escape/Escaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-jar/com/google/thirdparty/publicsuffix/TrieParser.java

No license file was found, but licenses were detected in source scan.
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
* /

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/MinMaxPriorityQueue.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/AbstractSequentialIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingListMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/RowSortedTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingSortedSetMultimap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ForwardingImmutableCollection.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/UnmodifiableListIterator.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/SortedMapDifference.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collection/SortedMultiset.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collection/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collection/DenseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/util/concurrent/ForwardingFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/net/HostSpecifier.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
  jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/reflect>TypeResolver.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/MapMaker.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/Cut.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/ByteArrayDataInput.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/ByteArrayDataOutput.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/html/HtmlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/xml/XmlEscapers.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/RegularImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/Callable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collect/RegularImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/io/ByteProcessor.java

Open Source Used In DNAC Platform Antelope 8603
jar/com/google/common/escape/ArrayBasedCharEscaper.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/net/InternetDomainName.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/base/Splitter.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/escape/Escapers.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
/**
 * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
 * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
 * version.
 *
 * @throws UnsupportedOperationException always
 *
 * @deprecated <b>Pass a key of type {@code Comparable} to use {@link
 *     ImmutableSortedMap#of(Comparable, Object)}.</b>
 */

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1.jar/com/google/common/collection/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2020 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 */
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of [@code Floats]. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/primitives/FloatsMethodsForWeb.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License);
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/ComputationException.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/DiscreteDomain.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/ImmutableClassToInstanceMap.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/ImmutableSortedSetFauxverideShim.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/ImmutableTable.java
* /opt/cola/permits/1119092615_1608716259.54/0/guava-30-1-jre-sources-1-
jar/com/google/common/collection/ImmutableAsList.java
1.2240 kind-of 4.0.0

1.2240.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2241 opentype-js 0.8.0
1.2241.1 Available under license:

The MIT License (MIT)

Copyright (c) 2017 Frederik De Bleser

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2242 parquet-hadoop 1.8.1
1.2242.1 Available under license:

Apache Parquet Hadoop
Copyright 2015 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.2243 babel-helper-hoist-variables 6.24.1

6.24.1

1.2243.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2244 libjpeg-turbo 1.4.2-0ubuntu3.3

1.2244.1 Available under license:

libjpeg-turbo Licenses

libjpeg-turbo is covered by three compatible BSD-style open source licenses:

- The IJG (Independent JPEG Group) License, which is listed in [README.ijg](README.ijg)

This license applies to the libjpeg API library and associated programs (any code inherited from libjpeg, and any modifications to that code.)

- The Modified (3-clause) BSD License, which is listed below

This license covers the TurboJPEG API library and associated programs, as well as the build system.
- The [zlib License](https://opensource.org/licenses/Zlib)

This license is a subset of the other two, and it covers the libjpeg-turbo SIMD extensions.

Complying with the libjpeg-turbo Licenses
=========================================

This section provides a roll-up of the libjpeg-turbo licensing terms, to the best of our understanding.

1. If you are distributing a modified version of the libjpeg-turbo source, then:

   1. You cannot alter or remove any existing copyright or license notices from the source.

      **Origin**
      - Clause 1 of the IJG License
      - Clause 1 of the Modified BSD License
      - Clauses 1 and 3 of the zlib License

   2. You must add your own copyright notice to the header of each source file you modified, so others can tell that you modified that file (if there is not an existing copyright header in that file, then you can simply add a notice stating that you modified the file.)

      **Origin**
      - Clause 1 of the IJG License
      - Clause 2 of the zlib License

   3. You must include the IJG README file, and you must not alter any of the copyright or license text in that file.

      **Origin**
      - Clause 1 of the IJG License

2. If you are distributing only libjpeg-turbo binaries without the source, or if you are distributing an application that statically links with libjpeg-turbo, then:

   1. Your product documentation must include a message stating:

      This software is based in part on the work of the Independent JPEG Group.

      **Origin**
2. If your binary distribution includes or uses the TurboJPEG API, then your product documentation must include the text of the Modified BSD License (see below.)

**Origin**
- Clause 2 of the Modified BSD License

3. You cannot use the name of the IJG or The libjpeg-turbo Project or the contributors thereof in advertising, publicity, etc.

**Origin**
- IJG License
- Clause 3 of the Modified BSD License

4. The IJG and The libjpeg-turbo Project do not warrant libjpeg-turbo to be free of defects, nor do we accept any liability for undesirable consequences resulting from your use of the software.

**Origin**
- IJG License
- Modified BSD License
- zlib License

The Modified (3-clause) BSD License
__________________________________________

Copyright (C)2009-2020 D. R. Commander. All Rights Reserved.
Copyright (C)2015 Viktor Szathmry. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the libjpeg-turbo Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS", AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE
Why Three Licenses?
===================

The zlib License could have been used instead of the Modified (3-clause) BSD License, and since the IJG License effectively subsumes the distribution conditions of the zlib License, this would have effectively placed libjpeg-turbo binary distributions under the IJG License. However, the IJG License specifically refers to the Independent JPEG Group and does not extend attribution and endorsement protections to other entities. Thus, it was desirable to choose a license that granted us the same protections for new code that were granted to the IJG for code derived from their software.

1.2245 hoek 2.16.3

1.2245.1 Available under license:

Copyright (c) 2011-2014, Walmart and other contributors.
Copyright (c) 2011, Yahoo Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/hapi/graphs/contributors
Portions of this project were initially based on the Yahoo! Inc. Postmile project,

1.2246 cookie-signature 1.0.6
1.2246.1 Available under license:
No license file was found, but licenses were detected in source scan.

```
# cookie-signature

Sign and unsign cookies.

```js

```js
var cookie = require('cookie-signature');

var val = cookie.sign('hello', 'tobiiscool');
val.should.equal('hello.DGDUkGlIkCzPz+C0B064FNgHdEjox7ch8tOBGslZ5QI');

var val = cookie.sign('hello', 'tobiiscool');
cookie.unsign(val, 'tobiiscool').should.equal('hello');
cookie.unsign(val, 'luna').should.be.false;
```

```
## License

(The MIT License)

Copyright (c) 2012 LearnBoost &lt;tj@learnboost.com&gt;

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be

Open Source Used In DNAC Platform Antelope  8616
The software is provided 'as is', without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

1.2247 growl 1.10.5

1.2247.1 Available under license:

No license file was found, but licenses were detected in source scan.

(The MIT License)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to the above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

1.2248 oas-schema-walker 1.1.5 1.1.5

1.2248.1 Available under license:

BSD 3-Clause License

Copyright (c) 2016, Mermade Software
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2249 shimmer 1.0.0 1.0.0

1.2249.1 Available under license:

Copyright 2013, the Dart project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2250 babel-plugin-polyfill-regenerator 0.1.6
1.2250.1 Available under license:
MIT License

Copyright (c) 2014-present Nicol Ribaudo and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2251 hk2-locator 2.2.0-b21
1.2251.1 Available under license:
Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 2010-2011 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in '* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'
Found license 'General Public License 2.0' in 'Copyright (c) 2010-2011 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License.'

Found license 'General Public License 2.0' in '* Copyright (c) 2012 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License.'

1.2252 y18n 4.0.0
1.2252.1 Available under license :
Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2253 oas-validator 5.0.5 5.0.5
1.2253.1 Available under license :
BSD 3-Clause License

Copyright (c) 2016, Mermade Software
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2254 joda-time 2.9.9
1.2254.1 Available under license:

=============================================================================  
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =  
=============================================================================  

This product includes software developed by
Joda.org (http://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2255 netty-transport-native-unix-common

4.1.42.Final

1.2255.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/io/netty/channel/unix/ErrorsStaticallyReferencedJniMethods.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/netty_unix_limits.c
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/io/netty/channel/unix/LimitsStaticallyReferencedJniMethods.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/netty_unix_util.h
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/io/netty/channel/unix/SocketWritableByteChannel.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/io/netty/channel/unix/PeerCredentials.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/io/netty/channel/unix/Limits.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/netty_unix_limits.h
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3-
  jar/netty_unix_util.c

No license file was found, but licenses were detected in source scan.

~ Copyright 2016 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
Found in path(s):
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4.1.42-final-sources-3-
jar/META-INF/maven/io.netty/netty-transport-native-unix-common/pom.xml
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 *   http://www.apache.org/licenses/LICENSE-2.0
 * *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4.1.42-final-sources-3-
jar/io/netty/channel/unix/package-info.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4.1.42-final-sources-3-
jar/io/netty/channel/unix/UnixChannelOption.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4.1.42-final-sources-3-
jar/io/netty/channel/unix/IovArray.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 *   http://www.apache.org/licenses/LICENSE-2.0
 * *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4.1.42-final-sources-3-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3.jar/io/netty/channel/unix/Buffer.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3.jar/io/netty/channel/unix/PreferredDirectByteBufAllocator.java
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3.jar/netty_unix_buffer.h
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3.jar/netty_unix_buffer.c
No license file was found, but licenses were detected in source scan.

*/
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*  
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1132244357_1613132141.63/0/netty-transport-native-unix-common-4-1-42-final-sources-3.jar/netty_unix_jni.h

1.2256 process-on-spawn 1.0.0

1.2256.1 Available under license:
MIT License

Copyright (c) 2019 CFWare, LLC

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2257 is-property 1.0.2

1.2257.1 Available under license :
No license file was found, but licenses were detected in source scan.

(c) 2013 Mikola Lysenko. MIT License

Found in path(s):
* /opt/cola/permits/1127084563_1611780084.51/0/fis-components-is-property-1-0-2-0-g24ebed6-tar-gz/fis-components-is-property-4515de4/README.md

1.2258 reactive-streams 1.0.2

1.2258.1 Available under license :
No license file was found, but licenses were detected in source scan.

/************************************************************************
* Licensed under Public Domain (CC0)                                 *
*                                                                       *
* To the extent possible under law, the person who associated CC0 with *
* this code has waived all copyright and related or neighboring        *
* rights to this code.                                                 *
*                                                                       *
* You should have received a copy of the CC0 legalcode along with this *
* work. If not, see <http://creativecommons.org/publicdomain/zero/1.0/>.
************************************************************************/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092915250_1600810885.15/0/reactive-streams-1-0-2-sources-1-jar/org/reactivestreams/Subscriber.java
* /opt/ws_local/PERMITS_SQL/1092915250_1600810885.15/0/reactive-streams-1-0-2-sources-1-jar/org/reactivestreams/Subscription.java
1.2259 batik-rasterizer 1.13

1.2259.1 Available under license:
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1146690630_1616599309.0/0/batik-rasterizer-1-13-sources-1-jar/META-INF/maven/org.apache.xmlgraphics/batik-rasterizer/pom.xml
No license file was found, but licenses were detected in source scan.

// ------------------------------------------------------------------------
// // Licensed to the Apache Software Foundation (ASF) under one or more
// // contributor license agreements. See the NOTICE file distributed with
// // this work for additional information regarding copyright ownership.
// // The ASF licenses this file to You under the Apache License, Version 2.0
// // (the "License"); you may not use this file except in compliance with
// // the License. You may obtain a copy of the License at
// // http://www.apache.org/licenses/LICENSE-2.0
// // Unless required by applicable law or agreed to in writing, software
// // distributed under the License is distributed on an "AS IS" BASIS,
// // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// // See the License for the specific language governing permissions and
// // limitations under the License.

Open Source Used In DNAC Platform Antelope 8631
grant codeBase "${app.jar.base}/xml-apis-ext-${xmlapis.version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/xml-apis-${xmlapis.version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/xalan-${xalan.version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/xercesImpl-2.9.1.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/../batik-rasterizer-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-svgrasterizer-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-ext-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-dom-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-css-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-svg-dom-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-gvt-${version}.jar" {
    permission java.security.AllPermission;
};
grant codeBase "${app.jar.base}/batik-parser-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-script-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-bridge-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-swing-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-transcoder-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-gui-util-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-awt-util-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-util-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-xml-${version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/fop-transcoder-allinone-${xmlgraphics.fop.version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/xmlgraphics-commons-${xmlgraphics.commons.version}.jar" {
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-anim-${version}.jar" {
    permission java.security.AllPermission;
};
grant codeBase "${app.jar.base}/batik-codec-${version}.jar" { 
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-extension-${version}.jar" { 
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/batik-svggen-${version}.jar" { 
    permission java.security.AllPermission;
};

grant codeBase "${app.jar.base}/js.jar" { 
    permission java.lang.RuntimePermission "createClassLoader";
    permission java.net.SocketPermission "*", "listen, connect, resolve, accept";
    permission java.lang.RuntimePermission "accessDeclaredMembers";
    permission java.util.PropertyPermission "org.mozilla.javascript.JavaAdapter", "read";
    permission java.util.PropertyPermission "org.mozilla.javascript.JavaAdapterClassName", "read";
    permission java.io.FilePermission "<<ALL FILES>>", "read";
};

grant {
    permission java.io.FilePermission "${app.jar.base}/batik-svg-dom-${version}.jar", "read";
};

Found in path(s):
* /opt/cola/permits/1146690630_1616599309.0/0/batik-rasterizer-1-13-sources-1-
jar/org/apache/batik/apps/rasterizer/resources/rasterizer.policy

1.2260 hbase-shaded-netty 3.4.1
1.2260.1 Available under license :

Apache HBase Relocated (Shaded) Netty Libs
Copyright 2017-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

    Apache License
    Version 2.0, January 2004
    http://www.apache.org/licenses/

    TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

    1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
1.2261 stream-combiner 0.0.4 0.0.4

1.2261.1 Available under license:

Copyright (c) 2012 'Dominic Tarr'

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2262 sshpk 1.16.1

1.2262.1 Available under license:

Copyright Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
1.2263 node-environment-flags 1.0.6

1.2263.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2264 y18n 3.2.1

1.2264.1 Available under license:

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright notice
and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF
THIS SOFTWARE.

1.2265 object-inspect 1.8.0
1.2265.1 Available under license :
MIT License

Copyright (c) 2013 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2266 form-data 2.3.3 2.3.3
1.2266.1 Available under license :
Copyright (c) 2012 Felix Geisendrfer (felix@debuggable.com) and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2267 dot-prop 5.3.0
1.2267.1 Available under license : MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2268 escalade 3.1.1 3.1.1
1.2268.1 Available under license : MIT License

Copyright (c) Luke Edwards <luke.edwards05@gmail.com> (lukeed.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2269 string-width 4.2.2

1.2269.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.


1.2270.1 Available under license :

(The MIT License)

Copyright (C) 2011-2015 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2271 javax.servlet.jsp 2.3.2
1.2271.1 Available under license:

Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 1997-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License <bottom> Copyright (c) 1999-2012 Oracle and/or its affiliates. All Rights Reserved. Use is subject to license terms. </bottom>"

Found license 'General Public License 2.0' in '# Copyright (c) 1997-2010 Oracle and/or its affiliates. All rights reserved. # The contents of this file are subject to the terms of either the GNU # General Public License Version 2 only ("GPL") or the Common Development # and Distribution License("CDDL") (collectively, the "License"). You # may not use this file except in compliance with the License. You can # Oracle designates this particular file as subject to the "Classpath" # exception as provided by Oracle in the GPL Version 2 section of the License # Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. # You may obtain a copy of the License at # http://www.apache.org/licenses/LICENSE-2.0 # distributed under the License is distributed on an "AS IS" BASIS,'

Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 1997-2010 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in 'Copyright (c) 1997-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License'

Found license 'General Public License 2.0' in '* Copyright (c) 1997-2012 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2 only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You * may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License.
Found license 'General Public License 2.0' in Copyright (c) 1997-2010 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the License). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License.
Found license 'General Public License 2.0' in Copyright (c) 1997-2013 Oracle and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the License). You may not use this file except in compliance with the License. You can Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License.
Licensed under the Apache License, Version 2.0 (the License); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 distributed under the License is distributed on an "AS IS" BASIS,

1.2272 fontbox 2.0.22
1.2272.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a ”NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions.
for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

CONTRIBUTIONS TO THE ORIGINAL CODEBASE

Apache FontBox is based on contributions made to the original FontBox project:

Copyright (c) 2006-2007, www.fontbox.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of fontbox; nor the names of its contributors may be used to endorse or promote products derived from this software without
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Lohit-Bengali font (https://pagure.io/lohit):

Copyright 2011-13 Lohit Fonts Project contributors
<http://fedorahosted.org/lohit>

This Font Software is licensed under the SIL Open Font License, Version 1.1.
This license is copied below, and is also available with a FAQ at:
http://scripts.sil.org/OFL

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide
development of collaborative font projects, to support the font creation
efforts of academic and linguistic communities, and to provide a free and
open framework in which fonts may be shared and improved in partnership
with others.

The OFL allows the licensed fonts to be used, studied, modified and
redistributed freely as long as they are not sold by themselves. The
fonts, including any derivative works, can be bundled, embedded,
redistributed and/or sold with any software provided that any reserved
names are not used by derivative works. The fonts and derivatives,
however, cannot be released under any other type of license. The
requirement for fonts to remain under this license does not apply
to any document created using the fonts or their derivatives.

DEFINITIONS
"Font Software" refers to the set of files released by the Copyright
Holder(s) under this license and clearly marked as such. This may
include source files, build scripts and documentation.
"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS
Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.
TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Apache FontBox
Copyright 2008-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Based on source code contributed to the original FontBox project.
Copyright (c) 2006-2007, www.fontbox.org

Includes the Script Property (Scripts-10.0.0.txt)
Copyright 2017 Unicode, Inc.

1.2273 oauth-sign 0.9.0
1.2273.1 Available under license :
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of
such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2274 jline-terminal-jna 3.12.1

1.2274.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2016, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/solaris/SolarisNativePty.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/JnaNativePty.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/win/WindowsAnsiWriter.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/osx/OsXNativePty.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/freebsd/CLibrary.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/osx/CLibrary.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/linux/CLibrary.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/solaris/CLibrary.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/freebsd/FreeBsdNativePty.java
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/org/jline/terminal/implement/jna/linux/LinuxNativePty.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2017, the original author or authors.
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-
  jar/org/jline/terminal/impl/jna/win/JnaWinConsoleWriter.java
  No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2018, the original author or authors.
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-
  jar/org/jline/terminal/impl/jna/win/Kernel32.java
  No license file was found, but licenses were detected in source scan.

2016, the original author or authors.

This software is distributable under the BSD license. See the terms of the
BSD license in the documentation provided with this software.

https://opensource.org/licenses/BSD

Found in path(s):
* /opt/cola/permits/1114600364_1607534022.33/0/jline-terminal-jna-3-12-1-sources-jar/META-INF/maven/org.jline/jline-terminal-jna/pom.xml
  No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2019, the original author or authors.
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 */
1.2275 fresh 0.3.0 0.3.0

1.2275.1 Available under license :
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2276 jsr305 3.0.2

1.2276.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2005 Brian Goetz
 * Released under the Creative Commons Attribution License
 * (http://creativecommons.org/licenses/by/2.5)
 * Official home: http://www.jcip.net
 */

Found in path(s):
1.2277 ignore 3.3.10

1.2277.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2278 finalhandler 0.4.1

1.2278.1 Available under license:

(The MIT License)

Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND.
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2279 libbsd 0.8.2 1ubuntu0.1

1.2279.1 Available under license:
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/

Files:
*
Copyright:
Copyright 2004-2006, 2008-2015 Guillem Jover <guillem@hadrons.org>
License: BSD-3-clause

Files:
man/arc4random.3
man/tree.3
Copyright:
Copyright 1997 Niels Provos <provos@physnet.uni-hamburg.de>
All rights reserved.
License: BSD-4-clause-Niels-Provos
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software
   must display the following acknowledgement:
      This product includes software developed by Niels Provos.
4. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:
man/getprogname.3
Copyright:
Copyright 2001 Christopher G. Demetriou
All rights reserved.
License: BSD-4-clause-Christopher-G-Demetriou
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software
   must display the following acknowledgement:
      This product includes software developed for the
      NetBSD Project. See http://www.netbsd.org/ for
      information about NetBSD.
4. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:
include/bsd/err.h
include/bsd/stdlib.h
include/bsd/unistd.h
src/bsd_getopt.c
src/err.c
src/fgetln.c
src/progname.c
Copyright:
Copyright 2005, 2008-2012 Guillem Jover <guillem@hadrons.org>
Copyright 2005 Hector Garcia Alvarez
Copyright 2005 Aurelien Jarno
Copyright  2006 Robert Millan
License: BSD-3-clause

Files:
include/bsd/netinet/ip_icmp.h
include/bsd/sys/bitstring.h
include/bsd/sys/queue.h
include/bsd/timeconv.h
include/bsd/vis.h
man/bitstring.3
man/explicit_bzero.3
man/fgetln.3
man/fgetwln.3
man/funopen.3bsd
man/getbsize.3
man/heapsort.3
man/nlist.3
man/queue.3bsd
man/radixsort.3
man/reallocarray.3
man/reallocf.3
man/setmode.3
man/strmode.3
man/strnstr.3
man/unvis.3
man/vis.3
man/wcslepy.3
src/getbsize.c
src/heapsort.c
src/merge.c
src/nlist.c
src/radixsort.c
src/setmode.c
src/strmode.c
src/strnstr.c
src/unvis.c
src/vis.c

Copyright:  
    The Regents of the University of California.  All rights reserved.
Copyright  2001 Mike Barcroft <mike@FreeBSD.org>

Some code is derived from software contributed to Berkeley by the
American National Standards Committee X3, on Information Processing Systems.

Some code is derived from software contributed to Berkeley by Peter McIlroy.
Some code is derived from software contributed to Berkeley by Ronnie Kon at Mindcraft Inc., Kevin Lew and Elmer Yglesias.

Some code is derived from software contributed to Berkeley by Dave Borman at Cray Research, Inc.

Some code is derived from software contributed to Berkeley by Paul Vixie.

Some code is derived from software contributed to Berkeley by Chris Torek.

Copyright UNIX System Laboratories, Inc.
All or some portions of this file are derived from material licensed to the University of California by American Telephone and Telegraph Co. or Unix System Laboratories, Inc. and are reproduced herein with the permission of UNIX System Laboratories, Inc.
License: BSD-3-clause-Regents
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS `AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:
include/bsd/libutil.h
Copyright:
Copyright 1996 Peter Wemm <peter@FreeBSD.org>.
All rights reserved.
Copyright 2002 Networks Associates Technology, Inc.
All rights reserved.
License: BSD-3-clause-Peter-Wemm
Redistribution and use in source and binary forms, with or without
modification, is permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior written
   permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS \``AS IS\'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Files:
man/setproctitle.3
Copyright:
Copyright 1995 Peter Wemm <peter@FreeBSD.org>
All rights reserved.
License: BSD-5-clause-Peter-Wemm
Redistribution and use in source and binary forms, with or without
modification, is permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice immediately at the beginning of the file, without modification,
   this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. This work was done expressly for inclusion into FreeBSD. Other use
   is permitted provided this notation is included.
4. Absolutely no warranty of function or purpose is made by the author
   Peter Wemm.
5. Modifications may be freely made to this file providing the above
   conditions are met.
Files:
include/bsd/stringlist.h
man/fmtcheck.3
man/humanize_number.3
man/stringlist.3
src/fmtcheck.c
src/humanize_number.c
src/stringlist.c

Copyright:
All rights reserved.

Some code was contributed to The NetBSD Foundation by Allen Briggs.

Some code was contributed to The NetBSD Foundation by Luke Mewburn.


Some code is derived from software contributed to The NetBSD Foundation by Julio M. Merino Vidal, developed as part of Google's Summer of Code 2005 program.

Some code is derived from software contributed to The NetBSD Foundation by Christos Zoulas.

License: BSD-2-clause-NetBSD

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Files:
include/bsd/sys/endian.h
man/expand_number.3
man/closefrom.3
man/flopen.3
man/getpeereid.3
man/pidfile.3
src/expand_number.c
src/hash/sha512.h
src/hash/sha512c.c
src/pidfile.c
src/reallocf.c
src/timeconv.c
Copyright:
Copyright 1998, M. Warner Losh <imp@freebsd.org>
All rights reserved.

Copyright 2001 Dima Dorfman.
All rights reserved.

Copyright 2001 FreeBSD Inc.
All rights reserved.

Copyright 2002 Thomas Moestl <tmm@FreeBSD.org>
All rights reserved.

Copyright 2005 Pawel Jakub Dawidek <pjd@FreeBSD.org>
All rights reserved.

Copyright 2005 Colin Percival
All rights reserved.

Copyright 2007 Eric Anderson <anderson@FreeBSD.org>
Copyright 2007 Pawel Jakub Dawidek <pjd@FreeBSD.org>
All rights reserved.

Copyright 2007 Dag-Erling Codan Smrgrav
All rights reserved.

Copyright 2009 Advanced Computing Technologies LLC
Written by: John H. Baldwin <jhb@FreeBSD.org>
All rights reserved.

Copyright 2011 Guillem Jover <guillem@hadrons.org>
License: BSD-2-clause
Copyright:
Copyright 2007 Dag-Erling Codan Smrgrav
All rights reserved.
License: BSD-2-clause-verbatim
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer
   in this position and unchanged.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:
include/bsd/sys/tree.h
man/fparseln.3
src/fparseln.c
Copyright:
Copyright 1997 Christos Zoulas.
All rights reserved.

Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.
License: BSD-2-clause-author
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Files:
include/bsd/readpassphrase.h
man/readpassphrase.3
man/strlcpy.3
man/strtonum.3
src/arc4random.c
src/arc4random_openbsd.h
src/arc4random_uniform.c
src/arc4random_unix.h
src/closefrom.c
src/getentropy_aix.c
src/getentropy_bsd.c
src/getentropy_hpux.c
src/getentropy_hurd.c
src/getentropy_linux.c
src/getentropy_osx.c
src/getentropy_solaris.c
src/readpassphrase.c
src/reallocarray.c
src/strlcat.c
src/strlcpy.c
src/strtonum.c
Copyright:
Copyright 2004 Ted Unangst and Todd Miller
All rights reserved.

Copyright 1996 David Mazieres <dm@uun.org>
   Todd C. Miller <Todd.Miller@courtesan.com>
Copyright 2004 Ted Unangst
Copyright 2008 Damien Miller <djm@openbsd.org>
Copyright 2008 Otto Moerbeek <otto@drijf.net>
Copyright 2013 Markus Friedl <markus@openbsd.org>
Copyright 2014 Bob Beck <beck@obtuse.com>
Copyright 2014 Brent Cook <bcook@openbsd.org>
Copyright 2014 Pawel Jakub Dawidek <pjd@FreeBSD.org>
Copyright 2014 Theo de Raadt <deraadt@openbsd.org>
Copyright 2015 Michael Felt <aixtools@gmail.com>
Copyright 2015 Guillem Jover <guillem@hadrons.org>
License: ISC
Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:
src/inet_net_pton.c
Copyright:
Copyright 1996 by Internet Software Consortium.
License: ISC-Original
Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Files:
src/setproctitle.c
Copyright:
Copyright 2010 William Ahern
Copyright 2012 Guillem Jover <guillem@hadrons.org>
License: Expat
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Files:
include/bsd/md5.h
src/hash/md5.c
Copyright:
None
License: public-domain-Colin-Plumb
This code implements the MD5 message-digest algorithm.
The algorithm is due to Ron Rivest. This code was written by Colin Plumb in 1993, no copyright is claimed.
This code is in the public domain; do with it what you wish.

Files:
src/explicit_bzero.c
src/chacha_private.h
Copyright:
None
License: public-domain
Public domain.

Files:
man/mdX.3bsd
src/hash/md5hl.c
src/hash/helper.c
Copyright:
None
License: Beerware
"THE BEER-WARE LICENSE" (Revision 42):
<phk@login.dkuug.dk> wrote this file. As long as you retain this notice you can do whatever you want with this stuff. If we meet some day, and you think this stuff is worth it, you can buy me a beer in return. Poul-Henning Kamp

License: BSD-3-clause
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: BSD-2-clause
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

1.2280 json-refs 3.0.15 3.0.15
1.2280.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Jeremy Whitlock

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2281 lodash-pick 4.4.0
1.2281.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code displayed within the prose of the
documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own
licenses; we recommend you read them, as their terms may differ from the
terms above.

1.2282 hpkp 1.2.0
1.2282.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

/*
 * Copyright (C) 2009 The JSR-330 Expert Group
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

1.2284 reftools 1.1.8 1.1.8

1.2284.1 Available under license:

BSD 3-Clause License

Copyright (c) 2017, Mike Ralphson
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.
* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2285 destroy 1.0.4 1.0.4

1.2285.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2286 iconv-lite 0.4.23 0.4.23
1.2286.1 Available under license:
Copyright (c) 2011 Alexander Shtuchkin

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2287 deepmerge 2.1.0

1.2287.1 Available under license:
The MIT License (MIT)

Copyright (c) 2012 Nicholas Fisher

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.2288 pre-push 0.1.1 0.1.1

1.2288.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Arnout Kazemier, Martijn Swaagman, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2289 asm-all-repackaged 2.2.0-b21

1.2289.1 Available under license:

No license file was found, but licenses were detected in source scan.

ASM XML Adapter
Copyright (c) 2004, Eugene Kuleshov
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/xml/asm-xml.dtd
No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2005 INRIA, France Telecom
* All rights reserved.
* *
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
* *
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
No license file was found, but licenses were detected in source scan.

2011 Oracle and/or its affiliates. All rights reserved.

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can obtain a copy of the License at https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html or packager/legal/LICENSE.txt. See the License for the specific language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each file and include the License file at packager/legal/LICENSE.txt.

GPL Classpath Exception:
Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or only the GPL Version 2, indicate your decision by adding "[Contributor] elects to include this software in this distribution under the [CDDL or GPL Version 2] license." If you don't indicate a single choice of license, a recipient has the option to distribute your version of this file under either the CDDL, the GPL Version 2 or to extend the choice of license to its licensees as provided above. However, if you add GPL Version 2 code and therefore, elected the GPL Version 2 license, then the option applies only if the new code is made subject to such option by the copyright holder.

/****
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2007 INRIA, France Telecom
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/Type.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/commons/GeneratorAdapter.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/signature/SignatureVisitor.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/tree/LdcInsnNode.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/ClassVisitor.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/MethodAdapter.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/util/ASMifierFieldVisitor.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/commons/TableSwitchGenerator.java
No license file was found, but licenses were detected in source scan.

/**
 * ASM XML Adapter
 * Copyright (c) 2004, Eugene Kuleshov
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 * contributors may be used to endorse or promote products derived from
 * this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
OPEN SOURCE USED IN DNAC PLATFORM ANTELOPE

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/xml/SAXFieldAdapter.java
  * /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/xml/SAXAdapter.java
  * /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/xml/SAXClassAdapter.java
  * /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/xml/SAXCodeAdapter.java
  * /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/xml/ASMContentHandler.java
  * /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/xml/SAXAnnotationAdapter.java
  * /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-jar/org/objectweb/asm/xml/Processor.java

No license file was found, but licenses were detected in source scan.

2005 INRIA, France Telecom
* All rights reserved.

* Redistribution and use in source and binary forms, with or without
  modification, are permitted provided that the following conditions
  are met:
  * 1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
  * 2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
  * 3. Neither the name of the copyright holders nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/tree/analysis/package.html
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/package.html
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/commons/package.html
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/tree/package.html
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/signature/package.html
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/util/package.html
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/attrs/package.html
* No license file was found, but licenses were detected in source scan.

* ASM XML Adapter
* Copyright (c) 2004, Eugene Kuleshov
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
Open Source Used In DNAC Platform Antelope 8694

* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/xml/package.html

No license file was found, but licenses were detected in source scan.

/**
 * ASM: a very small and fast Java bytecode manipulation framework
 * Copyright (c) 2000-2007 INRIA, France Telecom
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holders nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */

Found in path(s):
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/util/Traceable.java
* /opt/cola/permits/1136447751_1613767073.0/0/asm-all-repackaged-2-2-0-b21-sources-3-
  jar/org/objectweb/asm/util/ASMifiable.java
1.2290 follow-redirects 1.5.10 1.5.10

1.2290.1 Available under license:

Copyright 2014-present Olivier Lalonde <olalonde@gmail.com>, James Talmage <james@talmage.io>, Ruben Verborgh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2291 ansi-styles 4.3.0 4.3.0

1.2291.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2292 es-abstract 1.17.7

1.2292.1 Available under license:

The MIT License (MIT)

Copyright (C) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2293 vector 1.19.0

1.2293.1 Available under license:

<#--

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

<#--
/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Drill : Exec : Vectors
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2294 esri-geometry-api 2.2.2

1.2294.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
Copyright 1995-2013 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

e-mail: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/CombineOperator.java
No license file was found, but licenses were detected in source scan.

*/

Copyright 1995-2017 Esri

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

e-mail: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
Open Source Used In DNAC Platform Antelope 8703

- jar/com/esri/core/geometry/OperatorCentroid2DLocal.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/OperatorImportFromJsonLocal.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/ogc/OGCMultiPoint.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/JSONUtils.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/AttributeStreamOfFloat.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/AttributeStreamOfDbl.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/JsonParserReader.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/ogc/OGCMultiCurve.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/OperatorImportFromGeoJsonLocal.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/OperatorFactoryLocal.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/ogc/OGCMultiSurface.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/OperatorImportFromJson.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/AttributeStreamOfInt8.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/JsonReader.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/ogc/OGCLinearRing.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/AttributeStreamOfInt64.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/OperatorCentroid2D.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/Operator.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/ogc/OGCLineString.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/AttributeStreamOfInt16.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/ogc/OGCSurface.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/JsonGeometryException.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
  jar/com/esri/core/geometry/OperatorImportFromJsonCursor.java
- /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/SimpleJsonReaderCursor.java
No license file was found, but licenses were detected in source scan.

/*
Copyright 1995-2015 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

email: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/ShapeImportFlags.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/OperatorExportToWktLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/SweepComparator.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/OperatorIntersectionLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/TopologicalOperations.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/OperatorWithinLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/OperatorConvexHull.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/IntervalTreeImpl.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/OperatorOffset.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/Point3D.java
Open Source Used In DNAC Platform Antelope 8706

* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorOffsetCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorCrosses.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/GenericGeometrySerializer.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/MultiPathImpl.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorDisjointLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorSimplifyCursorOGC.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorInternalRelationUtils.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorDifferenceCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorImportFromWktLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/JsonWriter.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorClipLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorConvexHullLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorConvexHullCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorSimplifyLocalHelper.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorProximity2DLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SimpleByteBufferCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/StringUtils.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorProximity2DLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorEquals.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ConstructOffset.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorExportToWkb.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/VertexDescriptionHash.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorGeodeticAreaLocal.java
Open Source Used In DNAC Platform Antelope  8707

* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SegmentIterator.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorSimplifyCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/MultiVertexGeometry.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorCrossesLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/GeoJsonCrsTables.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorBoundaryLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ObjectCacheTable.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Clusterer.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/WkbByteOrder.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorDistance.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/JsonStringWriter.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/NonSimpleResult.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Proximity2DResultComparator.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SegmentBuffer.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorGeneralize.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorProjectLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Simplificator.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorExportToJsonLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Envelope1D.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/RingOrientationFixer.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/GeoDist.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/EditShape.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorGeodesicBuffer.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ShapeType.java
Open Source Used In DNAC Platform Antelope 8708

* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Cracker.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Interop.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ShapeModifiers.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorUnionLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorSymmetricDifferenceLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorSimplifyLocalOGC.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorImportFromWkb.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorExportToJsonCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/JsonCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorClipCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorDifferenceLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/MapOGCStructure.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorImportFromESRIShapeLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Envelope2DIntersectorImpl.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorOverlapsLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/PathFlags.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorImportFromGeoJsonLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/WkbExportFlags.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/AttributeStreamBase.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorGeodeticDensifyByLength.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Cutter.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorProject.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/BucketSort.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorFactory.java
Open Source Used In DNAC Platform Antelope 8709

* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Point.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorExportToESRIShapeCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/MultiPointImpl.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/DirtyFlags.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ListeningGeometryCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Envelope3D.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/GeometryCursorAppend.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorWithin.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/IndexMultiList.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SimpleJsonCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/PairwiseIntersectorImpl.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorCutLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorExportToWkbLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorDistanceLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorOverlaps.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/GeometrySerializer.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorGeodeticLengthLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorDensifyByLength.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/NumberUtils.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorGeodeticArea.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorSimplify.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/RelationalOperations.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/RasterizedGeometry2D.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ByteBufferCursor.java
Open Source Used In DNAC Platform Antelope 8711
Open Source Used In DNAC Platform Antelope 8712

* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorTouches.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SegmentIntersector.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/WktExportFlags.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorOffsetLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorClip.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/MultiPoint.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/GeometryCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorIntersectionCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SegmentIteratorImpl.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorIntersection.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/GeoJsonImportFlags.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/CrackAndCluster.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/EnvSrlzr.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SimpleMapGeometryCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorImportFromESRIShapeCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Line.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorShapePreservingDensify.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ProjectionTransformation.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/PointInPolygonHelper.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/HashTable.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorBoundaryLocalCursor.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ConvexHull.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorImportFromWkt.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/ECordinate.java
Open Source Used in DNAC Platform Antelope 8713

* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorCut.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SegmentFlags.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorProximity2D.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorGeodeticDensifyLocal.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/SpatialReferenceSerializer.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Proximity2DResult.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/OperatorDifference.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/Proximity2DResult.java

No license file was found, but licenses were detected in source scan.

/*
Copyright 1995-2013 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

e-mail: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/envelope3d-java-gz/Envelope3D.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/point3d-java-gz/Point3D.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-jar/com/esri/core/geometry/index3d-java-gz/IndexMultiDCLList.java

---

Open Source Used in DNAC Platform Antelope 8713
jar/com/esri/core/geometry/RasterizedGeometry2DImpl.java
No license file was found, but licenses were detected in source scan.

/*/  
Copyright 1995-2017 Esri  

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at  

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

For additional information, contact:  
Environmental Systems Research Institute, Inc.  
Attn: Contracts Dept  
380 New York Street  
Redlands, California, USA 92373  

email: contracts@esri.com  
*/  
/*  
COPYRIGHT 1995-2017 ESRI  

TRADE SECRETS: ESRI PROPRIETARY AND CONFIDENTIAL  
Unpublished material - all rights reserved under the  
Copyright Laws of the United States.

For additional information, contact:  
Environmental Systems Research Institute, Inc.  
Attn: Contracts Dept  
380 New York Street  
Redlands, California, USA 92373  

email: contracts@esri.com  
*/  

Found in path(s):  
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-  
jar/com/esri/core/geometry/JsonReaderCursor.java  
No license file was found, but licenses were detected in source scan.  

/*/
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

e-mail: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/OperatorExportToGeoJson.java
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/OperatorExportToGeoJsonLocal.java

No license file was found, but licenses were detected in source scan.

/*
Copyright 1995-2018 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373
jar/com/esri/core/geometry/ogc/OGCMultiLineString.java

No license file was found, but licenses were detected in source scan.

/*
Copyright 2013-2015 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

email: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092327381_1600729703.66/0/esri-geometry-api-2-2-2-sources-
jar/com/esri/core/geometry/SimpleRasterizer.java

1.2295 commons-pool 2.4.3

1.2295.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

APACHE COMMONS POOL DERIVATIVE WORKS:
The LinkedBlockingDeque implementation is based on an implementation written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/licenses/publicdomain

Apache Commons Pool

Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.2296 @babel/plugin-proposal-numeric-separator 7.12.13

1.2296.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2297 detect-indent 4.0.0 4.0.0

1.2297.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy

Open Source Used In DNAC Platform Antelope 8722
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2298 window-size 0.1.4

1.2298.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2299 scala-reflect 2.13.5
1.2299.1 Available under license:

Scala
Copyright (c) 2002-2020 EPFL
Copyright (c) 2011-2020 Lightbend, Inc.

Scala includes software developed at
LAMP/EPFL (https://lamp.epfl.ch/) and
Lightbend, Inc. (https://www.lightbend.com/).

Licensed under the Apache License, Version 2.0 (the "License").
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This software includes projects with other licenses -- see `doc/LICENSE.md`.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2300 nice-try 1.0.5

1.2300.1 Available under license :
The MIT License (MIT)

Copyright (c) 2018 Tobias Reich

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2301 etag 1.7.0

1.2301.1 Available under license:

(The MIT License)

Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2302 colors.js 1.4.0

1.2302.1 Available under license:

MIT License

Original Library
- Copyright (c) Marak Squires

Additional Functionality
- Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell.
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2303 caller-path 0.1.0

1.2303.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2304 is-number-object 1.0.4

1.2304.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Jordan Harband
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2305 activation-api 1.2.0

1.2305.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.
1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it
absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and
make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under
Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software
with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction’s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys’ fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.
The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

**TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost
of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to
distribute corresponding source code. (This alternative is allowed
only for noncommercial distribution and only if you received the
program in object code or executable form with such an offer, in
accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source code
means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to control
compilation and installation of the executable. However, as a special
exception, the source code distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies the
executable.

If distribution of executable or object code is made by offering access
to copy from a designated place, then offering equivalent access to copy
the source code from the same place counts as distribution of the source
code, even though third parties are not compelled to copy the source
along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt otherwise
to copy, modify, sublicense or distribute the Program is void, and will
automatically terminate your rights under this License. However, parties
who have received copies, or rights, from you under this License will
not have their licenses terminated so long as such parties remain in
full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and all
its terms and conditions for copying, distributing or modifying the
Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further restrictions
on the recipients' exercise of the rights granted herein. You are not
responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot distribute
so as to satisfy simultaneously your obligations under this License and
any other pertinent obligations, then as a consequence you may not
distribute the Program at all. For example, if a patent license would
not permit royalty-free redistribution of the Program by all those who
receive copies directly or indirectly through you, then the only way you
could satisfy both it and this License would be to refrain entirely from
distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is implemented
by public license practices. Many people have made generous
contributions to the wide range of software distributed through that
system in reliance on consistent application of that system; it is up to
the author/donor to decide if he or she is willing to distribute
software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be
a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License may
add an explicit geographical distribution limitation excluding those
countries, so that distribution is permitted only in or among countries
not thus excluded. In such case, this License incorporates the
limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new
versions of the General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

#

Certain source files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

You should also note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result. Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications to this package which may subsequently be redistributed and/or involve the use of third party software.

CLASSPATH EXCEPTION
Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole
As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

1.2306 react-lifecycles-compat 3.0.4 3.0.4
1.2306.1 Available under license :
MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2307 base64-js 0.0.2
1.2307.1 Available under license :
MIT
1.2308 p-finally 1.0.0

1.2308.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2309 @babel/helper-wrap-function 7.13.0

1.2309.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
1.2310 okhttp 4.9.1

1.2310.1 Available under license:

Note that publicsuffixes.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
https://mozilla.org/MPL/2.0/
Mozilla Public License
Version 2.0
1. Definitions
1.1. “Contributor”
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. “Contributor Version”
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.

1.3. “Contribution”
means Covered Software of a particular Contributor.

1.4. “Covered Software”
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. “Incompatible With Secondary Licenses”
means
that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. “Executable Form”
means any form of the work other than Source Code Form.

1.7. “Larger Work”
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. “License”
means this document.
1.9. “Licensable”
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. “Modifications”
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. “Patent Claims” of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. “Secondary License”
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. “Source Code Form”
means the form of the work preferred for making modifications.

1.14. “You” (or “Your”)
means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions
2.1. Grants
Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date
The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope
The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses
No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation
Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use
This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions
Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities
3.1. Distribution of Source Form
All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients’ rights in the Source Code Form.

3.2. Distribution of Executable Form
If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients’ rights in the Source Code Form under this License.
3.3. Distribution of a Larger Work
You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices
You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms
You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation
If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination
5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.
6. Disclaimer of Warranty
Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability
Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation
Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party’s ability to bring cross-claims or counter-claims.

9. Miscellaneous
This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License
10.1. New Versions
Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions
You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions
If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses
If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this
version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice
This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - “Incompatible With Secondary Licenses” Notice
This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.
/*
 * Copyright (C) 2014 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

1.2311 jetty-xml 9.3.27.v20190418

1.2312 scala 2.11.12
1.2312.1 Available under license :

Scala includes the Sizzle library:

Copyright (c) 2010 The Dojo Foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
Scala is licensed under the [BSD 3-Clause License](http://opensource.org/licenses/BSD-3-Clause).

## Scala License

Copyright (c) 2002-2017 EPFL

Copyright (c) 2011-2017 Lightbend, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the EPFL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Other Licenses

This software includes projects with the following licenses, which are also included in the `licenses/` directory:

### [Apache License](http://www.apache.org/licenses/LICENSE-2.0.html)

This license is used by the following third-party libraries:

* jansi

### [BSD License](http://www.opensource.org/licenses/bsd-license.php)
This license is used by the following third-party libraries:

* jline

### [BSD 3-Clause License](http://opensource.org/licenses/BSD-3-Clause)
This license is used by the following third-party libraries:

* asm

### [MIT License](http://www.opensource.org/licenses/MIT)
This license is used by the following third-party libraries:

* jquery
* jquery-ui
* jquery-layout
* sizzle
* tools tooltip

### Public Domain
The following libraries are freely available in the public domain:

* forkjoin

Scala includes the JLine library:

Copyright (c) 2002-2006, Marc Prud’hommeaux <mwp1@cornell.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2011 Paul Bakaus, http://jqueryui.com/

This software consists of voluntary contributions made by many individuals (AUTHORS.txt, http://jqueryui.com/about) For exact contribution history, see the revision history and logs, available at http://jquery-ui.googlecode.com/svn/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2010 Fabrizio Balliano, Kevin Dalman

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Scala includes the ASM library.

Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

not-a-legal-formal-parameter-tuple.scala:2: error: not a legal formal parameter.
Note: Tuples cannot be directly destructured in method or function parameters.

Either create a single parameter accepting the Tuple2,

or consider a pattern matching anonymous function: `\{ case (a, b) => ... \}
val x: ((Int, Int) => Int) = (((a, b)) => a)
^

not-a-legal-formal-parameter-tuple.scala:3: error: not a legal formal parameter.
Note: Tuples cannot be directly destructured in method or function parameters.

Either create a single parameter accepting the Tuple2,

or consider a pattern matching anonymous function: `\{ case (param1, param2) => ... \}
val y: ((Int, Int, Int) => Int) = (((a, !!)) => a)
not-a-legal-formal-parameter-tuple.scala:4: error: not a legal formal parameter.
Note: Tuples cannot be directly destructured in method or function parameters.
Either create a single parameter accepting the Tuple3,
    or consider a pattern matching anonymous function: `{ case (param1, ..., param3) => ... }`
val z: ((Int, Int, Int) => Int) = {((a, NotAPatternVariableName, c)) => a}
^ three errors found

Scala includes the Tools Tooltip library:

Copyright (c) 2009 Tero Piirainen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
(c) 2012-2014 GitHub

When using the GitHub logos, be sure to follow the GitHub logo guidelines (https://github.com/logos)

Font License: SIL OFL 1.1 (http://scripts.sil.org/OFL)
Applies to all font files

Code License: MIT (http://choosealicense.com/licenses/mit/)
Applies to all other files
Copyright (c) 2006, Ivan Sagalaev
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of highlight.js nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS `AS IS` AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Scala includes the JLine library, which includes the Jansi library.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Scala includes the jQuery library:

Copyright (c) 2010 John Resig

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

(The MIT License)

Copyright (c) 2013 Greg Allen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2313 bson 4.2.3

1.2313.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2314 thrift 0.9.2

1.2314.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of
fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source
code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form,
including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as
indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix
below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices. 
that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]" replaced with your own identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2315 kerb-util 1.0.0
1.2315.1 Available under license:

Kerby-kerb Util
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"; you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2316 setimmediate 1.0.5

1.2316.1 Available under license :

   Copyright (c) 2012 Barnesandnoble.com, llc, Donavon West, and Domenic Denicola

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2317 acorn 5.7.4
1.2317.1 Available under license :
Copyright (C) 2012-2018 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2318 reactor-core 3.0.7.RELEASE
1.2318.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2011-2016 Pivotal Software Inc, All Rights Reserved.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
* /

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
  jar/reactor/core/scheduler/RejectedDisposable.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
  jar/reactor/core/publisher/ImmutableSignal.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
  jar/reactor/core/publisher/MutableNextSignal.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
  jar/reactor/core/publisher/BufferOverflowStrategy.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2011-2017 Pivotal Software Inc, All Rights Reserved.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 * */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
  jar/reactor/core/publisher/BlockingMonoSubscriber.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
  jar/reactor/core/publisher/MonoFilterWhen.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
  jar/reactor/core/publisher/MonoFilterWhen.java

Open Source Used In DNAC Platform Antelope 8778
jar/reactor/core/scheduler/DisposableContainer.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxFilterWhen.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/EmptyDisposableContainer.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/ScheduledRunnable.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2011-2016 Pivotal Software Inc, All Rights Reserved.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/util/concurrent/OpenHashSet.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/util/concurrent/QueueSupplier.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/ParallelScheduler.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/ExecutorScheduler.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/ElasticScheduler.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/util/function/Tuples.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/ExecutorScheduler.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/ElasticScheduler.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/BlockingFirstSubscriber.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/Scheduler.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/scheduler/Scheduler.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/util/package-info.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2011-2017 Pivotal Software Inc, All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *       http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxBufferBoundary.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoPublishOn.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/ParallelFluxHide.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/BlockingSink.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxSkipUntil.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxRepeat.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/ConnectableFluxOnAssembly.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoMaterialize.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxPublishMulticast.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoRunnable.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/SpscArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/util/concurrent/SpscLinkedArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/util/concurrent/SpscArrayQueue.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/util/Loggers.java
*/
Open Source Used In DNAC Platform Antelope

jar/reactor/core/publisher/SignalLogger.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxWindowPredicate.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/DelegateProcessor.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxRetry.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/ParallelMergeReduce.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoAny.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxTake.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoThenMap.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoSubscribeOnCallable.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxError.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxPublishOn.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoTakeLastOne.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoFlatMap.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxRetryPredicate.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoElementAt.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxDelaySubscription.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxPeekFuseable.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxCancelOn.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/ParallelMap.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxSink.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/MonoFilter.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxCallable.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxInterval.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxTakeUntil.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-jar/reactor/core/publisher/FluxTake.java

8784
jar/reactor/core/publisher/MonoEmpty.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoThenIgnore.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoDelayElement.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoDelay.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoDoFinally.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoAll.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoPeek.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxGroupBy.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxGroupJoin.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxDoFinally.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxBuffer.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoStreamCollector.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/EmitterProcessor.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/Scannable.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/ParallelSource.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/BlockingSingleSubscriber.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxOnBackpressureDrop.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxOnBackpressureBuffer.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxAutoConnectFuseable.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxStream.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/SerializedSubscriber.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoDelayElement.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoDelay.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoSubscribeOn.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxRange.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxGroupJoin.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoSubscribeOn.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxGroupBy.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxDoFinally.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoAll.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoPeek.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxOnBackpressureDrop.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoDoFinally.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxBuffer.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxDoFinally.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/MonoDoFinally.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
jar/reactor/core/publisher/FluxOnBackpressureDrop.java
* /opt/ws_local/PERMITS_SQL/1069088736_1594483108.64/0/reactor-core-3-0-7-release-sources-
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such
license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2320 fn-name 3.0.0 3.0.0

1.2320.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2321 esri-geometry-api 2.2.0

1.2321.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
Copyright 1995-2017 Esri
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

e-mail: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/JSONUtils.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/AttributeStreamOfInt8.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/ogc/OGCCurve.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/JsonGeometryException.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/ogc/OGCMultiSurface.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorImportFromJsonLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/ogc/OGCMultiPoint.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/JsonReader.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/JsonParserReader.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/AttributeStreamOfInt32.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorImportFromJsonCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/ogc/OGCSurface.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorFactoryLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorImportFromJson.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/AttributeStreamOfDbl.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/ogc/OGCMultiCurve.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/SimpleJsonReaderCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/AttributeStreamOfInt64.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/AttributeStreamOfInt16.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorCentroid2D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorImportFromGeoJson.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/Operator.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/ogc/OGCLinearRing.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/AttributeStreamOfFloat.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorCentroid2DLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/ogc/OGCGeometryCollection.java

No license file was found, but licenses were detected in source scan.

/

Copyright 1995-2013 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

email: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/point3d-java-gz/Point3D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/envelope3d-java-gz/Envelope3D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/RasterizedGeometry2Dimpl.java
No license file was found, but licenses were detected in source scan.

/**
Copyright 1995-2017 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

email: contracts@esri.com
*/
*/
COPYRIGHT 1995-2017 ESRI

TRADE SECRETS: ESRI PROPRIETARY AND CONFIDENTIAL
Unpublished material - all rights reserved under the
Copyright Laws of the United States.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

email: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/JsonReaderCursor.java
No license file was found, but licenses were detected in source scan.

/*
Copyright 2013-2015 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

email: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/SimpleRasterizer.java
No license file was found, but licenses were detected in source scan.

/*
Copyright 1995-2013 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

e-mail: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/CombineOperator.java
No license file was found, but licenses were detected in source scan.

/*
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

e-mail: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/OperatorExportToGeoJson.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/OperatorExportToGeoJsonLocal.java
No license file was found, but licenses were detected in source scan.

/*
Copyright 1995-2015 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

e-mail: contracts@esri.com
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorShapePreservingDensifyLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorExportToJson.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorSymmetricDifferenceLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorUnionLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorDistance.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/Treap.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/SpatialReferenceImpl.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/MathUtils.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/Envelope3D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorDistance.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorUnionLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorDistance.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/Treap.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorUnionLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
Open Source Used In DNAC Platform Antelope 8801
jar/com/esri/core/geometry/RingOrientationFixer.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/PsSrlzr.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/RasterizedGeometry2D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorExportToESRIShapeCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorEquals.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/TopoGraph.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorUnionCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/SegmentIteratorImpl.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/WktExportFlags.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/JsonStringWriter.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/AttributeStreamBase.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorIntersectionCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorConvexHullCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/ECoordinate.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorGeodeticLength.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/ShapeExportFlags.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/LnSrlzr.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/IntervalTreeImpl.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/SegmentFlags.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorInternalRelationUtils.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/ListeningGeometryCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/PathFlags.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/MultiVertexGeometry.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorShapePreservingDensify.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
Open Source Used In DNAC Platform Antelope 8804

jar/com/esri/core/geometry/NonSimpleResult.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/GeometryCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorGeneralizeCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/Proximity2DResult.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/ConvexHull.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OGCStructure.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorContainsLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/GeodeticCurveType.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorSimplifyCursorOGC.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/Cracker.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/Point3D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorTouches.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorBuffer.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorSimplifyLocalOGC.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorImportFromESRIShape.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/PointInPolygonHelper.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorIntersection.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorProject.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/IndexMultiList.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/Clipper.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/ShapeType.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/SweepMonkierComparator.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/PlaneSweepCrackerHelper.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
  jar/com/esri/core/geometry/OperatorOffset.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/WkbImportFlags.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorGeodeticDensifyLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/Transformation2D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/PairwiseIntersectorImpl.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/WktImportFlags.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/VertexDescriptionDesignerImpl.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorGeodesicBufferLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/Wkt.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/MgrsConversionMode.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/Wkid.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorDensifyByLength.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/JsonCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorBoundaryLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorSimplifyOGC.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorExportToJsonLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorCutCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/UserCancelException.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/IndexMultiDCLlist.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/EnvSrkr.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorExportToGeoJsonCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/Line.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorGeodeticLengthLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/OperatorCrosses.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/GeoJsonExportFlags.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-
jar/com/esri/core/geometry/Simplificator.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/DirtyFlags.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/StringUtils.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/OperatorImportFromWktLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/OperatorSimplifyCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/OperatorSymmetricDifference.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/SimpleByteBufferCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/IndexHashTable.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/GeometrySerializer.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/OperatorBoundaryLocalCursor.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/OperatorIntersectionLocal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/QuadTreeImpl.java

No license file was found, but licenses were detected in source scan.

/*

Copyright 1995-2018 Esri

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

For additional information, contact:
Environmental Systems Research Institute, Inc.
Attn: Contracts Dept
380 New York Street
Redlands, California, USA 92373

email: contracts@esri.com
*/
Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/ogc/OGCGeometry.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/Polyline.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/Geometry.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/MapGeometry.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/SizeOf.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/Point2D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/Envelope2D.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/ogc/OGCPolygon.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/SpatialReference.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/GeometryEngine.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/Polygon.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/Envelope.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/ogc/OGCLineString.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/ogc/OGCMultiPolygon.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/ogc/OGCMultiLineString.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/OGCStructureInternal.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/ogc/OGCConcreteGeometryCollection.java
* /opt/ws_local/PERMITS_SQL/1066571557_1593680380.14/0/esri-geometry-api-2-2-0-sources-jar/com/esri/core/geometry/ogc/OGCPolygon.java

1.2322 mkdirp-promise 1.1.0 1.1.0
1.2322.1 Available under license :
Copyright (c) 2015, Ahmad Nassri <ahmad@ahmadnassri.com>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2323 drill-memory-base 1.19.0

1.2323.1 Available under license:

Drill : Exec : Memory : Base
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2324 arr-union 3.1.0

1.2324.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2325 batik-extension 1.13

1.2325.1 Available under license:

Apache Batik

Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for Standardization for the definition of character entities used in the software's documentation.

This product includes images from the Tango Desktop Project (http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme (http://www.jesusda.com/projects/pasodoble).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2326 kerb-simplekdc 1.0.0

1.2326.1 Available under license:

Kerb Simple Kdc
Copyright 2014-2017 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2327 lcid 1.0.0

1.2327.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2328 @babel/helper-member-expression-to-functions 7.13.12

1.2328.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2329 libsoup 2.52.2-1ubuntu0.3

1.2329.1 Available under license:

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not
Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is
analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is
"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.
(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.
You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the

Open Source Used In DNAC Platform Antelope 8832
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above,

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
 distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
d version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Library General Public License for more details.

You should have received a copy of the GNU Library General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.2330 lazystream 1.0.0

1.2330.1 Available under license:

Copyright (c) 2013 J. Pommerening, contributors.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
1.2331 homedir-polyfill 1.0.3
1.2331.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Brian Woodward

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2332 annotations 13.0
1.2332.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2006 Sascha Weinreuter
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
* See the License for the specific language governing permissions and 
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Subst.java 
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Pattern.java 
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Language.java 
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/PrintFormat.java 
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/RegExp.java 
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/Identifier.java 

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2000-2012 JetBrains s.r.o.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/TestOnly.java 
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/MagicConstant.java 
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/intellij/lang/annotations/JdkConstants.java 
* /opt/cola/permits/1138123071_1614251056.08/0/annotations-13-0-sources-3-jar/org/jetbrains/annotations/NotNull.java 

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2000-2009 JetBrains s.r.o.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-
  jar/org/jetbrains/annotations/NonNls.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-
  jar/org/jetbrains/annotations/Nullable.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-
  jar/org/jetbrains/annotations/Nls.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-
  jar/org/jetbrains/annotations/PropertyKey.java
No license file was found, but licenses were detected in source scan.

/***************************************************************************/

* Copyright 2000-2013 JetBrains s.r.o.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-
  jar/org/intellij/lang/annotations/Flow.java
* /opt/cola/permits/1138123144_1614251056.08/0/annotations-13-0-sources-3-
  jar/org/jetbrains/annotations/Contract.java
1.2333 caseless 0.12.0 0.12.0

1.2333.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was
submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2334 lodash 4.17.10 4.17.10

1.2334.1 Available under license:

Copyright JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/
Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

### 1.2335 chalk 3.0.0
#### 1.2335.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.2336 extsprintf 1.4.0 1.4.0
#### 1.2336.1 Available under license:
Copyright (c) 2012, Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, 
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE 
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER 
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, 
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN 
THE SOFTWARE

1.2337 sax 1.2.4 1.2.4

1.2337.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any 
purpose with or without fee is hereby granted, provided that the above 
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES 
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF 
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR 
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES 
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN 
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR 
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

=====

`String.fromCharCode` by Mathias Bynens used according to terms of MIT 
License, as follows:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining 
a copy of this software and associated documentation files (the 
"Software"), to deal in the Software without restriction, including 
without limitation the rights to use, copy, modify, merge, publish, 
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
1.2338 xmlbuilder 9.0.4
1.2338.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013 Ozgur Ozcitak

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2339 isstream 0.1.2
1.2339.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Rod Vagg

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2340 abbrev 1.1.1 1.1.1

1.2340.1 Available under license:

This software is dual-licensed under the ISC and MIT licenses.
You may use this software under EITHER of the following licenses.

--------

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

--------

Copyright Isaac Z. Schlueter and Contributors
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2341 babel-plugin-dynamic-import-node

2.3.3

1.2341.1 Available under license:
MIT License

Copyright (c) 2016 Airbnb

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2342 calcite-core 1.21.0-drill-r2

1.2342.1 Available under license:

# Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.
# The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with
# Database of contributors to Apache Calcite.  
# Pages such as developer.md use this data.  
# List must be sorted by first name, last name.  
# 
- name: Alan Gates  
  emeritus: 2018/05/04  
  apacheId: gates  
  githubId: alanfgates  
  org: Hortonworks  
  role: PMC  
- name: Aman Sinha  
  apacheId: amansinha  
  githubId: amansinha100  
  org: MapR  
  role: PMC  
- name: Andrei Sereda  
  apacheId: sereda  
  githubId: asereda-gs  
  org:  
  role: Committer  
- name: Ashutosh Chauhan  
  apacheId: hashutosh  
  githubId: ashutoshe  
  org: Hortonworks  
  role: PMC  
- name: Chris Baynes  
  apacheId: cbaynes  
  githubId: chris-baynes  
  org: Contiamo  
  role: Committer  
- name: Christian Beikov  
  apacheId: cbeikov  
  githubId: beikov  
  org: Blazebit  
  role: Committer  
- name: Chunwei Lei  
  apacheId: chunwei  
  githubId: chunweilei
org: Alibaba
role: Committer
- name: Danny Chan
  apacheId: danny0405
githubId: danny0405
org: Alibaba
role: Committer
- name: Edmon Begoli
  apacheId: ebegoli
githubId: ebegoli
org: Oak Ridge National Laboratory
role: Committer
- name: Francis Chuang
  apacheId: francischuang
githubId: F21
org: Boostport
role: PMC Chair
- name: Gian Merlino
  apacheId: gian
githubId: gianm
org: Imply
role: Committer
- name: Haisheng Yuan
  apacheId: hyuan
githubId: hsyuan
org: Alibaba
role: Committer
- name: Hongze Zhang
  apacheId: hongze
githubId: zhztheplayer
org: Tencent
role: Committer
- name: James R. Taylor
  apacheId: jamestaylor
githubId: JamesRTaylor
org: Salesforce
role: PMC
- name: Jacques Nadeau
  apacheId: jacques
githubId: jacques-n
org: Dremio
role: PMC
- name: Jess Balint
  apacheId: jbalint
githubId: jbalint
org: Stardog
role: Committer
- name: Jess Camacho Rodriguez
- name: Jinfeng Ni
  apacheId: jni
  githubId: jinfengni
  org: MapR
  role: PMC
- name: John Pullokkaran
  apacheId: jpullokk
  githubId: jpullokkaran
  org: Hortonworks
  role: PMC
- name: Josh Elser
  apacheId: elserj
  githubId: joshelser
  org: Hortonworks
  role: PMC
- name: Julian Hyde
  apacheId: jhyde
  githubId: julianhyde
  org: Looker
  role: PMC
  homepage: http://people.apache.org/~jhyde
- name: Kevin Liew
  apacheId: kliew
  githubId: kliewkliew
  org:
  role: Committer
- name: Kevin Risden
  apacheId: krisden
  githubId: risdenk
  org:
  role: PMC
- name: Laurent Goujon
  apacheId: laurent
  githubId: laurentgo
  org: Dremio
  role: PMC
- name: Maryann Xue
  apacheId: maryannxue
  githubId: maryannxue
  org: Intel
  role: PMC
- name: Michael Mior
  apacheId: mmior
  githubId: michaelmior
org: Rochester Institute of Technology
role: PMC
homepage: https://michael.mior.ca/
- name: Milinda Pathirage
apacheId: milinda
githubId: milinda
org: Indiana University
role: Committer
homepage: http://milinda.pathirage.org/
- name: MinJi Kim
apacheId: minji
githubId: minji-kim
org: Oracle
role: Committer

- name: Nick Dimiduk
apacheId: ndimiduk
githubId: ndimiduk
org: Oracle
role: PMC
homepage: http://www.n10k.com

- name: Nishant Bangarwa
apacheId: nishant
githubId: nishantmonu51
org: Hortonworks
role: Committer

- name: Ruben Quesada Lopez
apacheId: rubenql
githubId: rubenada
org: TIBCO
role: Committer

- name: Sergey Nuyanzin
apacheId: snuyanzin
githubId: snuyanzin
org: EPAM
role: Committer

- name: Shuyi Chen
apacheId: shuyichen
githubId: suetz224
org: Uber
role: Committer

- name: Slim Bouguerra
apacheId: bslim
githubId: b-slim
org: Hortonworks
role: Committer

- name: Stamatis Zampetakis
apacheId: zabetak
githubId: zabetak
org: TIBCO
role: PMC
- name: Steven Noels
  apacheId: stevenn
  githubId: stevenn
org: NGData
role: PMC
- name: Ted Dunning
  apacheId: tdunning
  githubId: tdunning
org: MapR
role: PMC
  avatar: https://mapr.com/blog/author/ted-dunning/assets/tdunning-panama.jpg
- name: Vladimir Sitnikov
  apacheId: vladimirsitnikov
  githubId: vlsi
org: NetCracker
role: PMC
- name: Volodymyr Vysotskyi
  apacheId: volodymyr
  githubId: vvysotskyi
org: 
role: PMC
- name: Zhen Wang
  apacheId: zhenw
  githubId: zinking
org: 
role: Committer
- name: Zhiqiang He
  apacheId: zhiqianghe
  githubId: Zhiqiang-He
org: Huawei
role: Committer
- name: Zhiwei Peng
  apacheId: zhiwei
  githubId: pengzhiwei2018
org: Ant Financial Group
role: Committer
- name: Zoltan Haindrich
  apacheId: kgyrtkirk
  githubId: kgyrtkirk
org: Hortonworks
role: Committer

# End contributors.yml

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE CALCITE SUBCOMPONENTS:

The Apache Calcite project contains subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

The MIT License

The Apache Calcite project bundles the following files under the MIT License:

- site
  Parts of the web site generated by Jekyll (http://jekyllrb.com/)
  Copyright (c) 2008-2015 Tom Preston-Werner
- site/_sass/_font-awesome.scss
  Font-awesome css files v4.1.0 (http://fortawesome.github.io/Font-Awesome/)
  Copyright (c) 2013 Dave Gandy
- site/_sass/_normalize.scss
  normalize.css v3.0.2 | git.io/normalize
  Copyright (c) Nicolas Gallagher and Jonathan Neal
- site/_sass/_gridism.scss
  Gridism: A simple, responsive, and handy CSS grid by @cobyism
  https://github.com/cobyism/gridism
  Copyright (c) 2013 Coby Chapple
- site/js/html5shiv.min.js
  HTML5 Shiv 3.7.2 | @afarkas @dalton @jon_neal @rem
- site/js/respond.min.js
  Respond.js v1.4.2: min/max-width media query polyfill
  Copyright 2013 Scott Jehl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----------------------------------------------------------------------

The Open Font License
-----------------------------------------------------------------------

The Apache Calcite project bundles the following fonts under the SIL Open Font License (OFL) - http://scripts.sil.org/OFL/

- site/fonts/fontawesome-webfont.*
  Font-awesome font files v4.0.3 (http://fortawesome.github.io/Font-Awesome/)
Apache Calcite
Copyright 2012-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product is based on source code originally developed by DynamoBI Corporation, LucidEra Inc., SQLstream Inc. and others under the auspices of the Eigenbase Foundation and released as the LucidDB project.

The web site includes files generated by Jekyll.

1.2343 escodegen 1.14.3 1.14.3

1.2343.1 Available under license :
Copyright (C) 2012 Yusuke Suzuki (twitter: @Constellation) and other contributors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2344 spdx-license-ids 3.0.6

1.2344.1 Available under license :

```
root = true

[*]
charset = utf-8
end_of_line = lf
indent_style = tab
tab_width = 2
insert_final_newline = true
trim_trailing_whitespace = true
```
A list of [SPDX license](https://spdx.org/licenses/) identifiers

## Installation

[Download JSON directly](https://raw.githubusercontent.com/shinnn/spdx-license-ids/master/index.json), or [use](https://docs.npmjs.com/cli/install) [npm](https://docs.npmjs.com/about-npm):

```bash
npm install spdx-license-ids
```

## [Node.js](https://nodejs.org/) API

```javascript
const ids = require('spdx-license-ids');
//=> ['0BSD', 'AAL', 'ADSL', 'AFL-1.1', 'AFL-1.2', 'AFL-2.0', 'AFL-2.1', 'AFL-3.0', 'AGPL-1.0-only', ...
ids.includes('BSD-3-Clause'); //=> true
ids.includes('CC-BY-1.0'); //=> true
```
***
```javascript
const deprecatedIds = require('spdx-license-ids/deprecated');
//=> ['AGPL-1.0', 'AGPL-3.0', 'GFDL-1.1', 'GFDL-1.2', 'GFDL-1.3', 'GPL-1.0', 'GPL-2.0', ...]
```

deprecatedIds.includes('BSD-3-Clause'); //=> false
deprecatedIds.includes('CC-BY-1.0'); //=> false

deprecatedIds.includes('GPL-3.0'); //=> true
***

### License

[Creative Commons Zero v1.0 Universal](https://creativecommons.org/publicdomain/zero/1.0/deed)
[*_.workflow_*]
indent_style = space

1.2345 is-string 1.0.5

1.2345.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.2346 touch 3.1.0

1.2346.1 Available under license:

MIT License

Copyright (c) 2018 livelybone(2631541504@qq.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2347 express 4.17.1

1.2347.1 Available under license:

(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2013-2014 Roman Shylman <shtylman+expressjs@gmail.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
 included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2348 jsreport-child-templates 1.4.0

1.2348.1 Available under license:
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
ingengineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license
document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this
License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer
system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
Version.

e) Provide Installation Information, but only if you would otherwise
be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is
necessary to install and execute a modified version of the
Combined Work produced by recombining or relinking the
Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)


You may place library facilities that are a work based on the
Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.2349 glob-parent 5.1.1
1.2349.1 Available under license:
The ISC License

Copyright (c) 2015, 2019 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2350 node-promise 0.5.13 0.5.13
1.2350.1 Available under license :

No license file was found, but licenses were detected in source scan.

MIT License.

The node-promise project provides a complete promise implementation. Promises provide a clean separation of concerns between asynchronous behavior and the interface so asynchronous functions can be called without callbacks, and callback interaction can be done on the generic promise interface. The node-promise package provides just a promise implementation, however, https://github.com/kriszyp/promised-io is recommended for more complete promise-based IO functionality. The promised-io includes the promise implementation from node-promise, as well as wrappers around Node's filesystem and other system I/O APIs for consistent promise-based interaction.

The node-promise module features a promise implementation with:

* Chainable promises
* Promises throw errors if an error handler is not provided
* CommonJS promise proposal [1] compliant
* Immutable once fulfilled to reduce possible side-effects
* Promises can be used securely (as separate resolver/promise pairs in ocap situations)
* Backwards compatibility where possible (addCallback, addErrback, emitSuccess, and emitError should still behave as expected)

Utility functions, including:

* when() - Normalization of sync (normal values) and async (promises)
* all() - Create a promise that accumulate multiple concurrent promises (failed promises resolve to Error objects)
* allOrNone() - Ditto, but the first promise to fail causes the composition to fail as well
* first() - Find the first promise to be fulfilled in a group of promises
* seq() - Sequentially execute a set of promise returning functions
* delay() - Returns a promise that is fulfilled after a given amount of time
* execute() - Executes a function that takes a callback and returns a promise (thank you Benjamin Thomas for providing this)

Much of this is adapted from Tyler Close's ref_send and Kris Kowal's work on promises.
Some quick examples from test-promise.js (again, it is recommended that you use http://github.com/kriszyp/promised-io for file and other I/O interaction):

```
util = require('util');
var fs = require('./fs-promise);

// open a file and read it
fs.open("fs-promise.js", process.O_RDONLY).then(function(fd){
 return fs.read(fd, 4096);
}).then(function(args){
 util.puts(args[0]); // print the contents of the file
});

// does the same thing
fs.readFile("fs-promise.js").addCallback(util.puts);
```

A default Promise constructor can be used to create a self-resolving deferred/promise:

```
var Promise = require("node-promise").Promise;
var promise = new Promise();
asyncOperation(function(){
 promise.resolve("successful result");
});
promise -> given to the consumer
```

A consumer can use the promise:

```
promise.then(function(result){
 ... when the action is complete this is executed ...
 },
 function(error){
 ... executed when the promise fails
 });
```

Alternately, a provider can create a deferred and resolve it when it completes an action. The deferred object a promise object that provides a separation of consumer and producer to protect promises from being fulfilled by untrusted code.

```
var defer = require("node-promise").defer;
var deferred = defer();
asyncOperation(function(){
 deferred.resolve("successful result");
});
deferred.promise -> given to the consumer
```

Another way that a consumer can use promises:

```
var when = require("node-promise").when;
when(promise,function(result){
```
... when the action is complete this is executed ...
},
function(error){
    ... executed when the promise fails
});

More examples:

function printFirstAndLast(itemsDeferred){
    findFirst(itemsDeferred).then(util.puts);
    findLast(itemsDeferred).then(util.puts);
}
function findFirst(itemsDeferred){
    return itemsDeferred.then(function(items){
        return items[0];
    });
}
function findLast(itemsDeferred){
    return itemsDeferred.then(function(items){
        return items[items.length];
    });
}

And now you can do:

printFirstAndLast(someAsyncFunction());

The workhorse function of this library is the "when" function, which provides a means for normalizing interaction with values and functions that may be a normal synchronous value, or may be a promise (asynchronously fulfilled). The when() function takes a value that may be a promise or a normal value for the first function, and when the value is ready executes the function provided as the second argument (immediately in the case of a non-promise normal value). The value returned from when() is the result of the execution of the provided function, and returns a promise if provided a promise or synchronously returns a normal value if provided a non-promise value. This makes it easy to "chain" computations together. This allows us to write code that is agnostic to sync/async interfaces:

var when = require("node-promise").when;
function printFirstAndLast(items){
    // print the first and last item
    when(findFirst(items), util.puts);
    when(findLast(items), util.puts);
}
function findFirst(items){
    // return the first item
    return when(items, function(items){
        return items[0];
    });
}
function findLast(items){
    // return the last item
    return when(items, function(items){
        return items[items.length - 1];
    });
}

Now we can do:

> printFirstAndLast([1,2,3,4,5]);
1
5

And we can also provide asynchronous promise:

    var promise = new process.Promise();
    > printFirstAndLast(promise);

(nothing printed yet)

> promise.emitSuccess([2,4,6,8,10]);
2
10

The "all" function is intended to provide a means for waiting for the completion of an array of promises. The "all" function should be passed an array of promises, and it returns an promise that is fulfilled once all the promises in the array are fulfilled. The returned promise's resolved value will be an array with the resolved values of all of the promises in the passed in array.

The "first" function is intended to provide a means for waiting for the completion of the first promise in an array of promises to be fulfilled. The "first" function should be passed an array of promises, and it returns an promise that is fulfilled once the first promise in the array is fulfilled. The returned promise's resolved value will be the resolved value of the first fulfilled promise.

Found in path(s):
* /opt/cola/permits/1025855972_1636567465.07/0/node-promise-0-5-13-tgz/package/README.md
  No license file was found, but licenses were detected in source scan.

    // // MIT License

Found in path(s):
* /opt/cola/permits/1025855972_1636567465.07/0/node-promise-0-5-13-tgz/package/promise.js

1.2351 safer-buffer 2.1.2
1.2351.1 Available under license:
MIT License

Copyright (c) 2018 Nikita Skovoroda <chalkerx@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2352 jsreport-version-control 1.6.0
1.2352.1 Available under license:
MIT License

Copyright (c) 2017 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2353 jquery 3.4.0 3.4.0

1.2353.1 Available under license:
Copyright JS Foundation and other contributors, https://js.foundation/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2354 get-stream 5.1.0

1.2354.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Apache Commons Codec
Copyright 2002-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===============================================================================
The content of package org.apache.commons.codec.language has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

1.2356 babel-plugin-transform-es2015-destructuring 6.23.0
1.2356.1 Available under license :
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2357 hadoop-auth 3.2.2
1.2357.1 Available under license :
This product includes software developed by The Apache Software Foundation (http://www.apache.org/).
The binary distribution of this product bundles binaries of org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:

* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of Gson 2.2.4, which has the following notices:

The Netty Project
=================

Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

------------------------------------------------------------------------

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:
This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:
This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression
and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/
This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasternxml.com/AaltoHome
This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
* license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
* https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
* license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
* https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
* https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
* http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4,
which has the following notices:
* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)
===============================================================================
The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.
Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6,
which has the following notices:
* This product includes software from the Spring Framework,under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17,
which has the following notices:
* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
TypeUtil.java
The binary distribution of this product bundles binaries of “Java Concurrency in Practice” book annotations 1.0, which has the following notices:

* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)

Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,

which has the following notices:

* ==============================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
  ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL:

* org.eclipse.jetty.orbit.org.eclipse.jdt.core
The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.
org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------
MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies./

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
  (Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by
  "GCC Runtime Library Exception"

== Contributors ==
  * Tatu Saloranta
    * Providing benchmark suite
  * Alec Wysoker
    * Performance and memory usage improvement

The binary distribution of this product bundles binaries of
  Xerces2 Java Parser 2.9.1,
  which has the following notices:
  *
  == NOTICE file corresponding to section 4(d) of the Apache License, ==
  == Version 2.0, in this case for the Apache Xerces Java distribution. ==

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
  - voluntary contributions made by Paul Eng on behalf of the
    Apache Software Foundation that were originally developed at iClick, Inc.,
    software copyright (c) 1999.

The binary distribution of this product bundles binaries of
  Logback Classic Module 1.1.2,
  Logback Core Module 1.1.2,
  which has the following notices:
  * Logback: the reliable, generic, fast and flexible logging framework.
    Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
  Apache HBase - Annotations 1.2.6,
  Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0
--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting
it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was
relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:
--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically 'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of Phoenix Core 4.7.0, which has the following notices:

Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5 (https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 2008-2015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 2007-2008 Robert Niederreiter, Jens Klein
# Copyright 2003-2005 Chad J. Schroeder
# Copyright 2002-2001 Jrgen Hermann

The binary distribution of this product bundles binaries of Plexus Cipher: encryption/decryption Component 1.4, which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of software.amazon.ion:ion-java 1.0.1, which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.
The binary distribution of this product bundles binaries of joda-time:joda-time:2.9.9
which has the following notices:

* = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=============================================================================
This product includes software developed by Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its...
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 * Copyright (c) 2005, European Commission project OneLab under contract 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 * - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*++

* Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
*
* This software program is licensed subject to the BSD License,
* available at http://www.opensource.org/licenses/bsd-license.html
*
* Abstract: The main routine
*
* --*/

For src/main/native/src/org/apache/hadoop/io/compress/lz4/\{lz4.h,lz4.c,lz4hc.h,lz4hc.c\},

/*

LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at:
- LZ4 source repository: http://code.google.com/p/lz4/
- LZ4 public forum: https://groups.google.com/forum/#!forum/lz4c
*/

For hadoop-common-project/hadoop-common/src/main/native/gtest

---------------------------------------------------------------------
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1

---------------------------------------------------------------
(GO license)
This is a work derived from Russ Cox's RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni
(https://github.com/fusesource/leveldbjni), which is available under the
following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of FuseSource Corp. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=============

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:
Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

--------------------------------------------------------------------------------

Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

--------------------------------------------------------------------------------

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
ewm-helpers v0.5.13
ewm-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js

The MIT License (MIT)

Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js

The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
oj! Algorithms - version 43.0

-------------------------------------------------------------


This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY. WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are
externally maintained libraries used by this software which have their
own licenses; we recommend you read them, as their terms may differ from
the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js
--------------------------------------------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
D3 is available under a 3-clause BSD license. For details, see:
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
HSQLDB Database 2.3.4

(HSQL License)

"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the
Hypersonic SQL Group.”

The binary distribution of this product bundles these dependencies under the
following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
1. Definitions.

1.1. Contributor means each individual or entity
that creates or contributes to the creation of
1.2. Contributor Version means the combination of the
Original Software, prior Modifications used by a Contributor (if any), and the
Modifications made by that particular Contributor.

1.3. Covered
Software means (a) the Original Software, or (b) Modifications, or (c) the
combination of files containing Original Software with files containing
Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other
than Source Code.

1.5. Initial Developer means the individual or entity
that first makes Original Software available under this
License.

1.6. Larger Work means a work which combines Covered Software or
portions thereof with code not governed by the terms of this
License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time
of the initial grant or subsequently acquired, any and all of the rights
conveyed herein.

1.9. Modifications means the Source Code and Executable
form of any of the following:
A. Any file that results from an addition to,
deletion from or modification of the contents of a file containing Original
Software or previous Modifications;
B. Any new file that contains any part of the Original Software
or previous Modification; or
C. Any new file that is contributed or otherwise made available
under the terms of this License.

1.10. Original Software means the Source Code and Executable form of
computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or
hereafter acquired, including without limitation, method, process, and apparatus
claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which
modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor
(or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent
Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any
3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code
form that alters or restricts the applicable version of this License or the
recipients rights hereunder. You may choose to offer, and to charge a fee for,
warranty, support, indemnity or liability obligations to one or more recipients
of Covered Software. However, you may do so only on Your own behalf, and not on
behalf of the Initial Developer or any Contributor. You must make it absolutely
clear that any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial Developer
and every Contributor for any liability incurred by the Initial Developer or
such Contributor as a result of warranty, support, indemnity or liability terms
You offer.

3.5. Distribution of Executable Versions. You may distribute the
Executable form of the Covered Software under the terms of this License or under
the terms of a license of Your choice, which may contain terms different from
this License, provided that You are in compliance with the terms of this License
and that the license for the Executable form does not attempt to limit or alter
the recipients rights in the Source Code form from the rights set forth in this
License. If You distribute the Covered Software in Executable form under a
different license, You must make it absolutely clear that any terms which differ
from this License are offered by You alone, not by the Initial Developer or
Contributor. You hereby agree to indemnify the Initial Developer and every
Contributor for any liability incurred by the Initial Developer or such
Contributor as a result of any such terms You offer.

3.6. Larger Works. You
may create a Larger Work by combining Covered Software with other code not
governed by the terms of this License and distribute the Larger Work as a single
product. In such a case, You must make sure the requirements of this License are
fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may
publish revised and/or new versions of this License from time to time. Each
version will be given a distinguishing version number. Except as provided in
Section 4.3, no one other than the license steward has the right to modify this
License.

4.2. Effect of New Versions. You may always continue to use,
distribute or otherwise make the Covered Software available under the terms of
the version of the License under which You originally received the Covered
Software. If the Initial Developer includes a notice in the Original Software
prohibiting it from being distributed or otherwise made available under any
subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either
unilaterally or pursuant to a written agreement with Participant.

6.3. In
the event of termination under Sections 6.1 or 6.2 above, all end user licenses
that have been validly granted by You or any distributor hereunder prior to
termination (excluding licenses granted to You by any distributor) shall survive
termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER
NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE,
SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF
COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY
PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY
CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF
GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER
COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE
POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO
LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO
THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT
ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO
THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in
48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as
that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer
software documentation as such terms are used in 48C.F.R.12.212 (Sept.
227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software
with only those rights set forth herein. This U.S. Government Rights clause is
in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision
that addresses Government rights in computer software under this
License.

9. MISCELLANEOUS.
This License represents the complete agreement
concerning subject matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent necessary to
make it enforceable. This License shall be governed by the law of the
jurisdiction specified in a notice contained within the Original Software
(except to the extent applicable law, if any, provides otherwise), excluding
such jurisdictions conflict-of-law provisions. Any litigation relating to this
License shall be subject to the jurisdiction of the courts located in the
jurisdiction and venue specified in a notice contained within the Original
Software, with the losing party responsible for costs, including, without
limitation, court costs and reasonable attorneys fees and expenses. The
application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:

jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
grizzly-http 2.2.21
grizzly-http-server 2.2.21
grizzly-http-servlet 2.2.21
grizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims
infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the
terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.
4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.
6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is
defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
This file is copyrighted by Red Hat Inc.
Atomicops support for AIX/POWER, located in
src/google/protobuf/stubs/atomicops_internals_power.h.
This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

By obtaining, using and/or copying this work, you (the licensee) agree that you
have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation,
with or without modification, for any purpose and without fee or royalty is
hereby granted, provided that you include the following on ALL copies of the
software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the
redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.
“Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.
2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient’s responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a
medium customarily used for software exchange. When the Program is made available in source code form:

a) it must be made available under this Agreement; and 
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR
IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable
separate entity. Each new version of the Agreement will be given a
distinguishing version number. The Program (including Contributions) may always
be distributed subject to the version of the Agreement under which it was
received. In addition, after a new version of the Agreement is published,
Contributor may elect to distribute the Program (including its Contributions)
under the new version. Except as expressly stated in Sections 2(a) and 2(b)
above, Recipient receives no rights or licenses to the intellectual property of
any Contributor under this Agreement, whether expressly, by implication,
estoppel or otherwise. All rights in the Program not expressly granted under
this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in
any resulting litigation.

The binary distribution of this product bundles these dependencies under the
following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveledbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the
  names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

(2-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions
"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g.
ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by
other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., “French translation of the Work by Original Author,” or “Screenplay based on original Work by Original Author”). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective

Open Source Used In DNAC Platform Antelope 8936
Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

jamon-runtime 2.4.1

(MPL 2.0)

Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributors Contribution.

1.3. Contribution means Covered Software of a particular Contributor.

1.4. Covered Software means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses means
that the initial Contributor has attached the notice described in Exhibit B to
the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or
earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate
file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the
time of the initial grant or subsequently, any and all of the rights conveyed by
this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or
modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and
apparatus claims, in any patent Licensable by such Contributor that would be
infringed, but for the grant of the License, by the making, using, selling,
offering for sale, having made, import, or transfer of either its Contributions
or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General
Public License, Version 2.1, the GNU Affero General Public License, Version 3.0,
or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For
legal entities, You includes any entity that controls, is controlled by,
or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).
2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.
3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the
rights granted under this License from a particular Contributor are reinstated
(a) provisionally, unless and until such Contributor explicitly and finally
terminates Your grants, and (b) on an ongoing basis, if such Contributor fails
to notify You of the non-compliance by some reasonable means prior to 60 days
after You have come back into compliance. Moreover, Your grants from a
particular Contributor are reinstated on an ongoing basis if such Contributor
notifies You of the non-compliance by some reasonable means, this is the first
time You have received notice of non-compliance with this License from such
Contributor, and You become compliant prior to 30 days after Your receipt of the
notice.

5.2. If You initiate litigation against any entity by asserting a patent
infringement claim (excluding declaratory judgment actions, counter-claims, and
cross-claims) alleging that a Contributor Version directly or indirectly
infringes any patent, then the rights granted to You by any and all Contributors
for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user
license agreements (excluding distributors and resellers) which have been
validly granted by You or Your distributors under this License prior to
termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without
warranty of any kind, either expressed, implied, or statutory, including,
without limitation, warranties that the Covered Software is free of defects,
merchantable, fit for a particular purpose or non-infringing. The entire risk as
to the quality and performance of the Covered Software is with You. Should any
Covered Software prove defective in any respect, You (not any Contributor)
assume the cost of any necessary servicing, repair, or correction. This
disclaimer of warranty constitutes an essential part of this License. No use of
any Covered Software is authorized under this License except under this
disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including
negligence), contract, or otherwise, shall any Contributor, or anyone who
distributes Covered Software as permitted above, be liable to You for any
direct, indirect, special, incidental, or consequential damages of any character
including, without limitation, damages for lost profits, loss of goodwill, work
stoppage, computer failure or malfunction, or any and all other commercial
damages or losses, even if such party shall have been informed of the
possibility of such damages. This limitation of liability shall not apply to
liability for death or personal injury resulting from such party's negligence
to the extent applicable law prohibits such limitation. Some jurisdictions do
not allow the exclusion or limitation of incidental or consequential damages, so
this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If you choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.
Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:

JDOM 1.1

FLICTSIBILITYWARNING

//**--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact request_AT_jdom_DOT_org.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management request_AT_jdom_DOT_org.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:

“This product includes software developed by the
JDOM Project (http://www.jdom.org/).”
Alternatively, the acknowledgment may be graphical using the logos

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the
following license:
Hbase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is
available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE
LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN
ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS
INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES
REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR
DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE
COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.
c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a
performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive,
perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that
recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution (“Attribution Parties”) in Licensor’s copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URL, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be
otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this
License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages.
whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.


Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2358 babel-code-frame 6.26.0 6.26.0

1.2358.1 Available under license:

MIT License

Copyright (c) 2014-2017 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2359 my-scope-package-a 0.0.0
1.2359.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright \{yyyy\} [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
The JsonCpp library’s source code, including accompanying documentation,
tests and demonstration applications, are licensed under the following
conditions...

The author (Baptiste Lepilleur) explicitly disclaims copyright in all
jurisdictions which recognize such a disclaimer. In such jurisdictions,
this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of
2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur, and is
released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this
software may choose to accept it either as 1) Public Domain, 2) under the
conditions of the MIT License (see below), or 3) under the terms of dual
Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is
described in clear, concise terms at:

    http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

========================================================================
Copyright (c) 2007-2010 Baptiste Lepilleur

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
========================================================================
The MIT license is compatible with both the GPL and commercial software, affording one all of the rights of Public Domain with the minor nuisance of being required to keep the above copyright notice and license text in the source code. Note also that by accepting the Public Domain "license" you can re-license your copy using whatever license you like.

All files in this directory are provided by the following license if not stated otherwise in the individual file:

This file is provided under a dual BSD/GPLv2 license. When using or redistributing this file, you may do so under either license.

GPL LICENSE SUMMARY

Copyright (c) 2005-2012 Intel Corporation. All rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St - Fifth Floor, Boston, MA 02110-1301 USA. The full GNU General Public License is included in this distribution in the file called LICENSE.GPL.

Contact Information:

BSD LICENSE

Copyright (c) 2005-2012 Intel Corporation. All rights reserved. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the
distribution.
* Neither the name of Intel Corporation nor the names of its
  contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This license applies to all parts of V8 that are not externally
maintained libraries. The externally maintained libraries used by V8
are:

- PCRE test suite, located in
test/mjsunit/third_party/regexp-pcre/regexp-pcre.js. This is based on the
test suite from PCRE-7.3, which is copyrighted by the University
of Cambridge and Google, Inc. The copyright notice and license
are embedded in regexp-pcre.js.

- Layout tests, located in test/mjsunit/third_party/object-keys. These are
  based on layout tests from webkit.org which are copyrighted by
  Apple Computer, Inc. and released under a 3-clause BSD license.

- Strongtalk assembler, the basis of the files assembler-arm-inl.h,
  assembler-arm.cc, assembler-arm.h, assembler-ia32-inl.h,
  assembler-ia32.cc, assembler-ia32.h, assembler-x64-inl.h,
  assembler-x64.cc, assembler-x64.h, assembler-mips-inl.h,
  assembler-mips.cc, assembler-mips.h, assembler.cc and assembler.h.
  This code is copyrighted by Sun Microsystems Inc. and released
  under a 3-clause BSD license.

- Valgrind client API header, located at src/third_party/valgrind/valgrind.h
  This is released under the BSD license.

- The Wasm C/C++ API headers, located at third_party/wasm-api/wasm.{h,hh}
  This is released under the Apache license. The API's upstream prototype
  implementation also formed the basis of V8's implementation in
  src/wasm/c-api.cc.
These libraries have their own licenses; we recommend you read them, as their terms may differ from the terms below.

Further license information can be found in LICENSE files located in sub-directories.

Copyright 2014, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1994-2006 Sun Microsystems Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Sun Microsystems or the names of contributors may
be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Tests included here are based on build output generated by the six-speed benchmark suite.

Copyright (c) 2015 Kevin Decker

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CC0 1.0 Universal

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later
claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;

ii. moral rights retained by the original author(s) and/or performer(s);

iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;

iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;

v. rights protecting the extraction, dissemination, use and reuse of data in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of
action), in the Work (i) in all territories worldwide, (ii) for the maximum
duration provided by applicable law or treaty (including future time
extensions), (iii) in any current or future medium and for any number of
copies, and (iv) for any purpose whatsoever, including without limitation
commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes
the Waiver for the benefit of each member of the public at large and to the
detriment of Affirmer's heirs and successors, fully intending that such Waiver
shall not be subject to revocation, rescission, cancellation, termination, or
any other legal or equitable action to disrupt the quiet enjoyment of the Work
by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be
judged legally invalid or ineffective under applicable law, then the Waiver
shall be preserved to the maximum extent permitted taking into account
Affirmer's express Statement of Purpose. In addition, to the extent the Waiver
is so judged Affirmer hereby grants to each affected person a royalty-free,
non transferable, non sublicensable, non exclusive, irrevocable and
unconditional license to exercise Affirmer's Copyright and Related Rights in
the Work (i) in all territories worldwide, (ii) for the maximum duration
provided by applicable law or treaty (including future time extensions), (iii)
in any current or future medium and for any number of copies, and (iv) for any
purpose whatsoever, including without limitation commercial, advertising or
promotional purposes (the "License"). The License shall be deemed effective as
of the date CC0 was applied by Affirmer to the Work. Should any part of the
License for any reason be judged legally invalid or ineffective under
applicable law, such partial invalidity or ineffectiveness shall not
invalidate the remainder of the License, and in such case Affirmer hereby
affirms that he or she will not (i) exercise any of his or her remaining
Copyright and Related Rights in the Work or (ii) assert any associated claims
and causes of action with respect to the Work, in either case contrary to
Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned,
surrendered, licensed or otherwise affected by this document.

b. Affirmer offers the Work as-is and makes no representations or warranties
of any kind concerning the Work, express, implied, statutory or otherwise,
including without limitation warranties of title, merchantability, fitness
for a particular purpose, non infringement, or the absence of latent or
other defects, accuracy, or the present or absence of errors, whether or not
discernable, all to the greatest extent permissible under applicable law.

c. Affirmer disclaims responsibility for clearing rights of other persons
that may apply to the Work or any use thereof, including without limitation
any person's Copyright and Related Rights in the Work. Further, Affirmer
disclaims responsibility for obtaining any necessary consents, permissions
or other rights required for any use of the Work.

d. Affirmer understands and acknowledges that Creative Commons is not a
party to this document and has no duty or obligation with respect to this
CC0 or use of the Work.

For more information, please see
<http://creativecommons.org/publicdomain/zero/1.0/>

----------------------------------------------------------------

Notice that the following BSD-style license applies to this one
file (valgrind.h) only. The rest of Valgrind is licensed under the
terms of the GNU General Public License, version 2, unless
otherwise indicated. See the COPYING file in the source
distribution for details.

----------------------------------------------------------------

This file is part of Valgrind, a dynamic binary instrumentation
framework.

Copyright (C) 2000-2010 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product
documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must
not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ”AS IS” AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Notice that the above BSD-style license applies to this one file
(valgrind.h) only. The entire rest of Valgrind is licensed under
the terms of the GNU General Public License, version 2. See the
COPYING file in the source distribution for details.

// Copyright 2016 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PCRE LICENCE
------------

PCRE is a library of functions to support regular expressions whose syntax
and semantics are as close as possible to those of the Perl 5 language.

Release 7 of PCRE is distributed under the terms of the "BSD" licence, as
specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself.

The basic library functions are written in C and are freestanding. Also included in the distribution is a set of C++ wrapper functions.

THE BASIC LIBRARY FUNCTIONS
-----------------------------
Written by: Philip Hazel
Email local part: ph10
Email domain: cam.ac.uk

University of Cambridge Computing Service,

Copyright (c) 1997-2007 University of Cambridge
All rights reserved.

THE C++ WRAPPER FUNCTIONS
-----------------------------
Contributed by: Google Inc.

Copyright (c) 2007, Google Inc.
All rights reserved.

THE "BSD" LICENCE
-------------------
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

End
Copyright (c) 2006 Apple Computer, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder(s) nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2010 Jonathan Hartley
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holders, nor those of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.
<table>
<thead>
<tr>
<th>Release</th>
<th>Derived</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td>1.2</td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.2 and above</td>
<td>2.1.1</td>
<td>2001-now</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright,

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").
2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1
---------------------------------------

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to
Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2
--------------------------------------------------

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (C) 1993-2004 by Sun Microsystems, Inc. All rights reserved.

Developed at SunSoft, a Sun Microsystems, Inc. business.
Permission to use, copy, modify, and distribute this software is freely granted, provided that this notice is preserved.

Notice that the following BSD-style license applies to this one file (valgrind.h) only. The rest of Valgrind is licensed under the terms of the GNU General Public License, version 2, unless otherwise indicated. See the COPYING file in the source.
This file is part of Valgrind, a dynamic binary instrumentation framework.

Copyright (C) 2000-2007 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

W3C 3-clause BSD License

http://www.w3.org/Consortium/Legal/2008/03-bsd-license.html

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of works must retain the original copyright notice,
this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the original copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the W3C nor the names of its contributors may be used to endorse or promote products derived from this work without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2006-2011, the V8 project authors. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.2360 uuid 3.2.1

1.2360.1 Available under license:

The MIT License (MIT)

Copyright (c) 2010-2016 Robert Kieffer and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2361 babel-helper-compilation-targets

7.13.13

1.2361.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2362 xpp 1.1.4c

1.2362.1 Notifications :
This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

1.2362.2 Available under license :
/*
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2001-2003 The Apache Software Foundation. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 *
 * 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:
 * "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Axis" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.

* ====================================================================

* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
*/

1.2363 path-dirname 1.0.2

1.2363.1 Available under license:

The MIT License (MIT)

Copyright (c) Elan Shanker and Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
1.2364 pi-spec-converter-web 0.0.0 0.0.0
1.2364.1 Available under license :

    core-js@2.5.1
    MIT
    Copyright (c) 2014-2017 Denis Pushkarev

    Permission is hereby granted, free of charge, to any person obtaining a copy
    of this software and associated documentation files (the "Software"), to deal
    in the Software without restriction, including without limitation the rights
    to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
    copies of the Software, and to permit persons to whom the Software is
    furnished to do so, subject to the following conditions:

    The above copyright notice and this permission notice shall be included in
    all copies or substantial portions of the Software.

    THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
    IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
    FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
    AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
    LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
    OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
    THE SOFTWARE.

    webpack@3.6.0
    MIT
    Copyright JS Foundation and other contributors

    Permission is hereby granted, free of charge, to any person obtaining
    a copy of this software and associated documentation files (the
    'Software'), to deal in the Software without restriction, including
    without limitation the rights to use, copy, modify, merge, publish,
    distribute, sublicense, and/or sell copies of the Software, and to
    permit persons to whom the Software is furnished to do so, subject to
    the following conditions:

    The above copyright notice and this permission notice shall be
    included in all copies or substantial portions of the Software.

    THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
    EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
    MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
    IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
    CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
    TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

zone.js@0.8.18
MIT
The MIT License

Copyright (c) 2016 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

@angular/core@4.4.5
MIT
MIT

@angular/router@4.4.5
MIT
MIT

@angular/http@4.4.5
MIT
MIT

@angular/forms@4.4.5
MIT
MIT

@angular/platform-browser@4.4.5
MIT
MIT

@angular/common@4.4.5
MIT
MIT
babel-polyfill@6.26.0
MIT
MIT

core-js@2.5.1
MIT
Copyright (c) 2014-2017 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

webpack@3.6.0
MIT
Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
zone.js@0.8.18
MIT
The MIT License
Copyright (c) 2016 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

@angular/core@4.4.5
MIT
MIT

@angular/http@4.4.5
MIT
MIT

file-saver@1.3.3
MIT
The MIT License

Copyright  2016 [Eli Grey][1].

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[1]: http://eligrey.com

@angular/forms@4.4.5
MIT
MIT

@angular/platform-browser@4.4.5
MIT
MIT

@angular/common@4.4.5
MIT
MIT
(MIT License)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2365 kryo-shaded 3.0.3
1.2366 @babel/traverse 7.11.5

1.2366.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2367 jsprim 1.4.1 1.4.1

1.2367.1 Available under license:
Copyright (c) 2012, Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE

1.2368 es6-symbol 3.1.3
1.2368.1 Available under license:
ISC License

Copyright (c) 2013-2019, Mariusz Nowak, @medikoo, medikoo.com

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

1.2369 use 3.1.1
1.2369.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
turnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.2370 sqlite 3.25.3

1.2370.1 Available under license:
The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:

May you do good and not evil.
May you find forgiveness for yourself and forgive others.
May you share freely, never taking more than you give.

1.2371 which 1.3.1

1.2371.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2372 semver 5.5.0 5.5.0

1.2372.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2373 entities 1.1.2

1.2373.1 Available under license:

Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2374 accepts 1.3.3

1.2374.1 Available under license:

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2375 ipaddr.js 1.9.1
1.2375.1 Available under license :
Copyright (C) 2011-2017 whitequark <whitequark@whitequark.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2376 har-validator 5.1.5 5.1.5
1.2376.1 Available under license :
MIT License

Copyright (c) 2018 Ahmad Nassri <ahmad@ahmadnassri.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2377 ant 1.10.9

1.2377.1 Available under license :

/*
   * Apache License
   * Version 2.0, January 2004
   * http://www.apache.org/licenses/
   *
   * TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
   *
   * 1. Definitions.
   *
   * "License" shall mean the terms and conditions for use, reproduction,
   * and distribution as defined by Sections 1 through 9 of this document.
   *
   * "Licensor" shall mean the copyright owner or entity authorized by
   * the copyright owner that is granting the License.
   *
   * "Legal Entity" shall mean the union of the acting entity and all
   * other entities that control, are controlled by, or are under common
   * control with that entity. For the purposes of this definition,
   * "control" means (i) the power, direct or indirect, to cause the
   * direction or management of such entity, whether by contract or
   * otherwise, or (ii) ownership of fifty percent (50%) or more of the
   * outstanding shares, or (iii) beneficial ownership of such entity.
   *
   * "You" (or "Your") shall mean an individual or Legal Entity
   * exercising permissions granted by this License.
   *
   * "Source" form shall mean the preferred form for making modifications,
   * including but not limited to software source code, documentation
   * source, and configuration files.
   *
   * "Object" form shall mean any form resulting from mechanical
   * transformation or translation of a Source form, including but
   * not limited to compiled object code, generated documentation,
   * and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

W3C SOFTWARE NOTICE AND LICENSE
https://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from...
which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

____________________________________

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered under http://.saxproject.org/?selected=pd

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05
Apache Ant
Copyright 1999-2020 The Apache Software Foundation

Open Source Used In DNAC Platform Antelope 8995
This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

The <sync> task is based on code Copyright (c) 2002, Landmark
Graphics Corp that has been kindly donated to the Apache Software
Foundation.

1.2378 pluralize 1.1.6 1.1.6
1.2378.1 Available under license :
The MIT License (MIT)

Copyright (c) 2013 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2379 hk2-utils 2.2.0-b21
1.2379.1 Available under license :
Found license 'General Public License 2.0' in * Copyright (c) 2007-2011 Oracle and/or its affiliates. All rights
reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2
only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License'
Found license 'Common Development and Distribution License (CDDL) 1.1' in 'Copyright (c) 2010-2011 Oracle
and/or its affiliates. All rights reserved. The contents of this file are subject to the terms of either the GNU General
Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL")
(collectively, the "License"). You may not use this file except in compliance with the License. You can Oracle
designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2
1.2380 amdefine 1.0.1

1.2380.1 Available under license:

Dojo is available under *either* the terms of the modified BSD license *or* the Academic Free License version 2.1. As a recipient of Dojo, you may choose which license to receive this code under (except as noted in per-module LICENSE files). Some modules may not be the copyright of the Dojo Foundation. These modules contain explicit declarations of copyright in both the LICENSE files in the directories in which they reside and in the code itself. No external contributions are allowed under licenses which are fundamentally incompatible with the AFL or BSD licenses that Dojo is distributed under.

The text of the AFL and BSD licenses is reproduced below.

-------------------------------------------------------------------------------
The "New" BSD License:
******************************************************************************

Copyright (c) 2005-2009, The Dojo Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Dojo Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----------------------------------------------------------------------------

The Academic Free License, v. 2.1:
******************************************************************************

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

a) to reproduce the Original Work in copies;

b) to prepare derivative works ("Derivative Works") based upon the Original Work;

c) to distribute copies of the Original Work and Derivative Works to the public;

d) to perform the Original Work publicly; and

e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent
claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately proceeding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or
seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

amdefine is released under two licenses: new BSD, and MIT. You may pick the license that best suits your development needs. The text of both licenses are provided below.

The "New" BSD License:
----------------------

Copyright (c) 2011-2016, The Dojo Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Dojo Foundation nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License
----------

Copyright (c) 2011-2016, The Dojo Foundation

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2381 read-all-stream 1.0.2 1.0.2
1.2381.1 Available under license :
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2382 hadoop-mapreduce-client-core 3.2.2

1.2382.1 Available under license:

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of
org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of
AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java),
which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of
Gson 2.2.4,
which has the following notices:

The Netty Project
=================
Please visit the Netty web site for more information:

* http://netty.io/
Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

-------------------------------------------------------------------------------

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
  * HOMEPAGE:
    * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
  * HOMEPAGE:
    * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
  * HOMEPAGE:
    * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
  * HOMEPAGE:
    * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:
This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:
This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:
This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of
Commons Codec 1.4,
which has the following notices:
* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.

Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of
Commons Lang 2.6,
which has the following notices:
* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of
Apache Log4j 1.2.17,
which has the following notices:
* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of
"Java Concurrency in Practice" book annotations 1.0,
which has the following notices:
* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,
which has the following notices:
* =============
Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.

===============================================================================

Open Source Used In DNAC Platform Antelope 9009
The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.
* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions. http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies./

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
(Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by "GCC Runtime Library Exception" http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html

== Contributors ==
* Tatu Saloranta
* Providing benchmark suite
* Alec Wysoker
* Performance and memory usage improvement

The binary distribution of this product bundles binaries of Xerces2 Java Parser 2.9.1, which has the following notices:
* =======================================================================
  == NOTICE file corresponding to section 4(d) of the Apache License, ==
  == Version 2.0, in this case for the Apache Xerces Java distribution. ==
  =======================================================================

Apache Xerces Java
Later we found that vectorfree.com image is not properly licensed. The original is owned by vectorportal.com. The original was relicensed so we could use it as Creative Commons Attribution 3.0. The license is bundled with the download available here: http://www.vectorportal.com/subcategory/205/KILLER-WHALE-FREE-VECTOR.eps/ifile/9136/detailtest.asp

This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

This product includes portions of the Guava project v14, specifically 'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of Phoenix Core 4.7.0, which has the following notices:

    Apache Phoenix
    Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This also includes:

    The phoenix-spark module has been adapted from the phoenix-spark library distributed under the terms of the Apache 2 license. Original source copyright:
    Copyright 2014 Simply Measured, Inc.
    Copyright 2015 Interset Software Inc.

    The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5 (https://pypi.python.org/pypi/python-daemon/). Original source copyright:
    # Copyright 2008-2015 Ben Finney <ben+python@benfinney.id.au>
The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
  which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1,
which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9
which has the following notices:
* =-------------------------------------------------------------------------------------------------------------------------------------
  = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
  =-------------------------------------------------------------------------------------------------------------------------------------

  This product includes software developed by
  Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 *
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL
 * nor the names of its contributors may be used to endorse or
 * promote products derived from this software without specific prior
* written permission.
* 
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
* ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*++
 *
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 *
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 *
 * Abstract: The main routine
 *
--*/

For src/main/native/src/org/apache/hadoop/io/compress/lz4/{lz4.h,lz4.c,lz4hc.h,lz4hc.c},

/*
 LZ4 - Fast LZ compression algorithm
 Header File
 Copyright (C) 2011-2014, Yann Collet.
 BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

 Redistribution and use in source and binary forms, with or without
 modification, are permitted provided that the following conditions are
 met:

 * Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above
   copyright notice, this list of conditions and the following disclaimer
   in the documentation and/or other materials provided with the
   distribution.

 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

 You can contact the author at:
 - LZ4 source repository : http://code.google.com/p/lz4/
 - LZ4 public forum : https://groups.google.com/forum/#!forum/lz4c
For hadoop-common-project/hadoop-common/src/main/native/gtest

---------------------------------------------------------------------
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the
following license:
re2j 1.1

---------------------------------------------------------------------

(GO license)
This is a work derived from Russ Cox's RE2 in Go, whose license
http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The binary distribution of this product bundles binaries of leveldbjni
(https://github.com/fusesource/leveldbjni), which is available under the
following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of FuseSource Corp. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-
nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from
the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==============

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy
(http://code.google.com/p/snappy/), which is available under the following
license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

--------------------------------------------------------------------------------

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCoding 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25
---------------------------------------------------------------------------------------------------------------------

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)
---------------------------------------------------------------------------------------------------------------------
Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js

--------------------------------------------------------------------------------

The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js

--------------------------------------------------------------------------------

The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery
- Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
- Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

For:
oj! Algorithms - version 43.0


This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as
documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

All files located in the node_modules and external directories are
externally maintained libraries used by this software which have their
own licenses; we recommend you read them, as their terms may differ from
the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js

-------------------------------------------------------------------------------------------------------------------------
Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

D3 is available under a 3-clause BSD license. For details, see:
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
- HSQLDB Database 2.3.4

(HSQL License)

"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group."

The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means
having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale,
and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.
3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such
Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.
6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in
48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
gizzly-http 2.2.21
gizzly-http-server 2.2.21
gizzly-http-servlet 2.2.21
gizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such...
combination), to make, use, sell, offer for sale, have made, and/or otherwise
dispose of: (1) Modifications made by that Contributor (or portions thereof);
and (2) the combination of Modifications made by that Contributor with its
Contributor Version (or portions of such combination).
(c) The licenses granted
in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first
distributes or otherwise makes the Modifications available to a third
party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2)
for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with
other software (except as part of the Contributor Version) or other devices; or
(3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or
otherwise make available in Executable form must also be made available in
Source Code form and that Source Code form must be distributed only under the
terms of this License. You must include a copy of this License with every copy
of the Source Code form of the Covered Software You distribute or otherwise make
available. You must inform recipients of any such Covered Software in Executable
form as to how they can obtain such Covered Software in Source Code form in a
reasonable manner on or through a medium customarily used for software
exchange.

3.2. Modifications.
The Modifications that You create or to which
You contribute are governed by the terms of this License. You represent that You
believe Your Modifications are Your original creation(s) and/or You have
sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that
identifies You as the Contributor of the Modification. You may not remove or
alter any copyright, patent or trademark notices contained within the Covered
Software, or any notices of licensing or any descriptive text giving attribution
to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source
Code form that alters or restricts the applicable version of this License or the
recipients' rights hereunder. You may choose to offer, and to charge a fee for,
warranty, support, indemnity or liability obligations to one or more recipients
of Covered Software. However, you may do so only on Your own behalf, and not on
behalf of the Initial Developer or any Contributor. You must make it absolutely
clear that any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial Developer
and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this
5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.
7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control
laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:
- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04

By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation
distributed under this Agreement, and
b) in the case of each subsequent Contributor:
  i) changes to the Program, and
  ii) additions to the Program;
where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution 'originates' from a
Contributor if it was added to the Program by such Contributor itself or anyone
acting on such Contributor's behalf. Contributions do not include additions to
the Program which: (i) are separate modules of software distributed in
conjunction with the Program under their own license agreement, and (ii) are not
derivative works of the Program.
“Contributor” means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when
combined with the Program.

“Program” means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free copyright license to
reproduce, prepare derivative works of, publicly display, publicly perform,
distribute and sublicense the Contribution of such Contributor, if any, and such
derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed
Patents to make, use, sell, offer to sell, import and otherwise transfer the
Contribution of such Contributor, if any, in source code and object code form.
This patent license shall apply to the combination of the Contribution and the
Program if, at the time the Contribution is added by the Contributor, such
addition of the Contribution causes such combination to be covered by the
Licensed Patents. The patent license shall not apply to any other combinations
which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to
its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other intellectual
property rights of any other entity. Each Contributor disclaims any liability to
Recipient for claims brought by any other entity based on infringement of
intellectual property rights or otherwise. As a condition to exercising the
rights and licenses granted hereunder, each Recipient hereby assumes sole
responsibility to secure any other intellectual property rights needed, if any.
For example, if a third party patent license is required to allow Recipient to
distribute the Program, it is Recipient's responsibility to acquire that license
before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively “Losses”) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product.
offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

(2-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or
publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients’ exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., “French translation of the Work by Original Author,” or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS
7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:
  jamon-runtime 2.4.1

(MPL 2.0)
Mozilla Public License
1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or
modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and
apparatus claims, in any patent Licensable by such Contributor that would be
infringed, but for the grant of the License, by the making, using, selling,
offering for sale, having made, import, or transfer of either its Contributions
or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General
Public License, Version 2.1, the GNU Affero General Public License, Version 3.0,
or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For
legal entities, You includes any entity that controls, is controlled by,
or is under common control with You. For purposes of this definition,
control means (a) the power, direct or indirect, to cause the direction or
management of such entity, whether by contract or otherwise, or (b) ownership of
more than fifty percent (50%) of the outstanding shares or beneficial ownership
of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive
license:

under intellectual property rights (other than patent or trademark) Licensable
by such Contributor to use, reproduce, make available, modify, display, perform,
distribute, and otherwise exploit its Contributions, either on an unmodified
basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have
made, import, and otherwise transfer either its Contributions or its Contributor
Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become
effective for each Contribution on the date the Contributor first distributes
such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form


All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You
must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including,
without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version
10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then you may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

******************************************************************************

/*--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/
The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4
--------------------------------------------------------------------------------
This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or
broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and
to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.

 e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You
of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the
extent reasonably practicable, the URI, if any, that Licensor
specifies to be associated with the Work, unless such URI does not
refer to the copyright notice or licensing information for the Work;
and (iv), consistent with Section 3(b), in the case of an Adaptation,
a credit identifying the use of the Work in the Adaptation (e.g.,
"French translation of the Work by Original Author," or "Screenplay
based on original Work by Original Author"). The credit required by
this Section 4(b) may be implemented in any reasonable manner;
provided, however, that in the case of an Adaptation or Collection, at
a minimum such credit will appear, if a credit for all contributing
authors of the Adaptation or Collection appears, then as part of these
credits and in a manner at least as prominent as the credits for the
other contributing authors. For the avoidance of doubt, You may only
use the credit required by this Section for the purpose of attribution
in the manner set out above and, by exercising Your rights under this
License, You may not implicitly or explicitly assert or imply any
connection with, sponsorship or endorsement by the Original Author,
Licensor and/or Attribution Parties, as appropriate, of You or Your
use of the Work, without the separate, express prior written
permission of the Original Author, Licensor and/or Attribution
Parties.
c. Except as otherwise agreed in writing by the Licensor or as may be
otherwise permitted by applicable law, if You Reproduce, Distribute or
Publicly Perform the Work either by itself or as part of any
Adaptations or Collections, You must not distort, mutilate, modify or
take other derogatory action in relation to the Work which would be
prejudicial to the Original Author's honor or reputation. Licensor
agrees that in those jurisdictions (e.g. Japan), in which any exercise
of the right granted in Section 3(b) of this License (the right to
make Adaptations) would be deemed to be a distortion, mutilation,
modification or other derogatory action prejudicial to the Original
Author's honor and reputation, the Licensor will waive or not assert,
as appropriate, this Section, to the fullest extent permitted by the
applicable national law, to enable You to reasonably exercise Your
right under Section 3(b) of this License (right to make Adaptations)
but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR
OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY
KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE,
INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF
LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS,
WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION
OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

Open Source Used in DNAC Platform Antelope 9072
6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.


Copyright (C) 2007 The Guava Authors
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2383 istanbul-reports 3.0.2
1.2383.1 Available under license:

Copyright 2012-2015 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2384 @babel/plugin-transform-spread
7.13.0
1.2384.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2385 for-in 1.0.2

1.2385.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2386 raw-body 2.4.0
1.2386.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013-2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2387 object-keys 1.1.1
1.2387.1 Available under license:
The MIT License (MIT)

Copyright (C) 2013 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2388 readable-stream 2.3.7

1.2388.1 Available under license:

Node.js is licensed for use as follows:

```
****
Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
****
```

This license applies to parts of Node.js originating from the
https://github.com/joyent/node repository:

```
****
Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2389 pure-color 1.3.0 1.3.0
1.2389.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Nick Williams
Copyright (c) 2011 Heather Arthur <fayearthur@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2390 babel-runtime 6.26.0 6.26.0
1.2390.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2391 is-accessor-descriptor 1.0.0

1.2391.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2392 to-regex-range 5.0.1

1.2392.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2393 lockfile 1.0.4

1.2393.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.2394 source-map 0.2.0
1.2394.1 Available under license:

Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the names of the Mozilla Foundation nor the names of project
  contributors may be used to endorse or promote products derived from this
  software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2395 annotations 2.0.1
1.2395.1 Available under license:

Licenses
========

Good news! You may use Kdyby Framework under the terms of either
the New BSD License or the GNU General Public License (GPL) version 2 or 3.

The BSD License is recommended for most projects. It is easy to understand and it
places almost no restrictions on what you can do with the framework. If the GPL
fits better to your project, you can use the framework under this license.

You don't have to notify anyone which license you are using. You can freely
use Kdyby Framework in commercial projects as long as the copyright header remains intact.

New BSD License

---------------

Copyright (c) 2008 Filip Prochzka (http://filip-prochazka.com)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of "Kdyby Framework" nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

GNU General Public License

--------------------------

GPL licenses are very very long, so instead of including them here we offer you URLs with full text:

- [GPL version 2](http://www.gnu.org/licenses/gpl-2.0.html)
- [GPL version 3](http://www.gnu.org/licenses/gpl-3.0.html)
1.2396 json-stringify-safe 5.0.1

1.2396.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2397 core 3.1.1

1.2397.1 Available under license:

Found license 'Eclipse Public License 1.0' in 'All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0'

Found license 'Eclipse Public License 1.0' in 'Eclipse Public License Version 1.0 (&quot;EPL&quot;). A copy of the EPL is available'


Found license 'Eclipse Public License 1.0' in 'All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0 &quot;All Rights Reserved.&quot; + //NON-NLS-1S'

Found license 'Eclipse Public License 1.0' in 'All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0 compiler.copyright = Copyright IBM Corp 2000, 2006. All rights reserved.'

Found license 'Eclipse Public License 1.0' in 'All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0 which accompanies'

1.2398 qs 6.5.1 6.5.1

1.2398.1 Available under license:

Copyright (c) 2014 Nathan LaFreniere and other contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors

1.2399 kite-hadoop-compatibility 1.1.0

1.2399.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 Cloudera Inc.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 *   http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
1.2400 cross-spawn 7.0.3

1.2400.1 Available under license:

The MIT License (MIT)

Copyright (c) 2018 Made With MOXY Lda <hello@moxy.studio>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2401 eslint-config-airbnb-base 9.0.0

1.2401.1 Available under license:

MIT

1.2402 kotlin 1.4.0

1.2402.1 Available under license:

No license file was found, but licenses were detected in source scan.

["version":3,"file":"kotlin.js","sources":["wrapper.js","arrayUtils.js","callableReferenceUtils.js","conversions.js","core.js","long.js","markerFunctions.js","misc.js","polyfills.js","rtti.js","runtime/arrayUtils.kt","runtime/Enum.kt","primitiveCompanionObjects.kt","generated/_Arrays.kt","generated/_Ranges.kt","kotlin/UByte.kt","kotlin/UInt.kt","kotlin/

---

Open Source Used In DNAC Platform Antelope 9086
Open Source Used In DNAC Platform Antelope 9088
Kotlin.toShort = function (a) {
  return (a & 0xFFFF) << 16;
};

Kotlin.toByte = function (a) {
  return (a & 0xFF) << 24 >> 24;
};

Kotlin.toChar = function (a) {
  return a & 0xFFFF;
};

Kotlin.numberToLong = function (a) {
  return (a instanceof Kotlin.Long ? a : Kotlin.Long.fromNumber(a)).numberToLong();
};

Kotlin.numberToInt = function (a) {
  return a instanceof Kotlin.Long ? a.toInt() : Kotlin.doubleToInt(a);
};

Kotlin.numberToShort = function (a) {
  return Kotlin.toShort(Kotlin.numberToInt(a));
};

Kotlin.numberToByte = function (a) {
  return Kotlin.toByte(Kotlin.numberToInt(a));
};

Kotlin.numberToDouble = function (a) {
  return a instanceof Kotlin.Long ? a : Kotlin.Long.fromNumber(a);
};

Kotlin.numberToChar = function (a) {
  return a & 0xFFFF;
};

Kotlin.equals = function (obj1, obj2) {
  if (obj1 == null) {
    return obj2 == null;
  }
  if (obj2 == null) {
    return false;
  }
  if (obj1 !== obj1) {
    return obj2 !== obj2;
  }
  if (typeof obj1 === "object" && typeof obj1.equals === "function") {
    return obj1.equals(obj2);
  }
  if (typeof obj1 === "number" && typeof obj2 === "number") {
    return obj1 === obj2 && (obj1 !== 0 || 1 / obj1 === 1 / obj2);
  }
  return obj1 === obj2;
};

Kotlin.hashCode = function (obj) {
  if (obj == null) {
    return 0;
  }
  var str = String(obj);
  return "[..."]
};

Kotlin.toString = function (o) {
  if (o == null) {
    return "null";
  }
  else if (Kotlin.isArrayish(o)) {
    return "[...]
";
  }
  return o.toString();
};

/**
 * Constructs a 64-bit two's-complement integer, given its low and high 32-bit values as signed integers.
 * See the from* functions below for more convenient ways of constructing Longs.
 */

Open Source Used In DNAC Platform Antelope 9089
are MIN_VALUE\n
* (-2^63) because -MIN_VALUE == MIN_VALUE (since 2^63 cannot be represented as\n * a positive number, it overflows back into a negative). Not handling this\n * case would often result in infinite recursion.\n
* @param {number} low The low (signed) 32 bits of the long.\n
* @param {number} high The high (signed) 32 bits of the long.\n
* @constructor

* @final

*/

Kotlin.Long = function(low, high) {
  /**
   *
   * @type {number}
   * @private
   */
  this.low_ = low | 0;  // force into 32 signed bits.

  /**
   * @type {number}
   * @private
   */
  this.high_ = high | 0;  // force into 32 signed bits.
};

Kotlin.Long.$metadata$ =

{ 
    kind: "class",
    simpleName: "Long",
    interfaces:

};

// NOTE: Common constant values
ZERO, ONE, NEG_ONE, etc. are defined below the\n// from* methods on which they depend.\n

/**
 * A cache of the Long representations of small integer values.\n * @type {number}
 * @private
 */
Kotlin.Long.IntCache_ = {};

/**
 * Returns a Long representing the given (32-bit) integer value.\n * @param {number} value The\n * 32-bit integer in question.\n *
 * @return {Kotlin.Long} The corresponding Long value.
 */
Kotlin.Long.fromInt = function(value) {
  if (-128 <= value && value < 128) {
    var cachedObj = Kotlin.Long.IntCache_[value];
    if (cachedObj) {
      return cachedObj;
    }
  }

  var obj = new Kotlin.Long(value | 0, value < 0 ? -1 : 0);
  if (-128 <= value && value < 128) {
  }
  return obj;
};

/**
 * Converts this number value to 'Long'.\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this 'Double' value is 'NaN', 'Long.MIN_VALUE' if it's less than 'Long.MIN_VALUE',\n * 'Long.MAX_VALUE' if it's bigger than 'Long.MAX_VALUE'.\n * @param {number} value The number in\n * question.\n *
 * @return {Kotlin.Long} The corresponding Long value.
 */
Kotlin.Long.fromNumber = function(value) {
  if (isNaN(value)) {
    return Kotlin.Long.ZERO;
  } else if (value <= -Kotlin.Long.TWO_PWR_63_DBL_) {
    return Kotlin.Long.MIN_VALUE;
  } else if (value + 1 >= Kotlin.Long.TWO_PWR_63_DBL_) {
    return Kotlin.Long.MAX_VALUE;
  } else if (value < 0) {
    return Kotlin.Long.fromNumber(-value).negate();
  } else {
    return new Kotlin.Long((value % Kotlin.Long.TWO_PWR_32_DBL_) | 0, (value / Kotlin.Long.TWO_PWR_32_DBL_) | 0);
  }
};

/**
 * Returns a Long representation of the given string, written using the given\n * radix.\n * @param {string} str The textual\n * representation of the Long.\n * @param {number=} opt_radix The radix in which the text is written.\n *
 * @return {Kotlin.Long} The corresponding Long value.\n */
Kotlin.Long.fromString = function(str, opt_radix) {
  if (str.length == 0) {
    throw Error('number format error: empty string');
  }

  var radix = opt_radix || 10;
  if (radix < 2 || 36 < radix) {
    throw Error('radix out of range: ' + radix);
  }

  if (str.charAt(0) == '-') {
    return Kotlin.Long.fromString(str.substring(1), radix).negate();
  } else if (str.indexOf('-') >= 0) {
    throw Error('number format error: interior "-" character: ' + str);
  }

  // Do several (8) digits each time through the loop, so as to minimize the calls to the very expensive emulated div.:\n
  // var radixToPower = Kotlin.Long.fromNumber(Math.pow(radix, 8));\n
  var var size = Math.min(8, str.length - i);\n  var var value = parseInt(str.substring(i, i + size), radix);\n  if (size < 8) {
    var size = Kotlin.Long.fromNumber(Math.pow(radix, size));\n    result = result.multiply(power).add(Kotlin.Long.fromNumber(value));\n  } else {
    var size = result = result.multiply(radixToPower);\n    result = result.add(Kotlin.Long.fromNumber(value));\n    }\n  }\n
  // Do several (8) digits each time through the loop, so as to minimize the calls to the very expensive emulated div.:\n
  return result;
};

// NOTE: the compiler should inline these constant values below and then remove\n// these variables, so should be no runtime penalty for these.\n
*/

*/

Open Source Used In DNAC Platform Antelope 9090
Long divided by the given one.!
* @param {Kotlin.Long} other Long by which to divide.
* @return

[Kotlin.Long] This Long divided by the given one.
* An Kotlin.Long.prototype.div = function(other) {
  if (other.isZero()) {
    throw Error('division by zero');
  } else if (this.isZero()) {
    return Kotlin.Long.ZERO;
  } else if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return Kotlin.Long.MIN_VALUE; // recall that -MIN_VALUE
    } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
      return Kotlin.Long.ONE;
  } else {
    // At this point, we have |other| > 2, so |this/other| < |MIN_VALUE|.
    var halfThis = 
      this.shiftRight(1);
    var approx = halfThis.div(other).shiftLeft(1);
    if (approx.equalsLong(Kotlin.Long.ZERO)) {
      return other.isNegative() ? Kotlin.Long.ONE : Kotlin.Long.NEG_ONE;
    } else {
      return this.negate().div(other.negate());
    }
  }

  if (this.isNegative()) {
    if (other.isNegative()) {
      return this.negate().div(other.negate());
    } else {
      return this.div(other).negate();
    }
  } else if (other.isNegative()) {
    return this.div(other.negate());
  }

  // Repeat the following until the remainder is less than other: find an
  // floating-point that approximates remainder / other *from below*, add this
  // into the result, and subtract it from the remainder. It is critical that
  // the approximate value is less than or equal to the real value so that then
  // remainder never becomes negative.
  var res = Kotlin.Long.ZERO;
  var rem = this;
  while (rem.greaterThanOrEqual(other)) {
    // Approximate the result of division. This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.
    var log2 = Math.ceil(Math.log(approx) / Math.LN2); // Decrease the approximation until it is smaller than the remainder. Note
    // that if it is too large, the product overflows and is negative.
    var approxRes = Kotlin.Long.fromNumber(approx);
    var approxRem = approxRes.multiply(other);
    while (approxRem.isNegative() || approxRem.greaterThan(rem)) {
      approx -= delta;
      approxRes = Kotlin.Long.fromNumber(approx);
      approxRem = approxRes.multiply(other);
    }
    // We know the answer can't be zero... and actually, zero would cause
    // infinite recursion since we would make no progress.
    if (approxRes.isZero()) {
      approxRes = Kotlin.Long.ZERO;
    }
    res = res.add(approxRes);
    rem = rem.subtract(approxRem);
  }
  return

  if (other.isNegative()) {
    return other.isNegative() ? Kotlin.Long.ONE : Kotlin.Long.NEG_ONE;
  } else {
    return other.isZero() ? Kotlin.Long.ZERO : other;
  }

  // This Long modulo the given one.
  * An Kotlin.Long.prototype.modulo = function(other) {
    return this.subtract(this.div(other).multiply(other));
  }

  // This Long with bits shifted to the left by the given amount.
  * @param {number} numBits The number of bits by which to shift
  * @return

  * An Kotlin.Long.prototype.shiftLeft = function(numBits) {
    numBits &= 63;
    if (numBits == 0) {
      return this;
    } else {
      var low = this.low_;
      if (numBits < 32) {
        var high = this.high_;
        return Kotlin.Long.fromBits(low << numBits, high << numBits);
      } else

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. 
 * Use of this source code is governed by the Apache 2.0 license that can be
 * found in the license/LICENSE.txt file.
 */

Kotlin.Long.prototype.equals = function (other) {
    return other instanceof Kotlin.Long &&
    this.equalsLong(other);
};

Kotlin.defineModule = function (id, declaration) {
};

Kotlin.defineInlineFunction = function(tag, fun) {
    return fun;
};

Kotlin.wrapFunction = function(fun) {
    var f = function() {
        f = fun();
        return
    };
    return function() {
        return f.apply(this, arguments);
    };
};

Kotlin.isTypeOf = function(type) {
    return function (object) {
        return typeof object === type;
    }
};

Kotlin.isInstanceOf = function (klass) {
    return function (object) {
        return Kotlin.isType(object,
                          klass);
    }
};

Kotlin.orNull = function (fn) {
    return function (object) {
        return object == null ||
        fn(object);
    }
};

Kotlin.andPredicate = function (a, b) {
    return function (object) {
        return a(object) 
        && b(object);
    }
};

Kotlin.kotlinModuleMetadata = function (abiVersion, moduleName, data)
{
};

Kotlin.suspendCall = function(value) {
    return value;
};

Kotlin.coroutineResult = function(qualifier)
{
    throwMarkerError();
};

Kotlin.coroutineController = function(qualifier) {
    throwMarkerError();
};

Kotlin.coroutineReceiver = function(qualifier) {
    throwMarkerError();
};

Kotlin.setCoroutineResult = function(value, qualifier) {
    throwMarkerError();
};

Kotlin.getLongFromBits = function(numBits, high, low)
{
    return Kotlin.Long.fromBits(0, low << (numBits - 32));
}
Math.exp(+x), b = Math.exp(-x);
                return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);
result -= (x * x * x) / 3;
                }
                return result;
            }
            else {
                var a =
                (Math.abs(x) < taylor_n_bound) {
                var result = x;
                if (Math.abs(x) > taylor_2_bound) {

                y1 = 1 / y;
            if (!isFinite(y) || !isFinite(y1)) return Math.exp(Math.abs(x) - Math.LN2);
            return (y +
                (typeof Math.cosh === "undefined") {
        Math.cosh = function(x) {
            var y = Math.exp(x);
            var
        };
    }
    if
Math.exp(x);
                var y1 = 1 / y;
            if (!isFinite(y)) return Math.exp(x - Math.LN2);
        };
    }
    if
result = (x * x * x) / 3;
        }]
        return result;
        }
        return Math.exp(x);
    }

    Kotlin.imul = Math.imul || imul;
    Kotlin.imulEmulated = imul;
    function imul(a, b) {
        return ((a & 0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0)) | 0;
    }
    function imul(a, b) {
        return (a & 0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0); 0;
    }
    function imul(a, b) {
        return (a & 0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0); 0;
    }
    function imul(a, b) {
        return (a & 0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0); 0;
    }
    function imul(a, b) {
        return (a & 0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0); 0;
    }
}
Object) { \n    switch (typeof object) { \n        case 'string': \n            return true; \n        case 'number': \n            return true; \n        case 'boolean': \n            return true; \n        case 'function': \n            return true; \n        default: \n            return object instanceof Object; \n    }
}

if (object == null || klass == null || (typeof object !== 'object' && typeof object !== 'function')) {
    return false;
}

if (typeof klass === 'function' && object instanceof klass) {
    return true;
}

var proto = Object.getPrototypeOf(klass);
var constructor = proto != null ? proto.constructor : null;
if (constructor != null && metastdata$ in constructor) { \n    var metadata = constructor.$metadata$; \n    return Object(kotlin.isObject = function (a) { \n        return typeof a === 'object' || a instanceof Kotlin.Long; \n    });
}

Kotlin.isChar = function (value) {
    return value instanceof Kotlin.BoxedChar;
};

Kotlin.isComparable = function (value) {
    var type = typeof value;
    return type === 'string' ||
           type === 'boolean' ||
           Kotlin.isNumber(value) ||
           Kotlin.isType(value, Kotlin.kotlin.Comparable);
};

Kotlin.isCharSequence = function (value) {
    return typeof value === 'string' ||
           Kotlin.isType(value, Kotlin.kotlin.CharSequence);
};

if (@PublishedApi)
external internal fun <T> Array(size: Int): Array<T>

if (@JsName("newArray")
fun <T> newArray(size: Int, initValue: T) = fillArrayVal(Array<T>(size), initValue);

if (@JsName("newArrayF")
inline fun <T> arrayWithFun(size: Int, init: (Int) -> T) = fillArrayFun(array<T>(size), init);

if (@JsName("fillArray")
inline fun <T> fillArrayFun(array: Array<T>, init: (Int) -> T): Array<T> {
    for (i in 0..array.size - 1) {
        array[i] = init(i)
    }
    return array;
}

if (@JsName("booleanArray")
fun booleanArray(size: Int, init: dynamic): Array<Boolean> {
    val result: dynamic = Array<Boolean>(size)
    result.`$type$` = "BooleanArray"
    return when (init) {
        null, true -> fillArrayVal(result, false)
        false -> result
        else -> fillArrayFun<Boolean>(result, init);
    }
}

if (@JsName("booleanArrayF")
inline fun booleanArrayWithFun(size: Int, init: (Int) -> Boolean): Array<Boolean> {
    val array = booleanArray(size, false)
    for (i in 0..array.size - 1) {
        @Suppress("UNUSED_VARIABLE") // used in js block
        val value = init(i)
        js("array[i] = value;\")
    }
    return array;
}

if (@JsName("charArray")
@Suppress("UNUSED_PARAMETER")
fun charArray(size: Int, init: dynamic): Array<Char> {
    val result = js("new Uint16Array(size)")
    result.`$type$` = "CharArray"
    return when (init) {
        null, true, false -> result // For consistency
        else -> fillArrayFun<Char>(result, init);
    }
}

if (@JsName("charArrayF")
inline fun charArrayWithFun(size: Int, init: (Int) -> Char): Array<Char> {
    val array = charArray(size, null)
    for (i in 0..array.size - 1) {
        @Suppress("UNUSED_PARAMETER") // used in js block
        js("array[i] = value;\")
    }
    return array;
}

if (@JsName("untypedCharArray")
fun longArray(size: Int, init: dynamic): Array<Long> {
    val result: dynamic = Array<Long>(size)
    result.`$type$` = "LongArray"
    return when (init) {
        null, true, false -> result // For consistency
        else -> fillArrayFun<Long>(result, init);
    }
}

if (@JsName("longArrayF")
inline fun longArrayWithFun(size: Int, init: (Int) -> Long): Array<Long> {
    val array = longArray(size, null)
    for (i in 0..array.size - 1) {
        @Suppress("UNUSED_VARIABLE") // used in js block
        js("array[i] = value;\")
    }
    return array;
}
In
  get() = _ordinal
  override fun compareTo(other: Enum<T>) = ordinal.compareTo(other.ordinal)

override fun equals(other: Any?) = this === other

override fun hashCode(): Int = js("Kotlin.identityHashCode")(this)

override fun toString() = name

companion object
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("ArraysKt")

package kotlin.collections

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> Array<out T>.component1(): T {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun ByteArray.component1(): Byte {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun ShortArray.component1(): Short {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun IntArray.component1(): Int {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun LongArray.component1(): Long {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun FloatArray.component1(): Float {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component1(): Double {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component1(): Boolean {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component1(): Char {
    return get(0)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> Array<out T>.component2(): T {
    return get(1)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun ByteArray.component2(): Byte {
    return get(1)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun ShortArray.component2(): Short {
    return get(1)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun IntArray.component2(): Int {
    return get(1)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun LongArray.component2(): Long {
    return get(1)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun FloatArray.component2(): Float {
    return get(1)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component2(): Double {
    return get(1)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component2(): Boolean {
    return get(1)
}

/**
 * Returns 2nd *element* from the array.
 * 
 * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component2(): Char {
    return get(1)
}

Open Source Used In DNAC Platform Antelope  9099
Valuable information about Kotlin is also included in this file. It is structured as follows:

- **IndexOutOfBoundsException** is described where the behavior is unspecified.
- Kotlin/JS provides some examples of how to use these features:
  - `LongArray` is used to demonstrate how to define a `component2()` method.
  - `FloatArray` is used to demonstrate how to define a `component2()` method.
  - `DoubleArray` is used to demonstrate how to define a `component2()` method.
  - `BooleanArray` is used to demonstrate how to define a `component2()` method.
  - `CharArray` is used to demonstrate how to define a `component2()` method.

Moreover, the file also provides a generic method to access the 3rd and 4th elements of any array type, demonstrating its flexibility and potential for reuse in different contexts.
 IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun ShortArray.inline operator 4(): Short {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun IntArray.component4(): Int {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun LongArray.component4(): Long {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun FloatArray.component4(): Float {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component4(): Double {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component4(): Boolean {
    return get(3)
}

/**
 * Returns 4th *element* from the array.
 * 
 * If the size of this array is less than 4, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component4(): Char {
    return get(3)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun <T> Array<out T>.component5(): T {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun ByteArray.component5(): Byte {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun ShortArray.component5(): Short {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun IntArray.component5(): Int {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun LongArray.component5(): Long {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun FloatArray.component5(): Float {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component5(): Double {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component5(): Boolean {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.

 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component5(): Char {
    return get(4)
}

/**
 * Returns `true` if [element] is found in the array.
 */
public operator fun
```kotlin
<@kotlin.internal.OnlyInputTypes T> Array<out T>.contains(element: T): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 */
public operator fun ByteArray.contains(element: Byte): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 */
public operator fun ShortArray.contains(element: Short): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 */
public operator fun IntArray.contains(element: Int): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 */
public operator fun LongArray.contains(element: Long): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it == element }' instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior as in a list.'", ReplaceWith("any { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@Suppress("DEPRECATION")
public operator fun FloatArray.contains(element: Float): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it == element }' instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior as in a list.'", ReplaceWith("any { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@Suppress("DEPRECATION")
public operator fun DoubleArray.contains(element: Double): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 */
public operator fun BooleanArray.contains(element: Boolean): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 */
public operator fun CharArray.contains(element: Char): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```
calling the [defaultValue] function if the [index] is out of bounds of this array.

byteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Byte): Byte { if (index >= 0 && index <= lastIndex) byteArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

shortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Short): Short { if (index >= 0 && index <= lastIndex) shortArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

intArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Int): Int { if (index >= 0 && index <= lastIndex) intArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

longArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Long): Long { if (index >= 0 && index <= lastIndex) longArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

floatArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Float): Float { if (index >= 0 && index <= lastIndex) floatArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

doubleArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Double): Double { if (index >= 0 && index <= lastIndex) doubleArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

booleanArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean { if (index >= 0 && index <= lastIndex) booleanArray[index] else defaultValue(index) } * Returns an element at the given [index] or ‘null’ if the [index] is out of bounds of this array.

charArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char { if (index >= 0 && index <= lastIndex) charArray[index] else defaultValue(index) } * Returns an element at the given [index] or ‘null’ if the [index] is out of bounds of this array.

ByteArray.elementAtOrNull(index: Int): Byte? { byteArray.getOrNull(index) } * Returns an element at the given [index] or ‘null’ if the [index] is out of bounds of this array.

ShortArray.elementAtOrNull(index: Int): Short? { shortArray.getOrNull(index) } * Returns an element at the given [index] or ‘null’ if the [index] is out of bounds of this array.

IntArray.elementAtOrNull(index: Int): Int? { intArray.getOrNull(index) } * Returns an element at the given [index] or ‘null’ if the [index] is out of bounds of this array.

LongArray.elementAtOrNull(index: Int): Long? { longArray.getOrNull(index) } * Returns an element at the given [index] or ‘null’ if the [index] is out of bounds of this array.

ByteArray.elementAtOrDefault(index: Int, defaultValue: (Int) -> Byte): Byte { if (index >= 0 && index <= lastIndex) byteArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

ShortArray.elementAtOrDefault(index: Int, defaultValue: (Int) -> Short): Short { if (index >= 0 && index <= lastIndex) shortArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

IntArray.elementAtOrDefault(index: Int, defaultValue: (Int) -> Int): Int { if (index >= 0 && index <= lastIndex) intArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

LongArray.elementAtOrDefault(index: Int, defaultValue: (Int) -> Long): Long { if (index >= 0 && index <= lastIndex) longArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

ByteArray.elementAtOrDefault(index: Int, defaultValue: (Int) -> Byte): Byte { if (index >= 0 && index <= lastIndex) byteArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

ShortArray.elementAtOrDefault(index: Int, defaultValue: (Int) -> Short): Short { if (index >= 0 && index <= lastIndex) shortArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

IntArray.elementAtOrDefault(index: Int, defaultValue: (Int) -> Int): Int { if (index >= 0 && index <= lastIndex) intArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

LongArray.elementAtOrDefault(index: Int, defaultValue: (Int) -> Long): Long { if (index >= 0 && index <= lastIndex) longArray[index] else defaultValue(index) } * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.
samples.collections.Collections.Elements.elementAtOrNull */\n@kotlin.internal.InlineOnly\n\npublic inline fun FloatArray.elementAtOrNull(index: Int): Float? {
    return this.getOrNull(index)\n}\n\n/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * @sample samples.collections.Collections.Elements.elementAtOrNull*/\n@kotlin.internal.InlineOnly\n\npublic inline fun DoubleArray.elementAtOrNull(index: Int): Double? {
    return this.getOrNull(index)\n}\n\n/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * @sample samples.collections.Collections.Elements.elementAtOrNull*/\n@kotlin.internal.InlineOnly\n\npublic inline fun BooleanArray.elementAtOrNull(index: Int): Boolean? {
    return this.getOrNull(index)\n}\n\n/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * @sample samples.collections.Collections.Elements.elementAtOrNull*/\n@kotlin.internal.InlineOnly\n\npublic inline fun CharArray.elementAtOrNull(index: Int): Char? {
    return this.getOrNull(index)\n}\n\n/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.find*/\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> Array<out T>.find(predicate: (T) -> Boolean): T? {
    return firstOrNull(predicate)\n}\n\n/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.find*/\n@kotlin.internal.InlineOnly\n\npublic inline fun ByteArray.find(predicate: (Byte) -> Boolean): Byte? {
    return firstOrNull(predicate)\n}\n\n/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.find*/\n@kotlin.internal.InlineOnly\n\npublic inline fun ShortArray.find(predicate: (Short) -> Boolean): Short? {
    return firstOrNull(predicate)\n}\n\n/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.find*/\n@kotlin.internal.InlineOnly\n\npublic inline fun IntArray.find(predicate: (Int) -> Boolean): Int? {
    return firstOrNull(predicate)\n}\n\n/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.find*/\n@kotlin.internal.InlineOnly\n\npublic inline fun LongArray.find(predicate: (Long) -> Boolean): Long? {
    return firstOrNull(predicate)\n}\n\n/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.find*/\n@kotlin.internal.InlineOnly\n\npublic inline fun FloatArray.find(predicate: (Float) -> Boolean): Float? {
    return firstOrNull(predicate)\n}\n\n/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.find*/\n@kotlin.internal.InlineOnly\n\npublic inline fun DoubleArray.find(predicate: (Double) -> Boolean): Double? {
    return firstOrNull(predicate)\n}\n\n/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.find*/\n@kotlin.internal.InlineOnly\n\npublic inline fun BooleanArray.find(predicate: (Boolean) -> Boolean): Boolean? {
    return firstOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun <T> Array<out T>.findLast(predicate: (T) -> Boolean): T? {
    return lastOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? {
    return lastOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun ShortArray.findLast(predicate: (Short) -> Boolean): Short? {
    return lastOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun IntArray.findLast(predicate: (Int) -> Boolean): Int? {
    return lastOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun LongArray.findLast(predicate: (Long) -> Boolean): Long? {
    return lastOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun FloatArray.findLast(predicate: (Float) -> Boolean): Float? {
    return lastOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun DoubleArray.findLast(predicate: (Double) -> Boolean): Double? {
    return lastOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {
    return lastOrNull(predicate)\n}\n\n/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * @sample samples.collections.Collections.Elements.findLast*/\n@kotlin.internal.InlineOnly\n\npublic inline fun CharArray.findLast(predicate: (Char) -> Boolean): Char? {
    return lastOrNull(predicate)\n}
samples.collections.Collections.Elements.find
*\n@kotlin.internal.InlineOnly\n@public inline fun
IntArray.findLast(predicate: (Int) -> Boolean): Int? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly\n@public inline fun
LongArray.findLast(predicate: (Long) -> Boolean): Long? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly\n@public inline fun
FloatArray.findLast(predicate: (Float) -> Boolean): Float? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly\n@public inline fun
BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly\n@public inline fun
CharArray.findLast(predicate: (Char) -> Boolean): Char? {
    return lastOrNull(predicate)
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun <T> Array<out T>.first(): T {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun ByteArray.first(): Byte {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun ShortArray.first(): Short {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun IntArray.first(): Int {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun LongArray.first(): Long {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun FloatArray.first(): Float {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun DoubleArray.first(): Double {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun BooleanArray.first(): Boolean {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun CharArray.first(): Char {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[0]
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun <T> Array<out T>.first(predicate: (T) -> Boolean): T {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun ByteArray.first(predicate: (Byte) -> Boolean): Byte {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun ShortArray.first(predicate: (Short) -> Boolean): Short {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun IntArray.first(predicate: (Int) -> Boolean): Int {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun LongArray.first(predicate: (Long) -> Boolean): Long {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun FloatArray.first(predicate: (Float) -> Boolean): Float {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun DoubleArray.first(predicate: (Double) -> Boolean): Double {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun BooleanArray.first(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun CharArray.first(predicate: (Char) -> Boolean): Char {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
public inline fun IntArray.first(predicate: (Int) -> Boolean): Int {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given 
 * predicate.
 * @throws NoSuchElementException if no such element is found.
 */
public inline fun LongArray.first(predicate: (Long) -> Boolean): Long {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given 
 * predicate.
 * @throws NoSuchElementException if no such element is found.
 */
public inline fun FloatArray.first(predicate: (Float) -> Boolean): Float {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given 
 * predicate.
 * @throws NoSuchElementException if no such element is found.
 */
public inline fun DoubleArray.first(predicate: (Double) -> Boolean): Double {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given 
 * predicate.
 * @throws NoSuchElementException if no such element is found.
 */
public inline fun BooleanArray.first(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given 
 * predicate.
 * @throws NoSuchElementException if no such element is found.
 */
public inline fun CharArray.first(predicate: (Char) -> Boolean): Char {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun <T> Array<out T>.firstOrNull(): T? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun ByteArray.firstOrNull(predicate: (Byte) -> Boolean): Byte? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun ShortArray.firstOrNull(predicate: (Short) -> Boolean): Short? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun IntArray.firstOrNull(predicate: (Int) -> Boolean): Int? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun LongArray.firstOrNull(predicate: (Long) -> Boolean): Long? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun FloatArray.firstOrNull(predicate: (Float) -> Boolean): Float? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun DoubleArray.firstOrNull(predicate: (Double) -> Boolean): Double? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun BooleanArray.firstOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun <T> Array<out T>.firstOrNull(predicate: (T) -> Boolean): T? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun ByteArray.firstOrNull(predicate: (Byte) -> Boolean): Byte? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun ShortArray.firstOrNull(predicate: (Short) -> Boolean): Short? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun IntArray.firstOrNull(predicate: (Int) -> Boolean): Int? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun LongArray.firstOrNull(predicate: (Long) -> Boolean): Long? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun FloatArray.firstOrNull(predicate: (Float) -> Boolean): Float? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun DoubleArray.firstOrNull(predicate: (Double) -> Boolean): Double? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun BooleanArray.firstOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given 
 * predicate, or `null` if no such element is found.
 */
public inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {
    for (element in this) if (predicate(element)) return element
    return null
}
element was not found.

*/
public inline fun DoubleArray.firstOrNull(predicate: (Double) -> Boolean): Double? {
    for (element in this) if (predicate(element)) return element
    return null
}

*/
public inline fun BooleanArray.firstOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    for (element in this) if (predicate(element)) return element
    return null
}

*/
public inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {
    for (element in this) if (predicate(element)) return element
    return null
}

*/
public inline fun Array<out T>.getOrElse(index: Int, defaultValue: (Int) -> T): T {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public inline fun ByteArray.getOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public inline fun ShortArray.getOrElse(index: Int, defaultValue: (Int) -> Short): Short {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public inline fun IntArray.getOrElse(index: Int, defaultValue: (Int) -> Int): Int {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public inline fun LongArray.getOrElse(index: Int, defaultValue: (Int) -> Long): Long {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public inline fun FloatArray.getOrElse(index: Int, defaultValue: (Int) -> Float): Float {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public inline fun DoubleArray.getOrElse(index: Int, defaultValue: (Int) -> Double): Double {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public inline fun BooleanArray.getOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public inline fun CharArray.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    if (index >= 0 && index <= lastIndex) return get(index)
    return defaultValue(index)
}

*/
public fun <T> Array<out T>.getOrNull(index: Int): T? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun ByteArray.getOrNull(index: Int): Byte? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun ShortArray.getOrNull(index: Int): Short? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun IntArray.getOrNull(index: Int): Int? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun LongArray.getOrNull(index: Int): Long? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun FloatArray.getOrNull(index: Int): Float? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun DoubleArray.getOrNull(index: Int): Double? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun BooleanArray.getOrNull(index: Int): Boolean? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun CharArray.getOrNull(index: Int): Char? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}

*/
public fun <T> Array<out T>.getOrNull(index: Int): T? {
    if (index >= 0 && index <= lastIndex) return get(index)
    return null
}
[index] or `null` if the [index] is out of bounds of this array.

```kotlin
return if (index >= 0 && index <= lastIndex) get(index) else null
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun DoubleArray.indexOf(element: Double): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
public inline fun <T> Array<out T>.indexOfFirst(predicate: (T) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun FloatArray.indexOf(element: Float): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun IntArray.indexOf(element: Int): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun LongArray.indexOf(element: Long): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun ShortArray.indexOf(element: Short): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun BooleanArray.indexOf(element: Boolean): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@Deprecated("The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.", ReplaceWith("indexOfFirst { it == element }"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharArray.indexOf(element: Char): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
public fun LongArray.getOrNull(index: Int): Long? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
public fun FloatArray.getOrNull(index: Int): Float? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
public fun DoubleArray.getOrNull(index: Int): Double? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
public fun BooleanArray.getOrNull(index: Int): Boolean? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns element at the given [index] or `null` if the [index] is out of bounds of this array.
 */
public fun CharArray.getOrNull(index: Int): Char? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.indexOf(element: T): Int {
    if (element == null) {
        for (index in indices) {
            if (this[index] == null) {
                return index
            }
        }
    } else {
        for (index in indices) {
            if (element == this[index]) {
                return index
            }
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun ByteArray.indexOf(element: Byte): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun ShortArray.indexOf(element: Short): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun IntArray.indexOf(element: Int): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun LongArray.indexOf(element: Long): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun FloatArray.indexOf(element: Float): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun DoubleArray.indexOf(element: Double): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun BooleanArray.indexOf(element: Boolean): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```

* * @sample

```kotlin
samples.collections.Collections.Elements.getOrNull

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
public fun CharArray.indexOf(element: Char): Int {
    for (index in indices) {
        if (element == this[index]) {
            return index
        }
    }
    return -1
}
```
**ByteArray.indexOfFirst**(predicate: (Byte) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**ShortArray.indexOfFirst**(predicate: (Short) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**IntArray.indexOfFirst**(predicate: (Int) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**LongArray.indexOfFirst**(predicate: (Long) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**ByteArray.indexOfLast**(predicate: (Byte) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**ShortArray.indexOfLast**(predicate: (Short) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**IntArray.indexOfLast**(predicate: (Int) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**LongArray.indexOfLast**(predicate: (Long) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**CharArray.indexOfFirst**(predicate: (Char) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

**CharArray.indexOfLast**(predicate: (Char) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}
CharArray.indexOfLast(predicate: (Char) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) return index
    }
    return -1
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun <T> Array<out T>.last(): T {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun ByteArray.last(): Byte {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun ShortArray.last(): Short {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun IntArray.last(): Int {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun LongArray.last(): Long {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun FloatArray.last(): Float {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun DoubleArray.last(): Double {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun BooleanArray.last(): Boolean {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun <T> Array<out T>.last(predicate: (T) -> Boolean): T {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun ByteArray.last(predicate: (Byte) -> Boolean): Byte {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun ShortArray.last(predicate: (Short) -> Boolean): Short {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun IntArray.last(predicate: (Int) -> Boolean): Int {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun LongArray.last(predicate: (Long) -> Boolean): Long {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun FloatArray.last(predicate: (Float) -> Boolean): Float {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun DoubleArray.last(predicate: (Double) -> Boolean): Double {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun BooleanArray.last(predicate: (Boolean) -> Boolean): Boolean {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
Returns the last element, or `null` if the array is empty.

```kotlin
public fun LongArray.lastOrNull(): Long? {
    return if (isEmpty()) null else this[size - 1]
}

/**
 * Returns the last element, or `null` if no such element was found.
 */
public fun FloatArray.lastOrNull(): Float? {
    return if (isEmpty()) null else this[size - 1]
}

/**
 * Returns the last element, or `null` if no such element was found.
 */
public fun DoubleArray.lastOrNull(): Double? {
    return if (isEmpty()) null else this[size - 1]
}

/**
 * Returns the last element, or `null` if no such element was found.
 */
public fun BooleanArray.lastOrNull(): Boolean? {
    return if (isEmpty()) null else this[size - 1]
}

/**
 * Returns the last element, or `null` if no such element was found.
 */
public fun CharArray.lastOrNull(): Char? {
    return if (isEmpty()) null else this[size - 1]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
public inline fun <T> Array<out T>.lastOrNull(predicate: (T) -> Boolean): T? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
public inline fun ByteArray.lastOrNull(predicate: (Byte) -> Boolean): Byte? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
public inline fun ShortArray.lastOrNull(predicate: (Short) -> Boolean): Short? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
public inline fun IntArray.lastOrNull(predicate: (Int) -> Boolean): Int? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
public inline fun LongArray.lastOrNull(predicate: (Long) -> Boolean): Long? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
public inline fun FloatArray.lastOrNull(predicate: (Float) -> Boolean): Float? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
public inline fun DoubleArray.lastOrNull(predicate: (Double) -> Boolean): Double? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
public inline fun CharArray.lastOrNull(predicate: (Char) -> Boolean): Char? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random).
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun ByteArray.random(): Byte {
    return random(Random).
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun ShortArray.random(): Short {
    return random(Random).
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun IntArray.random(): Int {
    return random(Random).
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun LongArray.random(): Long {
    return random(Random).
}
inline fun FloatArray.random(): Float {
    return random(Random())
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.random(): Double {
    return random(Random())
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun BooleanArray.random(): Boolean {
    return random(Random())
}

/**
 * Returns a random element from this array.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun CharArray.random(): Char {
    return random(Random())
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun <T> Array<out T>.random(random: Random): T {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun ByteArray.random(random: Random): Byte {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun ShortArray.random(random: Random): Short {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun IntArray.random(random: Random): Int {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun LongArray.random(random: Random): Long {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun FloatArray.random(random: Random): Float {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun DoubleArray.random(random: Random): Double {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun BooleanArray.random(random: Random): Boolean {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun CharArray.random(random: Random): Char {
    if (isEmpty()) throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.randomOrNull(): T? {
    return randomOrNull(Random())
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun ByteArray.randomOrNull(): Byte? {
    return randomOrNull(Random())
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun ShortArray.randomOrNull(): Short? {
    return randomOrNull(Random())
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun IntArray.randomOrNull(): Int? {
    return randomOrNull(Random())
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun LongArray.randomOrNull(): Long? {
    return randomOrNull(Random())
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun FloatArray.randomOrNull(): Float? {
    return randomOrNull(Random())
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun DoubleArray.randomOrNull(): Double? {
    return randomOrNull(Random())
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun BooleanArray.randomOrNull(): Boolean? {
    return randomOrNull(Random())
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun CharArray.randomOrNull(): Char? {
    return randomOrNull(Random())
}
c inline fun ShortArray.randomOrOrNull(): Short? {\n    return randomOrNull(Random)\n}\n
/**
 * Returns a random element from this array, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.randomOrOrNull(): Int? {\n    return randomOrNull(Random)\n}\n
/**
 * Returns a random element from this array, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.randomOrOrNull(): Long? {\n    return randomOrNull(Random)\n}\n
/**
 * Returns a random element from this array, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.randomOrOrNull(): Float? {\n    return randomOrNull(Random)\n}\n
/**
 * Returns a random element from this array, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.randomOrOrNull(): Double? {\n    return randomOrNull(Random)\n}\n
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> Array<out T>.randomOrOrNull(random: Random): T? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))\n}\n
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun ByteArray.randomOrOrNull(random: Random): Byte? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))\n}\n
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun ShortArray.randomOrOrNull(random: Random): Short? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))\n}\n
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun IntArray.randomOrOrNull(random: Random): Int? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))\n}\n
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun LongArray.randomOrOrNull(random: Random): Long? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))\n}\n
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun FloatArray.randomOrOrNull(random: Random): Float? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))\n}\n
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun DoubleArray.randomOrOrNull(random: Random): Double? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))\n}
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun BooleanArray.randomOrNull(random: Random): Boolean? {
    if (isEmpty())
        return null
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun CharArray.randomOrNull(random: Random): Char? {
    if (isEmpty())
        return null
    return get(random.nextInt(size))
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> Array<T>.single(): T {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.
")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.
")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun ByteArray.single(): Byte {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.
")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.
")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun ShortArray.single(): Short {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.
")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.
")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun IntArray.single(): Int {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.
")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.
")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun LongArray.single(): Long {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.
")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.
")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun BooleanArray.single(): Boolean {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.
")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.
")
    }
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
 */
public fun CharArray.single(): Char {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.
")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.
")
    }
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun <T> Array<T>.single(predicate: (T) -> Boolean): T {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.
")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.
")
    @Suppress("UNCHECKED_CAST")
    return single as T
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun ByteArray.single(predicate: (Byte) -> Boolean): Byte {
    var single: Byte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.
")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.
")
    @Suppress("UNCHECKED_CAST")
    return single as Byte
}
single element matching the given [predicate], or throws exception if there is no or more than one matching element.

```kotlin
@public inline fun ShortArray.single(predicate: (Short) -> Boolean): Short {
    var single: Short? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw
        NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Short
}

@public inline fun IntArray.single(predicate: (Int) -> Boolean): Int {
    var single: Int? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw
        NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Int
}

@public inline fun LongArray.single(predicate: (Long) -> Boolean): Long {
    var single: Long? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw
        NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Long
}

@public inline fun FloatArray.single(predicate: (Float) -> Boolean): Float {
    var single: Float? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw
        NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Float
}

@public inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {
    var single: Double? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw
        NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Double
}

@public inline fun BooleanArray.single(predicate: (Boolean) -> Boolean): Boolean {
    var single: Boolean? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw
        NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Boolean
}

@public inline fun CharArray.single(predicate: (Char) -> Boolean): Char {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw
                IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw
        NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as Char
}

public fun <T> Array<out T>.singleOrNull(): T? {
    return if (size == 1) this[0] else null
}

public fun ByteArray.singleOrNull(): Byte? {
    return if (size == 1) this[0] else null
}

public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

public fun DoubleArray.singleOrNull(): Double? {
    return if (size == 1) this[0] else null
}

public fun BooleanArray.singleOrNull(): Boolean? {
    return if (size == 1) this[0] else null
}

public fun CharArray.singleOrNull(): Char? {
    return if (size == 1) this[0] else null
}

public fun <T> Array<out T>.singleOrNull(): T? {
    return if (size == 1) this[0] else null
}

@public fun ByteArray.singleOrNull(): Byte? {
    return if (size == 1) this[0] else null
}

@public fun ShortArray.singleOrNull(): Short? {
    return if (size == 1) this[0] else null
}

@public fun DoubleArray.singleOrNull(): Double? {
    return if (size == 1) this[0] else null
}

@public fun BooleanArray.singleOrNull(): Boolean? {
    return if (size == 1) this[0] else null
}

@public fun CharArray.singleOrNull(): Char? {
    return if (size == 1) this[0] else null
}
```
fun IntArray.singleOrNull(): Int? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun LongArray.singleOrNull(): Long? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun ShortArray.singleOrNull(): Short? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun ByteArray.singleOrNull(): Byte? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun FloatArray.singleOrNull(): Float? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun DoubleArray.singleOrNull(): Double? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun CharArray.singleOrNull(): Char? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun BooleanArray.singleOrNull(): Boolean? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun IntArray.singleOrNull(predicate: (Int) -> Boolean): Int? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun ShortArray.singleOrNull(predicate: (Short) -> Boolean): Short? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun ByteArray.singleOrNull(predicate: (Byte) -> Boolean): Byte? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun FloatArray.singleOrNull(predicate: (Float) -> Boolean): Float? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun DoubleArray.singleOrNull(predicate: (Double) -> Boolean): Double? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun BooleanArray.singleOrNull(predicate: (Boolean) -> Boolean): Boolean? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun IntArray.singleOrNull(predicate: (Int) -> Boolean): Int? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun ShortArray.singleOrNull(predicate: (Short) -> Boolean): Short? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun ByteArray.singleOrNull(predicate: (Byte) -> Boolean): Byte? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun FloatArray.singleOrNull(predicate: (Float) -> Boolean): Float? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun DoubleArray.singleOrNull(predicate: (Double) -> Boolean): Double? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? { if (size == 1) return this[0] else null
}

/**
 * Returns single element, or null if the array is empty or has more than one element.
 */
public fun BooleanArray.singleOrNull(predicate: (Boolean) -> Boolean): Boolean? { if (size == 1) return this[0] else null
}
IllegalArgumentException if \( n \) is negative.\n
```java
public fun <T> Array<out T>.drop(n: Int): List<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun ByteArray.drop(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun ShortArray.drop(n: Int): List<Short> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun IntArray.drop(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun LongArray.drop(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun FloatArray.drop(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun DoubleArray.drop(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun BooleanArray.drop(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun CharArray.drop(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun <T> Array<out T>.dropLast(n: Int): List<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun ByteArray.dropLast(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun ShortArray.dropLast(n: Int): List<Short> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun IntArray.dropLast(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun LongArray.dropLast(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun FloatArray.dropLast(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun DoubleArray.dropLast(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun BooleanArray.dropLast(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```

```java
public fun CharArray.dropLast(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))\n}\n\n\nIllegalArgumentException if \( n \) is negative.\n
```
IllegalArgumentException if [n] is negative.

public fun IntArray.dropLast(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0)).unsorted
}

public fun LongArray.dropLast(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0)).unsorted
}

public fun FloatArray.dropLast(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0)).unsorted
}

public fun DoubleArray.dropLast(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0)).unsorted
}

public fun BooleanArray.dropLast(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0)).unsorted
}

public fun CharArray.dropLast(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0)).unsorted
}

public inline fun <T> Array<out T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

public inline fun ByteArray.dropLastWhile(predicate: (Byte) -> Boolean): List<Byte> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

public inline fun ShortArray.dropLastWhile(predicate: (Short) -> Boolean): List<Short> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

public inline fun IntArray.dropLastWhile(predicate: (Int) -> Boolean): List<Int> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

public inline fun LongArray.dropLastWhile(predicate: (Long) -> Boolean): List<Long> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

public inline fun FloatArray.dropLastWhile(predicate: (Float) -> Boolean): List<Float> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

public inline fun DoubleArray.dropLastWhile(predicate: (Double) -> Boolean): List<Double> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

public inline fun BooleanArray.dropLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

public inline fun CharArray.dropLastWhile(predicate: (Char) -> Boolean): List<Char> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

Open Source Used In DNAC Platform Antelope 9119
samples.collections.Collections.Transformations.dropWhile(predicate: (Double) -> Boolean): List<Double> {
    var yielding = false
    val list = ArrayList<Double>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
* Returns a list containing all elements except first elements that satisfy the given [predicate].
*
* @sample samples.collections.Collections.Transformations.dropWhile
*/
public inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {
    var yielding = false
    val list = ArrayList<Float>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
* Returns a list containing all elements except first elements that satisfy the given [predicate].
*
* @sample samples.collections.Collections.Transformations.dropWhile
*/
public inline fun DoubleArray.dropWhile(predicate: (Double) -> Boolean): List<Double> {
    var yielding = false
    val list = ArrayList<Double>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}

/**
* Returns a list containing all elements except first elements that satisfy the given [predicate].
*
* @sample samples.collections.Collections.Transformations.dropWhile
*/
public inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    var yielding = false
    val list = ArrayList<Boolean>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
}
samples.collections.Collections.Transformations.drop\n */\n\npublic inline fun CharArray.dropWhile(predicate: (Char) -> Boolean): List<Char> {\n var yielding = false \n val list = ArrayList<Char>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun CharArray.dropWhile(predicate: (Char) -> Boolean): List<Char> {\n var yielding = false \n val list = ArrayList<Char>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun ByteArray.dropWhile(predicate: (Byte) -> Boolean): List<Byte> {\n var yielding = false \n val list = ArrayList<Byte>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun ShortArray.dropWhile(predicate: (Short) -> Boolean): List<Short> {\n var yielding = false \n val list = ArrayList<Short>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {\n var yielding = false \n val list = ArrayList<Int>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun LongArray.dropWhile(predicate: (Long) -> Boolean): List<Long> {\n var yielding = false \n val list = ArrayList<Long>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {\n var yielding = false \n val list = ArrayList<Float>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun DoubleArray.dropWhile(predicate: (Double) -> Boolean): List<Double> {\n var yielding = false \n val list = ArrayList<Double>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n var yielding = false \n val list = ArrayList<Boolean>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {\n var yielding = false \n val list = ArrayList<Int>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun LongArray.dropWhile(predicate: (Long) -> Boolean): List<Long> {\n var yielding = false \n val list = ArrayList<Long>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {\n var yielding = false \n val list = ArrayList<Float>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun DoubleArray.dropWhile(predicate: (Double) -> Boolean): List<Double> {\n var yielding = false \n val list = ArrayList<Double>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n var yielding = false \n val list = ArrayList<Boolean>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {\n var yielding = false \n val list = ArrayList<Int>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun LongArray.dropWhile(predicate: (Long) -> Boolean): List<Long> {\n var yielding = false \n val list = ArrayList<Long>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {\n var yielding = false \n val list = ArrayList<Float>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun DoubleArray.dropWhile(predicate: (Double) -> Boolean): List<Double> {\n var yielding = false \n val list = ArrayList<Double>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
public inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n var yielding = false \n val list = ArrayList<Boolean>() \n for (item in this) {\n if (yielding) {\n list.add(item) \n } else if (!predicate(item)) {\n list.add(item) \n yielding = true \n } \n return list}\n
Open Source Used In DNAC Platform Antelope 9121
(index: Int, Long) -> Boolean): List<Long> {
    return filterIndexedTo(ArrayList<Long>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun 
FloatArray.filterIndexed(predicate: (index: Int, Float) -> Boolean): List<Float> {
    return filterIndexedTo(ArrayList<Float>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun DoubleArray.filterIndexed(predicate:
(index: Int, Double) -> Boolean): List<Double> {
    return filterIndexedTo(ArrayList<Double>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun BooleanArray.filterIndexed(predicate: (index: Int, Boolean) -> Boolean): List<Boolean> {
    return filterIndexedTo(ArrayList<Boolean>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun CharArray.filterIndexed(predicate:
(index: Int, Char) -> Boolean): List<Char> {
    return filterIndexedTo(ArrayList<Char>(), predicate)
}

/**
 * Appends all elements matching the given [predicate] to the given [destination],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <T, C : MutableCollection<in T>> Array<out T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Byte>> ByteArray.filterIndexedTo(destination: C, predicate:
(index: Int, Byte) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Short>> ShortArray.filterIndexedTo(destination: C, predicate:
(index: Int, Short) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Int>> IntArray.filterIndexedTo(destination: C, predicate:
(index: Int, Int) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination],
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Long>> LongArray.filterIndexedTo(destination: C, predicate:
(index: Int, Long) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}
predicate evaluation on the element.

```kotlin
/*public inline fun <C : MutableCollection<in Float>> FloatArray.filterIndexedTo(destination: C, predicate: (index: Int, Float) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}*/

public inline fun <C : MutableCollection<in Double>> DoubleArray.filterIndexedTo(destination: C, predicate: (index: Int, Double) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Char>> CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
 * Returns a list containing all elements that are instances of specified type parameter R.
 * @sample samples.collections.Collections.Filtering.filterIsInstance
 */
public inline fun <reified R> Array<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {
    return filterIsInstanceTo(ArrayList<R>())
}

/**
 * Appends all elements that are instances of specified type parameter R to the given [destination].
 * @sample samples.collections.Collections.Filtering.filterIsInstanceTo
 */
public inline fun <reified R, C : MutableCollection<in R>> Array<*>.filterIsInstanceTo(destination: C): C {
    for (element in this) if (element is R) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun <T> Array<out T>.filterNot(predicate: (T) -> Boolean): List<T> {
    return filterNotTo(ArrayList<T>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun ByteArray.filterNot(predicate: (Byte) -> Boolean): List<Byte> {
    return filterNotTo(ArrayList<Byte>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun ShortArray.filterNot(predicate: (Short) -> Boolean): List<Short> {
    return filterNotTo(ArrayList<Short>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun IntArray.filterNot(predicate: (Int) -> Boolean): List<Int> {
    return filterNotTo(ArrayList<Int>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun LongArray.filterNot(predicate: (Long) -> Boolean): List<Long> {
    return filterNotTo(ArrayList<Long>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun FloatArray.filterNot(predicate: (Float) -> Boolean): List<Float> {
    return filterNotTo(ArrayList<Float>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun DoubleArray.filterNot(predicate: (Double) -> Boolean): List<Double> {
    return filterNotTo(ArrayList<Double>(), predicate)
}
```

Open Source Used In DNAC Platform Antelope 9123
samples.collections.Collections.Filtering.filter

```java
public inline fun BooleanArray.filterNot(predicate: (Boolean) -> Boolean): List<Boolean>
```
Open Source Used In DNAC Platform Antelope 9125

```kotlin
Boolean): C {n for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n} \n/**
 * Appends all elements matching the given [predicate] to the given [destination].\n */ \n@sample samples.collections.Collections.Filtering.filterTo\n*\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {n for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n} \n/**
 * Appends all elements matching the given [predicate] to the given [destination].\n */ \n@sample samples.collections.Collections.Filtering.filterTo\n*\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterTo(destination: C, predicate: (Short) -> Boolean): C {n for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n} \n/**
 * Appends all elements matching the given [predicate] to the given [destination].\n */ \n@sample samples.collections.Collections.Filtering.filterTo\n*\npublic inline fun <C : MutableCollection<in Int>> IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {n for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n} \n/**
 * Appends all elements matching the given [predicate] to the given [destination].\n */ \n@sample samples.collections.Collections.Filtering.filterTo\n*\npublic inline fun <C : MutableCollection<in Long>> LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {n for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n} \n/**
 * Appends all elements matching the given [predicate] to the given [destination].\n */ \n@sample samples.collections.Collections.Filtering.filterTo\n*\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {n for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n} \n/**
 * Appends all elements matching the given [predicate] to the given [destination].\n */ \n@sample samples.collections.Collections.Filtering.filterTo\n*\npublic inline fun <C : MutableCollection<in Double>> DoubleArray.filterTo(destination: C, predicate: (Double) -> Boolean): C {n for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n} \n/**
 * Appends all elements matching the given [predicate] to the given [destination].\n */ \n@sample samples.collections.Collections.Filtering.filterTo\n*\npublic inline fun <C : MutableCollection<in Char>> CharArray.filterTo(destination: C, predicate: (Char) -> Boolean): C {n for (element in this) if (predicate(element)) destination.add(element)\n    return destination\n} \n/**
 * Returns a list containing elements at indices in the specified [indices] range.\n */ \n@sample samples.Collections.Filtering.slice\n*\npublic fun <T> Array<out T>.slice(indices: IntRange): List<T> {n if (indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n} \n/**
 * Returns a list containing elements at indices in the specified [indices] range.\n */ \n@sample samples.Collections.Filtering.slice\n*\npublic fun ByteArray.slice(indices: IntRange): List<Byte> {n if (indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n} \n/**
 * Returns a list containing elements at indices in the specified [indices] range.\n */ \n@sample samples.Collections.Filtering.slice\n*\npublic fun ShortArray.slice(indices: IntRange): List<Short> {n if (indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n} \n/**
 * Returns a list containing elements at indices in the specified [indices] range.\n */ \n@sample samples.Collections.Filtering.slice\n*\npublic fun IntArray.slice(indices: IntRange): List<Int> {n if (indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n} \n/**
 * Returns a list containing elements at indices in the specified [indices] range.\n */ \n@sample samples.Collections.Filtering.slice\n*\npublic fun LongArray.slice(indices: IntRange): List<Long> {n if (indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n} \n/**
 * Returns a list containing elements at indices in the specified [indices] range.\n */ \n@sample samples.Collections.Filtering.slice\n*\npublic fun DoubleArray.slice(indices: IntRange): List<Double> {n if (indices.isEmpty()) return listOf()\n    return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n} \n```

if (indices.isEmpty()) return listOff()  
return copyOfRange(indices.start, indices.endInclusive + 1).asList()  


FloatArray.sliceArray(indices: Collection<Int>): FloatArray {
    val result = FloatArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

/**
 * Returns an array containing elements at specified [indices].
 */
public fun FloatArray.slice(indices: Iterable<Int>): FloatArray {
    val result = FloatArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

LongArray.sliceArray(indices: Collection<Int>): LongArray {
    val result = LongArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

IntArray.sliceArray(indices: Collection<Int>): IntArray {
    val result = IntArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

ByteArray.sliceArray(indices: Collection<Int>): ByteArray {
    val result = ByteArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

Array<T>.sliceArray(indices: Collection<Int>): Array<T> {
    val result = arrayOfNulls<T>(this, indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

CharArray.sliceArray(indices: Collection<Int>): charArray {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyCharArray()
    val list = ArrayList>HelloTimer()
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Char> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Char>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List:Boolean> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList[Boolean](size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Double> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Double>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Float> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Float>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Long> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Long>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Int> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Int>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Short> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Short>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<Byte> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<Byte>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

List<T> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<T>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

Array<out T>.sliceArray(indices: Collection<Int>): List<T> {
    val result = ArrayList<T>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

CharArray.slice(indices: Collection<Int>): CharArray {
    val result = CharArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices)
return emptyArrayOf<T>(0)
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()


Open Source Used In DNAC Platform Antelope 9126
targetIndex = 0
for (sourceIndex in indices) {
    result[targetIndex++] = this[sourceIndex]
}
return result

* Returns an array containing elements of this array at specified [indices].

* //public fun DoubleArray.sliceArray(indices: Collection<Int>): DoubleArray {
    val result = DoubleArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

* //public fun BooleanArray.sliceArray(indices: Collection<Int>): BooleanArray {
    val result = BooleanArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

* //public fun CharArray.sliceArray(indices: Collection<Int>): CharArray {
    val result = CharArray(indices.size)
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

* //public fun <T> Array<T>.sliceArray(indices: IntRange): Array<T> {
    var targetIndex = 0
    for (sourceIndex in indices) {
        result[targetIndex++] = this[sourceIndex]
    }
    return result
}

* Returns an array containing elements of this array at specified [indices].

* @param index The index to retrieve.
* @return The element at the specified index.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.

* Returns an array containing elements at indices in the specified [indices] range.

* @param indices The indices to slice.
* @return The sliced array.
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun IntArray.take(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Int>(n)\n for
(item in this) {\n
list.add(item)\n
if (++count == n)\n
break\n }\n return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun LongArray.take(n: Int): List<Long> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Long>(n)\n for
(item in this) {\n
list.add(item)\n
if (++count == n)\n
break\n }\n return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun FloatArray.take(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Float>(n)\n for
(item in this) {\n
list.add(item)\n
if (++count == n)\n
break\n }\n return list\n}\n\n/**\n * Returns
a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun DoubleArray.take(n: Int): List<Double> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n if (n >=
size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list = ArrayList<Double>(n)\n
for (item in this) {\n
list.add(item)\n
if (++count == n)\n
break\n }\n return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.take\n */\npublic fun BooleanArray.take(n: Int):
List<Boolean> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return
emptyList()\n if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n val list =
ArrayList<Boolean>(n)\n for (item in this) {\n
list.add(item)\n
if (++count == n)\n
break\n }\n
return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun
CharArray.take(n: Int): List<Char> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if
(n == 0) return emptyList()\n if (n >= size) return toList()\n if (n == 1) return listOf(this[0])\n var count = 0\n
val list = ArrayList<Char>(n)\n for (item in this) {\n
list.add(item)\n
if (++count == n)\n
break\n
}\n return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> Array<out T>.takeLast(n: Int): List<T>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val
size = size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list =
ArrayList<T>(n)\n for (index in size - n until size)\n
list.add(this[index])\n return list\n}\n\n/**\n * Returns
a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun ByteArray.takeLast(n: Int): List<Byte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<Byte>(n)\n
for (index in size - n until size)\n
list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun ShortArray.takeLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n if (n == 0) return emptyList()\n val size =
size\n if (n >= size) return toList()\n if (n == 1) return listOf(this[size - 1])\n val list = ArrayList<Short>(n)\n
for (index in size - n until size)\n
list.add(this[index])\n return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun IntArray.takeLast(n: Int): List<Int> {\n

Open Source Used In DNAC Platform Antelope 9128


/**
 * Reverses elements in the array in-place.
 */
public fun ByteArray.reverse(): Unit {

    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
    }

    return list
}

/**
 * Reverses elements in the array in-place.
 */
public fun <T> Array<T>.reverse(): Unit {

    ArrayList<Char>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }

    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun CharArray.takeWhile(predicate: (Char) -> Boolean): List<Char> {
    var list = ArrayList<Char>()
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun ByteArray.takeLastWhile(predicate: (Byte) -> Boolean): List<Byte> {
    var list = ArrayList<Byte>()
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}
Unit { 
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun ByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun DoubleArray.reverse(): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun IntArray.reverse(): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun LongArray.reverse(): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun ShortArray.reverse(): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun CharArray.reverse(): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun BooleanArray.reverse(): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun FloatArray.reverse(): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}

/**
 * Reverses elements in the
 * array in-place. 
 */
public fun LongArray.reverse(): Unit {
val midPoint = (size / 2) - 1
if (midPoint < 0) return
var reverseIndex = lastIndex
for (index in 0..midPoint) { 
val tmp = this[index]
this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
}
}
array in the specified range in-place.

* @param fromIndex the start of the range (inclusive) to reverse.
* @param toIndex the end of the range (exclusive) to reverse.
* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

 */
@SinceKotlin("1.4")
public fun IntArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun LongArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun FloatArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun DoubleArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun BooleanArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Returns a
list with elements in reversed order.

*\*anonymous fun <T> Array<out T>.reversed(): List<T> { 
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
} *\*anonymous function ByteArray.reversed(): List<Byte> { 
    if (isEManaged()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
} *\*anonymous function ShortArray.reversed(): List<Short> { 
    if (isEManaged()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
} *\*anonymous function LongArray.reversed(): List<Long> { 
    if (isEManaged()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
} *\*anonymous function DoubleArray.reversed(): List<Double> { 
    if (isEManaged()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
} *\*anonymous function BooleanArray.reversed(): List<Boolean> { 
    if (isEManaged()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
} *\*anonymous function CharArray.reversed(): List<Char> { 
    if (isEManaged()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
} *\*anonymous function ByteArray.reversed(): List<Byte> { 
    if (isEManaged()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
} *\*anonymous fun <T> Array<T>.reversed(): Array<T> { 
    if (isEManaged()) return this
    val result = arrayOfNulls(this)
    val lastIndex = lastIndexOf
    for (i in 0..lastIndex) result[lastIndex - i] = this[i]
    return result
} *\*anonymous fun ByteArray.reversed(): Array<Byte> { 
    if (isEManaged()) return this
    val result = ByteArray(size)
    val lastIndex = lastIndexOf
    for (i in 0..lastIndex) result[lastIndex - i] = this[i]
    return result
} *\*anonymous fun ShortArray.reversed(): ShortArray { 
    if (isEManaged()) return this
    val result = ShortArray(size)
    val lastIndex = lastIndexOf
    for (i in 0..lastIndex) result[lastIndex - i] = this[i]
    return result
} *\*anonymous fun LongArray.reversed(): LongArray { 
    if (isEManaged()) return this
    val result = LongArray(size)
    val lastIndex = lastIndexOf
    for (i in 0..lastIndex) result[lastIndex - i] = this[i]
    return result
} *\*anonymous fun DoubleArray.reversed(): DoubleArray { 
    if (isEManaged()) return this
    val result = DoubleArray(size)
    val lastIndex = lastIndexOf
    for (i in 0..lastIndex) result[lastIndex - i] = this[i]
    return result
} *\*anonymous fun BooleanArray.reversed(): BooleanArray { 
    if (isEManaged()) return this
    val result = BooleanArray(size)
    val lastIndex = lastIndexOf
    for (i in 0..lastIndex) result[lastIndex - i] = this[i]
    return result
} *\*anonymous fun CharArray.reversed(): CharArray { 
    if (isEManaged()) return this
    val result = CharArray(size)
    val lastIndex = lastIndexOf
    for (i in 0..lastIndex) result[lastIndex - i] = this[i]
    return result
} *\*anonymous fun ByteArray.reversed(): ByteArray { 
    if (isEManaged()) return this
    val result = ByteArray(size)
    val lastIndex = lastIndexOf
    for (i in 0..lastIndex) result[lastIndex - i] = this[i]
    return result
}
shuffles elements in this array in-place.

```kotlin
@SinceKotlin("1.4")
public fun LongArray.shuffle(): Unit {  
shuffle(Random)
}@SinceKotlin("1.4")
public fun FloatArray.shuffle(): Unit {  
shuffle(Random)
}@SinceKotlin("1.4")
public fun DoubleArray.shuffle(): Unit {  
shuffle(Random)
}@SinceKotlin("1.4")
public fun BooleanArray.shuffle(): Unit {  
shuffle(Random)
}@SinceKotlin("1.4")
public fun CharArray.shuffle(): Unit {  
shuffle(Random)
}@SinceKotlin("1.4")
public fun ByteArray.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1) {
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
    }
}
```
source of randomness. See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm

```kotlin
public fun CharArray.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1) {
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
    }
}
```

```kotlin
/**
 * Sorts elements in the array in-place descending according to natural sort order of the value returned by specified [selector] function.
 */
public fun <T : Comparable<T>> Array<T>.sortDescending(): Unit {
    if (size > 1) sortWith(reverseOrder())
}
```

```kotlin
/**
 * Sorts elements in the array in-place descending according to their natural sort order.
 */
public fun ByteArray.sortDescending(): Unit {
    if (size > 1) { sort().reverse() }
}
```

```kotlin
/**
 * Sorts elements in the array in-place descending according to their natural sort order.
 */
public fun CharArray.sortDescending(): Unit {
    if (size > 1) { sort().reverse() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun <T : Comparable<T>> Array<T>.sortedArray(): Array<T> {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ByteArray.sortedArray(): ByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ShortArray.sortedArray(): ShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun IntArray.sortedArray(): IntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun LongArray.sortedArray(): LongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun FloatArray.sortedArray(): FloatArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun DoubleArray.sortedArray(): DoubleArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun CharArray.sortedArray(): CharArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun <T : Comparable<T>> Array<T>.sorted(): List<T> {
    return sortedArray().asList()
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ByteArray.sorted(): List<Byte> {
    return toTypedArray().apply { sort() }.asList()
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ShortArray.sorted(): List<Short> {
    return toTypedArray().apply { sort() }.asList()
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun IntArray.sorted(): List<Int> {
    return toTypedArray().apply { sort() }.asList()
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun LongArray.sorted(): List<Long> {
    return toTypedArray().apply { sort() }.asList()
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun FloatArray.sorted(): List<Float> {
    return toTypedArray().apply { sort() }.asList()
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun DoubleArray.sorted(): List<Double> {
    return toTypedArray().apply { sort() }.asList()
}
```

```kotlin
/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun CharArray.sorted(): List<Char> {
    return toTypedArray().apply { sort() }.asList()
}
```

```kotlin
/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ByteArray.sortedArray(): ByteArray.sortedArray(): Array<T> {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}
```

```kotlin
/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ShortArray.sortedArray(): ShortArray
```
(isEmpty()) return this
return this.copyOf().apply { sort() }

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun IntArray.sortedArray(): IntArray {
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun LongArray.sortedArray(): LongArray {
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun FloatArray.sortedArray(): FloatArray {
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun DoubleArray.sortedArray(): DoubleArray {
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun CharArray.sortedArray(): CharArray {
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ByteArray.sortedArrayDescending(): ByteArray {
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ShortArray.sortedArrayDescending(): ShortArray {
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun IntArray.sortedArrayDescending(): IntArray {
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun LongArray.sortedArrayDescending(): LongArray {
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun FloatArray.sortedArrayDescending(): FloatArray {
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun DoubleArray.sortedArrayDescending(): DoubleArray {
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun CharArray.sortedArrayDescending(): CharArray {
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ByteArray.sortedBy(selector: (Byte) -> R?): List<Byte> {
    return sortedBy { selector(it) } as List<Byte>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ShortArray.sortedBy(selector: (Short) -> R?): List<Short> {
    return sortedBy { selector(it) } as List<Short>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun IntArray.sortedBy(selector: (Int) -> R?): List<Int> {
    return sortedBy { selector(it) } as List<Int>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun LongArray.sortedBy(selector: (Long) -> R?): List<Long> {
    return sortedBy { selector(it) } as List<Long>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun FloatArray.sortedBy(selector: (Float) -> R?): List<Float> {
    return sortedBy { selector(it) } as List<Float>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun DoubleArray.sortedBy(selector: (Double) -> R?): List<Double> {
    return sortedBy { selector(it) } as List<Double>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun CharArray.sortedBy(selector: (Char) -> R?): List<Char> {
    return sortedBy { selector(it) } as List<Char>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun BooleanArray.sortedBy(selector: (Boolean) -> R?): List<Boolean> {
    return sortedBy { selector(it) } as List<Boolean>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ByteArray.sortedBy(selector: (Byte) -> R?): List<Byte> {
    return sortedBy { selector(it) } as List<Byte>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ShortArray.sortedBy(selector: (Short) -> R?): List<Short> {
    return sortedBy { selector(it) } as List<Short>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun IntArray.sortedBy(selector: (Int) -> R?): List<Int> {
    return sortedBy { selector(it) } as List<Int>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun LongArray.sortedBy(selector: (Long) -> R?): List<Long> {
    return sortedBy { selector(it) } as List<Long>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun FloatArray.sortedBy(selector: (Float) -> R?): List<Float> {
    return sortedBy { selector(it) } as List<Float>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun DoubleArray.sortedBy(selector: (Double) -> R?): List<Double> {
    return sortedBy { selector(it) } as List<Double>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun CharArray.sortedBy(selector: (Char) -> R?): List<Char> {
    return sortedBy { selector(it) } as List<Char>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun BooleanArray.sortedBy(selector: (Boolean) -> R?): List<Boolean> {
    return sortedBy { selector(it) } as List<Boolean>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ByteArray.sortedBy(selector: (Byte) -> R?): List<Byte> {
    return sortedBy { selector(it) } as List<Byte>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ShortArray.sortedBy(selector: (Short) -> R?): List<Short> {
    return sortedBy { selector(it) } as List<Short>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun IntArray.sortedBy(selector: (Int) -> R?): List<Int> {
    return sortedBy { selector(it) } as List<Int>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun LongArray.sortedBy(selector: (Long) -> R?): List<Long> {
    return sortedBy { selector(it) } as List<Long>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun FloatArray.sortedBy(selector: (Float) -> R?): List<Float> {
    return sortedBy { selector(it) } as List<Float>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun DoubleArray.sortedBy(selector: (Double) -> R?): List<Double> {
    return sortedBy { selector(it) } as List<Double>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun CharArray.sortedBy(selector: (Char) -> R?): List<Char> {
    return sortedBy { selector(it) } as List<Char>
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun BooleanArray.sortedBy(selector: (Boolean) -> R?): List<Boolean> {
    return sortedBy { selector(it) } as List<Boolean>
of the value returned by specified [selector] function.

```kotlin
IntArray.sortedBy(crossinline selector: (Int) -> R?): List<Int> { \n  return sortedWith(compareBy(selector))\n}
```

```kotlin
LongArray.sortedBy(crossinline selector: (Long) -> R?): List<Long> { \n  return sortedWith(compareBy(selector))\n}
```

```kotlin
FloatArray.sortedBy(crossinline selector: (Float) -> R?): List<Float> { \n  return sortedWith(compareBy(selector))\n}
```

```kotlin
DoubleArray.sortedBy(crossinline selector: (Double) -> R?): List<Double> { \n  return sortedWith(compareBy(selector))\n}
```

```kotlin
BooleanArray.sortedBy(crossinline selector: (Boolean) -> R?): List<Boolean> { \n  return sortedWith(compareBy(selector))\n}
```

```kotlin
CharArray.sortedBy(crossinline selector: (Char) -> R?): List<Char> { \n  return sortedWith(compareBy(selector))\n}
```

```kotlin
Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
ByteArray.sortedByDescending(crossinline selector: (Byte) -> R?): List<Byte> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
ShortArray.sortedByDescending(crossinline selector: (Short) -> R?): List<Short> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
IntArray.sortedByDescending(crossinline selector: (Int) -> R?): List<Int> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
LongArray.sortedByDescending(crossinline selector: (Long) -> R?): List<Long> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
FloatArray.sortedByDescending(crossinline selector: (Float) -> R?): List<Float> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
DoubleArray.sortedByDescending(crossinline selector: (Double) -> R?): List<Double> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> R?): List<Boolean> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
CharArray.sortedByDescending(crossinline selector: (Char) -> R?): List<Char> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
ByteArray.sortedByDescending(crossinline selector: (Byte) -> R?): List<Byte> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
ShortArray.sortedByDescending(crossinline selector: (Short) -> R?): List<Short> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
IntArray.sortedByDescending(crossinline selector: (Int) -> R?): List<Int> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
LongArray.sortedByDescending(crossinline selector: (Long) -> R?): List<Long> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
FloatArray.sortedByDescending(crossinline selector: (Float) -> R?): List<Float> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
DoubleArray.sortedByDescending(crossinline selector: (Double) -> R?): List<Double> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> R?): List<Boolean> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
CharArray.sortedByDescending(crossinline selector: (Char) -> R?): List<Char> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
ByteArray.sortedByDescending(crossinline selector: (Byte) -> R?): List<Byte> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
ShortArray.sortedByDescending(crossinline selector: (Short) -> R?): List<Short> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
IntArray.sortedByDescending(crossinline selector: (Int) -> R?): List<Int> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
LongArray.sortedByDescending(crossinline selector: (Long) -> R?): List<Long> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
FloatArray.sortedByDescending(crossinline selector: (Float) -> R?): List<Float> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
DoubleArray.sortedByDescending(crossinline selector: (Double) -> R?): List<Double> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> R?): List<Boolean> { \n  return sortedWith(compareByDescending(selector))\n}
```

```kotlin
CharArray.sortedByDescending(crossinline selector: (Char) -> R?): List<Char> { \n  return sortedWith(compareByDescending(selector))\n}
```
return sortedWith(compareByDescending(selector))

/**
 * Returns a list of all elements sorted according to natural sort order.
 */
public fun <T : Comparable<T>> T.sortedWith(comparator: Comparator<in T>): List<T> {
    return sortedArrayWith(comparator).asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun ByteArray.sortedWith(comparator: Comparator<in Byte>): List<Byte> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun ShortArray.sortedWith(comparator: Comparator<in Short>): List<Short> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun IntArray.sortedWith(comparator: Comparator<in Int>): List<Int> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun FloatArray.sortedWith(comparator: Comparator<in Float>): List<Float> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun DoubleArray.sortedWith(comparator: Comparator<in Double>): List<Double> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun BooleanArray.sortedWith(comparator: Comparator<in Boolean>): List<Boolean> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun CharArray.sortedWith(comparator: Comparator<in Char>): List<Char> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun <T> Array<out T>.asList(): List<T>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun ByteArray.asList(): List<Byte>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun ShortArray.asList(): List<Short>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun IntArray.asList(): List<Int>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun LongArray.asList(): List<Long>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun FloatArray.asList(): List<Float>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun DoubleArray.asList(): List<Double>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun BooleanArray.asList(): List<Boolean>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun CharArray.asList(): List<Char>

/**
 * Returns a list of all elements sorted according to the specified 
 * comparator.
 */
public fun <R : Comparable<R>> CharArray.sortedByDescending(crossinline selector: (Char) -> R?): List<Char> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ByteArray.sortedByDescending(crossinline selector: (Byte) -> Int?): List<Byte> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ShortArray.sortedByDescending(crossinline selector: (Short) -> Int?): List<Short> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun IntArray.sortedByDescending(crossinline selector: (Int) -> Int?): List<Int> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun LongArray.sortedByDescending(crossinline selector: (Long) -> Int?): List<Long> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun FloatArray.sortedByDescending(crossinline selector: (Float) -> Int?): List<Float> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun DoubleArray.sortedByDescending(crossinline selector: (Double) -> Int?): List<Double> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> Int?): List<Boolean> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun CharArray.sortedByDescending(crossinline selector: (Char) -> Int?): List<Char> {
    return sortedWith(compareByDescending(selector))
}
DoubleArray.asList(): List<Double> * Returns a [List] that wraps the original array.

BooleanArray.asList(): List<Boolean>

**
* Returns a [List] that wraps the original array.
*

CharArray.asList(): List<Char>

**
* Returns a [List] that wraps the original array.
*

/**
* Returns `true` if the two specified arrays are *deeply* equal to one another,
* i.e. contain the same number of the same elements in the same order.
* If two corresponding elements are nested arrays, they are also compared deeply.
* If any of arrays contains itself on any nesting level the behavior is undefined.
* The elements of other types are compared for equality with the
* [equals] [Any.equals] function.
* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level the behavior is undefined.

ByteArray.contentDeepHashCode(): Int

**
* Returns a hash code based on the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepEquals(other: ByteArray): Boolean

/**
* Returns `true` if the two specified arrays are *structurally* equal to one another,
* i.e. contain the same number of the same elements in the same order.
* The elements are compared for equality with the
* [equals] [Any.equals] function.
* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ShortArray.contentDeepEquals(other: ShortArray): Boolean

/**
* Returns `true` if the two specified arrays are *structurally* equal to one another,
* i.e. contain the same number of the same elements in the same order.
* The elements are compared for equality with the
* [equals] [Any.equals] function.
* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

*An@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")@SinceKotlin("1.1.1")@kotlin.internal.LowPriorityInOverloadResolution@public expect

<T> Array<out T>.contentDeepEquals(other: Array<out T>): Boolean

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepHashCode(): Int

**
* Returns a hash code based on the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.

ByteArray.contentDeepToString(): String

**
* Returns a string representation of the contents of this array as if it is a
* [List],
* Nested arrays are treated as lists too.
* If any of arrays contains itself on any nesting level that reference
* is rendered as `"[...]"` to prevent recursion.
order.

The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.

* The elements are compared for equality with the [equals][Any.equals] function. For floating point numbers it means that 'NaN' is equal to itself and '-0.0' is not equal to '0.0'.
the same number of the same elements in the same order.\n * The elements are compared for equality with the
[equals][Any.equals] function.\n * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not
equal to ‘0.0’.\n * *\n@SinceKotlin("1.4")\npublic expect infix fun LongArray?.contentEquals(other: LongArray?):
Boolean\n\n * Returns ‘true’ if the two specified arrays are *structurally* equal to one another.\n * * i.e. contain the
same number of the same elements in the same order.\n * *\n@SinceKotlin("1.4")\npublic expect infix fun FloatArray?.contentEquals(other: FloatArray?):
Boolean\n\n * Returns ‘true’ if the two specified arrays are *structurally* equal to one another.\n * * i.e. contain the
same number of the same elements in the same order.\n * *\n@SinceKotlin("1.4")\npublic expect infix fun DoubleArray?.contentEquals(other: DoubleArray?):
Boolean\n\n * Returns ‘true’ if the two specified arrays are *structurally* equal to one another.\n * * i.e. contain the
same number of the same elements in the same order.\n * *\n@SinceKotlin("1.4")\npublic expect infix fun BooleanArray?.contentEquals(other: BooleanArray?):
Boolean\n\n * Returns ‘true’ if the two specified arrays are *structurally* equal to one another.\n * * i.e. contain the
same number of the same elements in the same order.\n * *\n@SinceKotlin("1.4")\npublic expect infix fun CharArray?.contentEquals(other: CharArray?):
Boolean\n
 * Returns a hash code based on the
contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun <T> Array<out T>.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun ByteArray.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun ShortArray.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun IntArray.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun LongArray.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun FloatArray.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun DoubleArray.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun BooleanArray.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun CharArray.contentHashCode():
Int\n\n * Returns a hash code based on the contents of this array as if it is [List].\n * *\n@SinceKotlin("1.1")\n@DeprecatedSinceKotlin(hiddenSince = "1.4")\npublic expect fun <T> Array<out T>.contentHashCode():
Int\n
Returns a hash code based on the contents of this array as if it is [List].

@SinceKotlin("1.4")
public expect fun ByteArray?.contentHashCode(): Int

@SinceKotlin("1.4")
public expect fun ShortArray?.contentHashCode(): Int

@SinceKotlin("1.4")
public expect fun IntArray?.contentHashCode(): Int

@SinceKotlin("1.4")
public expect fun LongArray?.contentHashCode(): Int

@SinceKotlin("1.4")
public expect fun FloatArray?.contentHashCode(): Int

@SinceKotlin("1.4")
public expect fun DoubleArray?.contentHashCode(): Int

@SinceKotlin("1.4")
public expect fun BooleanArray?.contentHashCode(): Int

@SinceKotlin("1.4")
public expect fun CharArray?.contentHashCode(): Int

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun <T> Array<out T>.contentToString(): String

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun ByteArray.contentToString(): String

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun ShortArray.contentToString(): String

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun IntArray.contentToString(): String

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun LongArray.contentToString(): String

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun FloatArray.contentToString(): String

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun DoubleArray.contentToString(): String

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun BooleanArray.contentToString(): String

@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public expect fun CharArray.contentToString(): String
It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range. It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

```
*\n* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

*\n* @param destination the array to copy to, 0 by default.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

*\n* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

*\n* @Throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @Throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.

*\n* @return the [destination] array.

*\n* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

```
values.
* \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
*

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public expect fun ShortArray.copyOf(newSize: Int): ShortArray

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public expect fun IntArray.copyOf(newSize: Int): IntArray

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public expect fun LongArray.copyOf(newSize: Int): LongArray

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public expect fun BooleanArray.copyOf(newSize: Int): BooleanArray

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with null char (`\u0000`) values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\u0000`) values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public expect fun CharArray.copyOf(newSize: Int): CharArray

/**
 * Returns new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@Suppress("NO_ACTUAL_FOR_EXPECT")
public expect fun <T> Array<T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<T>
Returns a new array which is a copy of the specified range of the original array. */
public expect fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray

public expect fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray

public expect fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray

public expect fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray

public expect fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray

public expect fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray

public expect fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray

public expect fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray

@SinceKotlin("1.3")
public <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
n@public expect fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray

@SinceKotlin("1.3")
n@public expect fun IntArrayOf(fromIndex: Int, toIndex: Int): IntArray

@SinceKotlin("1.3")
public expect fun ShortArrayOf(fromIndex: Int, toIndex: Int): ShortArray

@SinceKotlin("1.3")
public expect fun LongArrayOf(fromIndex: Int, toIndex: Int): LongArray

@SinceKotlin("1.3")
public expect fun FloatArrayOf(fromIndex: Int, toIndex: Int): FloatArray

@SinceKotlin("1.3")
public expect fun DoubleArrayOf(fromIndex: Int, toIndex: Int): DoubleArray

@SinceKotlin("1.3")
public expect fun BooleanArrayOf(fromIndex: Int, toIndex: Int): BooleanArray

@SinceKotlin("1.3")
public expect fun CharArrayOf(fromIndex: Int, toIndex: Int): CharArray

@SinceKotlin("1.3")
public fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit

@SinceKotlin("1.3")
public fun IntArrayOf(element: Int, fromIndex: Int = 0, toIndex: Int = size): IntArray

@SinceKotlin("1.3")
public fun ShortArrayOf(element: Short, fromIndex: Int = 0, toIndex: Int = size): ShortArray

@SinceKotlin("1.3")
public fun LongArrayOf(element: Long, fromIndex: Int = 0, toIndex: Int = size): LongArray

@SinceKotlin("1.3")
public fun FloatArrayOf(element: Float, fromIndex: Int = 0, toIndex: Int = size): FloatArray

@SinceKotlin("1.3")
public fun DoubleArrayOf(element: Double, fromIndex: Int = 0, toIndex: Int = size): DoubleArray

@SinceKotlin("1.3")
public fun BooleanArrayOf(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): BooleanArray

@SinceKotlin("1.3")
public fun CharArrayOf(element: Char, fromIndex: Int = 0, toIndex: Int = size): CharArray

@SinceKotlin("1.3")
public fun IntArrayOf(element: Int, fromIndex: Int = 0, toIndex: Int = size): IntArray

@SinceKotlin("1.3")
public fun ShortArrayOf(element: Short, fromIndex: Int = 0, toIndex: Int = size): ShortArray

@SinceKotlin("1.3")
public fun LongArrayOf(element: Long, fromIndex: Int = 0, toIndex: Int = size): LongArray

@SinceKotlin("1.3")
public fun FloatArrayOf(element: Float, fromIndex: Int = 0, toIndex: Int = size): FloatArray

@SinceKotlin("1.3")
public fun DoubleArrayOf(element: Double, fromIndex: Int = 0, toIndex: Int = size): DoubleArray

@SinceKotlin("1.3")
public fun BooleanArrayOf(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): BooleanArray

@SinceKotlin("1.3")
public fun CharArrayOf(element: Char, fromIndex: Int = 0, toIndex: Int = size): CharArray

@SinceKotlin("1.3")
public fun IntArrayOf(element: Int, fromIndex: Int = 0, toIndex: Int = size): IntArray

@SinceKotlin("1.3")
public fun ShortArrayOf(element: Short, fromIndex: Int = 0, toIndex: Int = size): ShortArray

@SinceKotlin("1.3")
public fun LongArrayOf(element: Long, fromIndex: Int = 0, toIndex: Int = size): LongArray

@SinceKotlin("1.3")
public fun FloatArrayOf(element: Float, fromIndex: Int = 0, toIndex: Int = size): FloatArray

@SinceKotlin("1.3")
public fun DoubleArrayOf(element: Double, fromIndex: Int = 0, toIndex: Int = size): DoubleArray

@SinceKotlin("1.3")
public fun BooleanArrayOf(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): BooleanArray

@SinceKotlin("1.3")
public fun CharArrayOf(element: Char, fromIndex: Int = 0, toIndex: Int = size): CharArray

@SinceKotlin("1.3")
public fun IntArrayOf(element: Int, fromIndex: Int = 0, toIndex: Int = size): IntArray

@SinceKotlin("1.3")
public fun ShortArrayOf(element: Short, fromIndex: Int = 0, toIndex: Int = size): ShortArray

@SinceKotlin("1.3")
public fun LongArrayOf(element: Long, fromIndex: Int = 0, toIndex: Int = size): LongArray

@SinceKotlin("1.3")
public fun FloatArrayOf(element: Float, fromIndex: Int = 0, toIndex: Int = size): FloatArray

@SinceKotlin("1.3")
public fun DoubleArrayOf(element: Double, fromIndex: Int = 0, toIndex: Int = size): DoubleArray

@SinceKotlin("1.3")
public fun BooleanArrayOf(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): BooleanArray

@SinceKotlin("1.3")
public fun CharArrayOf(element: Char, fromIndex: Int = 0, toIndex: Int = size): CharArray
this array by default.

@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
greater than the size of this array.

@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

public expect fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size):
Unit

@SinceKotlin("1.3")
public expect fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size):
Unit

@SinceKotlin("1.3")
public expect fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size):
Unit

@SinceKotlin("1.3")
public expect fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size):
Unit

@SinceKotlin("1.3")
public expect fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size):
Unit

@SinceKotlin("1.3")
public expect fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size):
Unit

@SinceKotlin("1.3")
public expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size):
Unit

/**
 * Returns the range of valid indices for the array.
 */
public val <T> Array<out T>.indices:
IntRange
    get() = IntRange(0, lastIndex)

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.isEmpty(): Boolean {
    return size == 0
}
ByteArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if the array is not empty.
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.isEmpty(): Boolean {
    return !isEmpty()
}

/**
 * Returns the last valid index for the array.
 */
public val <T> Array<out T>.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ByteArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ShortArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val IntArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val LongArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val FloatArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val DoubleArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val BooleanArray.lastIndex: Int
    get() = size - 1

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NO_ACTUAL_FOR_EXPECT")
public expect operator fun <T> Array<T>.plus(element: T): Array<T>

/**
 * Returns an array containing all elements of the original array and then the given Byte.
 */
public expect operator fun ByteArray.plus(element: Byte): ByteArray

/**
 * Returns an array containing all elements of the original array and then the given Short.
 */
public expect operator fun ShortArray.plus(element: Short): ShortArray

/**
 * Returns an array containing all elements of the original array and then the given Int.
 */
public expect operator fun IntArray.plus(element: Int): IntArray

/**
 * Returns an array containing all elements of the original array and then the given Long.
 */
public expect operator fun LongArray.plus(element: Long): LongArray

/**
 * Returns an array containing all elements of the original array and then the given Float.
 */
public expect operator fun FloatArray.plus(element: Float): FloatArray

/**
 * Returns an array containing all elements of the original array and then the given Double.
 */
public expect operator fun DoubleArray.plus(element: Double): DoubleArray

println("Open Source Used In DNAC Platform Antelope 9149")
original array and then the given [element].

*public expect operator fun CharArray.plus(element: Char):
  CharArray
  * Returns an array containing all elements of the original array and then all elements of the
given [elements] collection.

*public expect operator fun <T> Array<T>.plus(elements: Collection<T>): Array<T>
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun ShortArray.plus(elements: Collection<Short>): ShortArray
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun IntArray.plus(elements: Collection<Int>): IntArray
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun LongArray.plus(elements: Collection<Long>): LongArray
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun FloatArray.plus(elements: Collection<Float>): FloatArray
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun CharArray.plus(elements: Collection<Character>): CharArray
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun <T> Array<T>.plusElement(element: T):
  Array<T>
  * Sorts the array in-place.

*public expect operator fun IntArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun LongArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun ByteArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun ShortArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun DoubleArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun FloatArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun DoubleArray.<T> plus(elements: Collection<T>): Array<T>
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun BooleanArray.<T> plus(elements: Collection<T>): Array<T>
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun CharArray.<T> plus(elements: Collection<T>): Array<T>
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun <T> Array<T>.plusElement(element: T):
  Array<T>
  * Sorts the array in-place.

*public expect operator fun IntArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun LongArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun ByteArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun ShortArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun DoubleArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun FloatArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun DoubleArray.<T> plus(elements: Collection<T>): Array<T>
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun BooleanArray.<T> plus(elements: Collection<T>): Array<T>
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun CharArray.<T> plus(elements: Collection<T>): Array<T>
  * Returns an array containing all elements of the
original array and then all elements of the given [elements] collection.

*public expect operator fun <T> Array<T>.plusElement(element: T):
  Array<T>
  * Sorts the array in-place.

*public expect operator fun IntArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun LongArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun ByteArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun ShortArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun DoubleArray.sort(): Unit
  * Sorts the array in-place.

*public expect operator fun FloatArray.sort(): Unit
  * Sorts the array in-place.
import 
array in-place.

public fun CharArray.sort(): Unit
* @param fromIndex the start of the range (inclusive) to sort, 0 by default.
* @param toIndex the end of the range (exclusive) to sort, size of this array by default.
* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
* @sample samples.collections.Arrays.Sorting.sortArrayOfComparable

ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit

IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit

LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit

FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit

DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit
array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @sample

samples.collections.Arrays.Sorting.sortRangeOfArray

* @SinceKotlin("1.4")

public expect fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @sample

samples.collections.Arrays.Sorting.sortRangeOfArray

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> Array<out T>.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> ByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> ShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> IntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> LongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FloatArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> DoubleArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @sample

samples.collections.Arrays.Sorting.sortRangeOfArray

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FunIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FunByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FunShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FunIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FunLongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FunCharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FunByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun <T : Comparable<T>> FunCharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunLongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunCharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunCharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

* @SinceKotlin("1.4")

public fun FunByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit

* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.

* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
@Throws(IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

*/
@SinceKotlin("1.4")
public fun CharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {
    sort(fromIndex, toIndex)
    reverse(fromIndex, toIndex)
}

/**
 * Sorts the array in-place according to the order specified by the given [comparator].
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>): Unit

/**
 * Returns an array of Boolean containing all of the elements of this generic array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentExpection if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun Array<out Boolean>.toBooleanArray(): BooleanArray {
    return BooleanArray(size) { index -> this[index] }
}

/**
 * Returns an array of Byte containing all of the elements of this generic array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun Array<out Byte>.toByteArray(): ByteArray {
    return ByteArray(size) { index -> this[index] }
}

/**
 * Returns an array of Char containing all of the elements of this generic array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun Array<out Char>.toCharArray(): CharArray {
    return CharArray(size) { index -> this[index] }
}

/**
 * Returns an array of Double containing all of the elements of this generic array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun Array<out Double>.toDoubleArray(): DoubleArray {
    return DoubleArray(size) { index -> this[index] }
}

/**
 * Returns an array of Float containing all of the elements of this generic array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun Array<out Float>.toFloatArray(): FloatArray {
    return FloatArray(size) { index -> this[index] }
}

/**
 * Returns an array of Int containing all of the elements of this generic array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun Array<out Int>.toIntArray(): IntArray {
    return IntArray(size) { index -> this[index] }
}

/**
 * Returns an array of Long containing all of the elements of this generic array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun Array<out Long>.tolongArray(): LongArray {
    return LongArray(size) { index -> this[index] }
}

/**
 * Returns an array of Short containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun Array<out Short>.toShortArray(): ShortArray {
    return ShortArray(size) { index -> this[index] }
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun ByteArray.toTypedArray(): Array<Byte>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun ShortArray.toTypedArray(): Array<Short>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun IntArray.toTypedArray(): Array<Int>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun LongArray.toTypedArray(): Array<Long>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun DoubleArray.toTypedArray(): Array<Double>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun FloatArray.toTypedArray(): Array<Float>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun BooleanArray.toTypedArray(): Array<Boolean>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun CharArray.toTypedArray(): Array<Char>

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun <T, K, V> Array<out T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

Open Source Used In DNAC Platform Antelope 9153
samples.collections.Arrays.Transformations.associateArrayOfPrimitives

public inline fun <K, V>: ByteArray.associate(transform: (Byte) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform, * * * )
}

public inline fun <K, V>: ShortArray.associate(transform: (Short) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform, * * * )
}

public inline fun <K, V>: IntArray.associate(transform: (Int) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform, * * * )
}

public inline fun <K, V>: LongArray.associate(transform: (Long) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform, * * * )
}

public inline fun <K, V>: FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform, * * * )
}

public inline fun <K, V>: DoubleArray.associate(transform: (Double) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform, * * * )
}

public inline fun <K, V>: BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform, * * * )
}

public inline fun <K, V>: CharArray.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform, * * * )
}
order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy

/**
 * Returns a Map containing the elements from the given array indexed by the key
 * returned from [keySelector] function applied to each element. If any two elements would have the same key
 * returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration
 * order of the original array.
 */

public inline fun <T, K

Array<out T>.associateBy(keySelector: (T) -> K): Map<K, T> {
    val capacity = 
    mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, T>(capacity),
    keySelector)
    
    return associateByTo(LinkedHashMap<K, T>(capacity),
    keySelector)
}

/**
 * Returns a Map containing the elements from the given array indexed by the key
 * returned from [keySelector] function applied to each element. If any two elements would have the same key
 * returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration
 * order of the original array.
 */

public inline fun <K

ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> {
    val capacity = 
    mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Byte>(capacity),
    keySelector)
    
    return associateByTo(LinkedHashMap<K, Byte>(capacity),
    keySelector)
}

/**
 * Returns a Map containing the elements from the given array indexed by the key
 * returned from [keySelector] function applied to each element. If any two elements would have the same key
 * returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration
 * order of the original array.
 */

public inline fun <K

ShortArray.associateBy(keySelector: (Short) -> K): Map<K, Short> {
    val capacity = 
    mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Short>(capacity),
    keySelector)
    
    return associateByTo(LinkedHashMap<K, Short>(capacity),
    keySelector)
}

/**
 * Returns a Map containing the elements from the given array indexed by the key
 * returned from [keySelector] function applied to each element. If any two elements would have the same key
 * returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration
 * order of the original array.
 */

public inline fun <K

IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> {
    val capacity = 
    mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Int>(capacity),
    keySelector)
    
    return associateByTo(LinkedHashMap<K, Int>(capacity),
    keySelector)
}

/**
 * Returns a Map containing the elements from the given array indexed by the key
 * returned from [keySelector] function applied to each element. If any two elements would have the same key
 * returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration
 * order of the original array.
 */

public inline fun <K

LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> {
    val capacity = 
    mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Long>(capacity),
    keySelector)
    
    return associateByTo(LinkedHashMap<K, Long>(capacity),
    keySelector)
}

/**
 * Returns a Map containing the elements from the given array indexed by the key
 * returned from [keySelector] function applied to each element. If any two elements would have the same key
 * returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration
 * order of the original array.
 */

public inline fun <K

FloatArray.associateBy(keySelector: (Float) -> K): Map<K, Float> {
    val capacity = 
    mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Float>(capacity),
    keySelector)
    
    return associateByTo(LinkedHashMap<K, Float>(capacity),
    keySelector)
}

/**
 * Returns a Map containing the elements from the given array indexed by the key
 * returned from [keySelector] function applied to each element. If any two elements would have the same key
 * returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration
 * order of the original array.
 */

public inline fun <K

DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> {
    val capacity = 
    mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Double>(capacity),
    keySelector)
    
    return associateByTo(LinkedHashMap<K, Double>(capacity),
    keySelector)
}
sample.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K> BooleanArray.associateBy(keySelector: (Boolean) -> K): Map<K, Boolean> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, Boolean>(capacity), keySelector) } Returns a [Map] containing the elements from the given array indexed by the key returned from [keySelector] function applied to each element. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K> CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K, V> StringArray.associateBy(keySelector: (String) -> K, valueTransform: (String) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy
* /public inline fun <K, V> ObjectArray.associateBy(keySelector: (Object) -> K, valueTransform: (Object) -> V): Map<K, V> { val capacity = mapCapacity(size).coerceAtLeast(16) return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform) } Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.
and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

```kotlin
public inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)"n
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

```
two elements would have the same key returned by [keySelector] the last one gets added to the map.

```kotlin
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo

public inline fun <K, M : MutableMap<in K, in Any>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M {
    for (element in this)
        destination.put(keySelector(element), element)
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,
 * where key is provided by the [keySelector] function applied to each element of the given array,
 * and value is the element itself.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo

```
valueTransform(element)\n    }\n    return destination
}\n
/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */

public inline fun <K, V, M : MutableMap<K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))\n    }\n    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */

public inline fun <K, V, M : MutableMap<K, in V>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))\n    }\n    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */

public inline fun <K, V, M : MutableMap<K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))\n    }\n    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */

public inline fun <K, V, M : MutableMap<K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))\n    }\n    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */

public inline fun <K, V, M : MutableMap<K, in V>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))\n    }\n    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */

public inline fun <K, V, M : MutableMap<K, in V>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))\n    }\n    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n */

public inline fun <K, V, M : MutableMap<K, in V>> CharArray.associateByTo(destination: M, keySelector: 
(Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

public inline fun <T, K, V, M : MutableMap<in K, in V>> Array<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

public inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform: (Byte) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

public inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

public inline fun <K, V, M :MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

public inline fun <K, V, M :MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform: (Double) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * If any of two pairs would have the same key the last one gets added to the map.
 */
@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo

public inline fun <K, V, M :MutableMap<in K, in V>> BooleanArray.associateTo(destination: M, transform: (Boolean) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}
map

*\n* \n* @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
\n*\n
//public inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateTo(destination: M, transform: (Char) ->
Pair<K, V>): M {
*\nfor (element in this) {
*\n\ndestination += transform(element)
*\n}
*\nreturn
*\ndestination
*\n}

//Returns a [Map] where keys are elements from the given array and values are\n*\nproduced by the [valueSelector] function applied to each element.\n*\n*\nIf any two elements are equal, the last one gets added to the map.\n*\n*\nThe returned map preserves the entry iteration order of the original array.\n*\n*\n@sample samples.collections.Collections.Transformations.associateWith

/*\n* @SinceKotlin("1.4")\n*\n@kotlin.internal.InlineOnly\n*\npublic inline fun <V> ByteArray.associateWith(valueSelector: (Byte) -> V): Map<Byte, V> {
*\nval result = LinkedHashMap<Byte, V>(mapCapacity(size).coerceAtLeast(16))\n*\n\nreturn associateWithTo(result, valueSelector)\n*\n}

//Returns a [Map] where keys are elements from the given array and values are\n*\nproduced by the [valueSelector] function applied to each element.\n*\n*\nIf any two elements are equal, the last one gets added to the map.\n*\n*\nThe returned map preserves the entry iteration order of the original array.\n*\n*\n@sample samples.collections.Collections.Transformations.associateWith

/*\n* @SinceKotlin("1.4")\n*\n@kotlin.internal.InlineOnly\n*\npublic inline fun <V> ShortArray.associateWith(valueSelector: (Short) -> V): Map<Short, V> {
*\nval result = LinkedHashMap<Short, V>(mapCapacity(size).coerceAtLeast(16))\n*\n\nreturn associateWithTo(result, valueSelector)\n*\n}

//Returns a [Map] where keys are elements from the given array and values are\n*\nproduced by the [valueSelector] function applied to each element.\n*\n*\nIf any two elements are equal, the last one gets added to the map.\n*\n*\nThe returned map preserves the entry iteration order of the original array.\n*\n*\n@sample samples.collections.Collections.Transformations.associateWith

/*\n* @SinceKotlin("1.4")\n*\n@kotlin.internal.InlineOnly\n*\npublic inline fun <V> IntArray.associateWith(valueSelector: (Int) -> V): Map<Int, V> {
*\nval result = LinkedHashMap<Int, V>(mapCapacity(size).coerceAtLeast(16))\n*\n\nreturn associateWithTo(result, valueSelector)\n*\n}

//Returns a [Map] where keys are elements from the given array and values are\n*\nproduced by the [valueSelector] function applied to each element.\n*\n*\nIf any two elements are equal, the last one gets added to the map.\n*\n*\nThe returned map preserves the entry iteration order of the original array.\n*\n*\n@sample samples.collections.Collections.Transformations.associateWith

/*\n* @SinceKotlin("1.4")\n*\n@kotlin.internal.InlineOnly\n*\npublic inline fun <V> LongArray.associateWith(valueSelector: (Long) -> V): Map<Long, V> {
*\nval result = LinkedHashMap<Long, V>(mapCapacity(size).coerceAtLeast(16))\n*\n\nreturn associateWithTo(result, valueSelector)\n*\n}

//Returns a [Map] where keys are elements from the given array and values are\n*\nproduced by the [valueSelector] function applied to each element.\n*\n*\nIf any two elements are equal, the last one gets added to the map.\n*\n*\nThe returned map preserves the entry iteration order of the original array.\n*\n*\n@sample samples.collections.Collections.Transformations.associateWith

/*\n* @SinceKotlin("1.4")\n*\n@kotlin.internal.InlineOnly\n*\npublic inline fun <V> FloatArray.associateWith(valueSelector: (Float) -> V): Map<Float, V> {
*\nval result = LinkedHashMap<Float, V>(mapCapacity(size).coerceAtLeast(16))\n*\n\nreturn associateWithTo(result, valueSelector)\n*\n}

//Returns a [Map] where keys are elements from the given array and values are\n*\nproduced by the [valueSelector] function applied to each element.\n*\n*\nIf any two elements are equal, the last one gets added to the map.\n*\n*\nThe returned map preserves the entry iteration order of the original array.\n*\n*\n@sample samples.collections.Collections.Transformations.associateWith


Open Source Used In DNAC Platform Antelope 9161
```kotlin
/**
 * Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element. If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original array.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in K, in V>> Array<out K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}
```
Open Source Used In DNAC Platform Antelope 9163

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Long, in V>>
LongArray.associateWithTo(destination: M, valueSelector: (Long) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array. If any two elements are equal, the last one overwrites the former value in the map.
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Float, in V>>
FloatArray.associateWithTo(destination: M, valueSelector: (Float) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array. If any two elements are equal, the last one overwrites the former value in the map.
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Double, in V>>
DoubleArray.associateWithTo(destination: M, valueSelector: (Double) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array. If any two elements are equal, the last one overwrites the former value in the map.
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Boolean, in V>>
CharArray.associateWithTo(destination: M, valueSelector: (Boolean) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <T, C : MutableCollection<in T>> Array<out T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Byte>> ByteArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Short>> ShortArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Int>> IntArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Long>>
LongArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Float>>
FloatArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}
```

---

Open Source Used In DNAC Platform Antelope 9163
BooleanArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

ByteArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

CharArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

CharArray.toHashSet(): HashSet<Char> {
    return toCollection(HashSet<Char>(mapCapacity(size.coerceAtMost(128))))
}

CharArray.toList(): List<Char> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}

CharArray.toMutableList(): MutableList<Char> {
    return ArrayList(this.asCollection())
}

IntArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

IntArray.toHashSet(): HashSet<Int> {
    return toCollection(HashSet<Int>(mapCapacity(size)))
}

IntArray.toList(): List<Int> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}

IntArray.toMutableList(): MutableList<Int> {
    return ArrayList(this.asCollection())
}

LongArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

LongArray.toHashSet(): HashSet<Long> {
    return toCollection(HashSet<Long>(mapCapacity(size)))
}

LongArray.toList(): List<Long> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}

LongArray.toMutableList(): MutableList<Long> {
    return ArrayList(this.asCollection())
}

FloatArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

FloatArray.toHashSet(): HashSet<Float> {
    return toCollection(HashSet<Float>(mapCapacity(size)))
}

FloatArray.toList(): List<Float> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}

FloatArray.toMutableList(): MutableList<Float> {
    return ArrayList(this.asCollection())
}

DoubleArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

DoubleArray.toHashSet(): HashSet<Double> {
    return toCollection(HashSet<Double>(mapCapacity(size)))
}

DoubleArray.toList(): List<Double> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}

DoubleArray.toMutableList(): MutableList<Double> {
    return ArrayList(this.asCollection())
}

CharArray.toFloatArray(): FloatArray {
    val array = FloatArray(size)
    var i = 0
    while (i < size) {
        array[i] = (this[i] as Char).toFloat()
        i++
    }
    return array
}

CharArray.toShortArray(): ShortArray {
    val array = ShortArray(size)
    var i = 0
    while (i < size) {
        array[i] = (this[i] as Char).toShort()
        i++
    }
    return array
}

CharArray.toString(): String {
    return this.joinToString()
array

```kotlin
fun LongArray.toMutableList(): MutableList<Long> { val list = ArrayList<Long>(size) for (item in this) list.add(item) return list } /* Returns a new [MutableList] filled with all elements of this array. */

```
list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `IntArray.flatMap(transform: (Int) -> Iterable<R>): List<R>`
  * `return flatMapTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `LongArray.flatMap(transform: (Long) -> Iterable<R>): List<R>`
  * `return flatMapTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `FloatArray.flatMap(transform: (Float) -> Iterable<R>): List<R>`
  * `return flatMapTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `DoubleArray.flatMap(transform: (Double) -> Iterable<R>): List<R>`
  * `return flatMapTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `BooleanArray.flatMap(transform: (Boolean) -> Iterable<R>): List<R>`
  * `return flatMapTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `CharArray.flatMap(transform: (Char) -> Iterable<R>): List<R>`
  * `return flatMapTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `Array<out T>.flatMap(transform: (T) -> Sequence<R>): List<R>`
  * `return flatMapTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `ByteArray.flatMapIndexed(transform: (index: Int, Byte) -> Iterable<R>): List<R>`
  * `return flatMapIndexedTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

* `ShortArray.flatMapIndexed(transform: (index: Int, Short) -> Iterable<R>): List<R>`
  * `return flatMapIndexedTo(ArrayList<R>(), transform)`: Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <R> IntArray.flatMapIndexed(transform: (index: Int, Int) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from
 * results of [transform] function being invoked on each element * and its index in the original array.
 * @sample samples.collections.Transformations.flatMapIndexed
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <R> LongArray.flatMapIndexed(transform: (index: Int, Long) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from
 * results of [transform] function being invoked on each element * and its index in the original array.
 * @sample samples.collections.Transformations.flatMapIndexed
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <R> FloatArray.flatMapIndexed(transform: (index: Int, Float) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from
 * results of [transform] function being invoked on each element * and its index in the original array.
 * @sample samples.collections.Transformations.flatMapIndexed
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <R> DoubleArray.flatMapIndexed(transform: (index: Int, Double) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from
 * results of [transform] function being invoked on each element * and its index in the original array.
 * @sample samples.collections.Transformations.flatMapIndexed
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <R> BooleanArray.flatMapIndexed(transform: (index: Int, Boolean) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from
 * results of [transform] function being invoked on each element * and its index in the original array.
 * @sample samples.collections.Transformations.flatMapIndexed
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element * and its index in the original array, to the given
 * destination.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}
```
public inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapIndexedTo(destination: C, transform: (index: Int, Byte) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element * and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, Short) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element * and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> IntArray.flatMapIndexedTo(destination: C, transform: (index: Int, Int) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element * and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> LongArray.flatMapIndexedTo(destination: C, transform: (index: Int, Long) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element * and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapIndexedTo(destination: C, transform: (index: Int, Float) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element * and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapIndexedTo(destination: C, transform: (index: Int, Double) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element * and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapIndexedTo(destination: C, transform: (index: Int, Boolean) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element * and its index in the original array, to the given [destination].
 */
results of [transform] function being invoked on each element\n and its index in the original array, to the given [destination].\n
```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> CharArray.flatMapIndexedTo(destination: C, transform: (index: Int, Char) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n and its index in the original array, to the given [destination].\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedSequenceTo")
@kotlin.internal.InlineOnly
public inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n of original array, to the given [destination].\n */
public inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n of original array, to the given [destination].\n */
public inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapTo(destination: C, transform: (Byte) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n of original array, to the given [destination].\n */
public inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapTo(destination: C, transform: (Short) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n of original array, to the given [destination].\n */
public inline fun <R, C : MutableCollection<in R>> IntArray.flatMapTo(destination: C, transform: (Int) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n of original array, to the given [destination].\n */
public inline fun <R, C : MutableCollection<in R>> LongArray.flatMapTo(destination: C, transform: (Long) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n of original array, to the given [destination].\n */
public inline fun <R, C :MutableCollection<in R>> FloatArray.flatMapTo(destination: C, transform: (Float) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n of original array, to the given [destination].\n */
public inline fun <R, C :MutableCollection<in R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> CharArray.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}
```
element of original array, to the given [destination].

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnValue
@kotlin.jvm.JvmName("flatMapSequenceTo")
public inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector]
 * function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <T, K> Array<out T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <T, K> ByteArray.groupBy(keySelector: (Byte) -> K): Map<K, List<Byte>> {
    return groupByTo(LinkedHashMap<K, MutableList<Byte>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <T, K> ShortArray.groupBy(keySelector: (Short) -> K): Map<K, List<Short>> {
    return groupByTo(LinkedHashMap<K, MutableList<Short>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <T, K> IntArray.groupBy(keySelector: (Int) -> K): Map<K, List<Int>> {
    return groupByTo(LinkedHashMap<K, MutableList<Int>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <T, K> LongArray.groupBy(keySelector: (Long) -> K): Map<K, List<Long>> {
    return groupByTo(LinkedHashMap<K, MutableList<Long>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <T, K> FloatArray.groupBy(keySelector: (Float) -> K): Map<K, List<Float>> {
    return groupByTo(LinkedHashMap<K, MutableList<Float>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <T, K> DoubleArray.groupBy(keySelector: (Double) -> K): Map<K, List<Double>> {
    return groupByTo(LinkedHashMap<K, MutableList<Double>>(), keySelector)
}
```

---

Open Source Used In DNAC Platform Antelope 9170
BooleanArray.groupby(keySelector: (Boolean) -> K): Map<K, List<Boolean>>

```java
:return groupByTo(LinkedHashMap<K, MutableList<Boolean>>(), keySelector)
```

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the original array.

---

CharArray.groupby(keySelector: (Char) -> K): Map<K, List<Char>>

```java
:return groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)
```

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and returns a map where each group key is associated with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array.

---

ByteArray.groupby(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, List<V>>

```java
:return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
```

Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and returns a map where each group key is associated with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array.

---

IntArray.groupby(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>>

```java
:return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
```

Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and returns a map where each group key is associated with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array.

---

LongArray.groupby(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, List<V>>

```java
:return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
```

Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and returns a map where each group key is associated with a list of corresponding values. The returned map preserves the entry iteration order of the keys produced from the original array.
groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)

Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element

and returns a map where each group key is associated with a list of corresponding values.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> DoubleArray.groupBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, List<V>>

Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element

and returns a map where each group key is associated with a list of corresponding values.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> BooleanArray.groupBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, List<V>>

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> CharArray.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>>

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ByteArray.groupBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, List<V>>

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ShortArray.groupBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, List<V>>

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> IntArray.groupBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>>

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ShortArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> IntArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ShortArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> IntArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ShortArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> IntArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ShortArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> IntArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ShortArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> IntArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ShortArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> IntArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> ShortArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.

samples.collections.Collections.Transformations.groupByKeysAndValues

public inline fun <K, V> IntArray.groupByTo(destination: M): M

Groups elements of the original array by the key returned by the given [keySelector] function applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.

The returned map preserves the entry iteration order of the keys produced from the original array.
public inline fun <K, M : MutableMap<in K, MutableList<Long>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K): M {
    return The [destination] map.
}

public inline fun <K, M : MutableMap<in K, MutableList<Float>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K): M {
    return The [destination] map.
}

public inline fun <K, M : MutableMap<in K, MutableList<Double>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K): M {
    return The [destination] map.
}

public inline fun <K, M : MutableMap<in K, MutableList<Boolean>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K): M {
    return The [destination] map.
}

public inline fun <K, M : MutableMap<in K, MutableList<Char>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K): M {
    return The [destination] map.
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * and puts to the [destination] map each group key associated with a list of corresponding elements.
 */
public inline fun <T, K, V, M : MutableMap<in K, MutableList<V>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    return The [destination] map.
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function and puts to the [destination] map each group key associated with a list of corresponding values.
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {
    return The [destination] map.
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function and puts to the [destination] map each group key associated with a list of corresponding values.
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ShortArray.groupByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    return The [destination] map.
}
(Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array* by the key returned by the given [keySelector] function applied to the element*
 * and puts to the [destination] map each group key associated with a list of corresponding values.\n * @return The [destination] map.*
 */
@Sample
fun <K, V, M : MutableMap<K, MutableList<V>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array* by the key returned by the given [keySelector] function applied to the element*
 * and puts to the [destination] map each group key associated with a list of corresponding values.\n * @return The [destination] map.*
 */
@Sample
fun <K, V, M : MutableMap<K, MutableList<V>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array* by the key returned by the given [keySelector] function applied to the element*
 * and puts to the [destination] map each group key associated with a list of corresponding values.\n * @return The [destination] map.*
 */
@Sample
fun <K, V, M : MutableMap<K, MutableList<V>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array* by the key returned by the given [keySelector] function applied to the element*
 * and puts to the [destination] map each group key associated with a list of corresponding values.\n * @return The [destination] map.*
 */
@Sample
fun <K, V, M : MutableMap<K, MutableList<V>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array* by the key returned by the given [keySelector] function applied to the element*
 * and puts to the [destination] map each group key associated with a list of corresponding values.\n * @return The [destination] map.*
 */
@Sample
fun <K, V, M : MutableMap<K, MutableList<V>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the
 * original array* by the key returned by the given [keySelector] function applied to the element*
 * and puts to the [destination] map each group key associated with a list of corresponding values.\n * @return The [destination] map.*
 */
@Sample
fun <K, V, M : MutableMap<K, MutableList<V>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Creates a [Grouping] source from an array to be used later with one of group-and-fold
 * operations* using the specified [keySelector] function to extract a key from each element.*
 * @since 1.0.*
 */
@SinceKotlin("1.1")
fun <T, K>
Array<out T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> { 
    override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
    override fun keyOf(element: T): K = keySelector(element)
    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @sample samples.collections.Collections.Transformations.map\n */
public inline fun <T, R> Array<out T>.map(transform: (T) -> R): List<R> { 
    return mapTo(ArrayList<R>(size), transform)\n    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @sample samples.collections.Collections.Transformations.map\n */
public inline fun <T, R> Array<out T>.mapIndexed(transform: (index: Int, T) -> R): List<R> { 
    return mapIndexedTo(ArrayList<R>(size), transform)\n    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @sample samples.collections.Collections.Transformations.map\n */
public inline fun <T, R> Array<out T>.mapIndexedTo(transform: (index: Int, T) -> R): List<R> { 
    return mapIndexedTo(ArrayList<R>(size), transform)\n    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}

/**
 * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * @sample samples.collections.Collections.Transformations.map\n */
public inline fun <T, R> Array<out T>.mapIndexedTo(transform: (index: Int, T) -> R): List<R> { 
    return mapIndexedTo(ArrayList<R>(size), transform)\n    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}
function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
public inline fun <R> LongArray.mapIndexed(transform: (index: Int, Long) -> R): List<R> {
    return mapIndexedTo(ArrayList<R>(size), transform)\n}

/**
 * Returns a list containing the results of applying the given \[transform\] function\n * to each element and its index in the original array.\n * @param \[transform\] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
public inline fun <R> FloatArray.mapIndexed(transform: (index: Int, Float) -> R): List<R> {
    return mapIndexedTo(ArrayList<R>(size), transform)\n}

/**
 * Returns a list containing the results of applying the given \[transform\] function\n * to each element and its index in the original array.\n * @param \[transform\] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
public inline fun <R> DoubleArray.mapIndexed(transform: (index: Int, Double) -> R): List<R> {
    return mapIndexedTo(ArrayList<R>(size), transform)\n}

/**
 * Returns a list containing the results of applying the given \[transform\] function\n * to each element and its index in the original array.\n * @param \[transform\] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
public inline fun <R> BooleanArray.mapIndexed(transform: (index: Int, Boolean) -> R): List<R> {
    return mapIndexedTo(ArrayList<R>(size), transform)\n}

/**
 * Returns a list containing only the non-null results of applying the given \[transform\] function\n * to each element and its index in the original array.\n * @param \[transform\] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
public inline fun <R, C : MutableCollection<in R>> Array<out T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {
    forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }
    return destination
}

/**
 * Applies the given \[transform\] function to each element and its index in the original array\n * and appends only the non-null results to the given \[destination\].\n * @param \[transform\] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
public inline fun <R, C : MutableCollection<in R>> ByteArray.mapIndexedNotNullTo(destination: C, transform: (index: Int, Byte) -> R): C {
    forEachIndexed { index, item -> transform(index, item)?.let { destination.add(it) } }
    return destination
}

/**
 * Applies the given \[transform\] function to each element and its index in the original array\n * and appends only the non-null results to the given \[destination\].\n * @param \[transform\] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
public inline fun <R, C : MutableCollection<in R>> ShortArray.mapIndexedNotNullTo(destination: C, transform: (index: Int, Short) -> R): C {
    forEachIndexed { index, item -> transform(index, item)?.let { destination.add(it) } }
    return destination
}

/**
 * Returns a list containing the results of applying the given \[transform\] function\n * to each element and its index in the original array.\n * @param \[transform\] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */
public inline fun <R, C : MutableCollection<in R>>
IntArray.mapIndexedTo(destination: C, transform: (index: Int, item) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    LongArray.mapIndexedTo(destination: C, transform: (index: Int, Long) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    FloatArray.mapIndexedTo(destination: C, transform: (index: Int, Float) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    DoubleArray.mapIndexedTo(destination: C, transform: (index: Int, Double) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    Array<out T>.mapIndexedTo(destination: C, transform: (index: Int, item: T) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    Array<out T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {
    forEach { element -> transform(element)?.let { destination.add(it) } }
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    Array<out T>.mapTo(destination: C, transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    ByteArray.mapTo(destination: C, transform: (Byte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    ShortArray.mapTo(destination: C, transform: (Short) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    IntArray.mapTo(destination: C, transform: (Int) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element and returns the result
 * to the given [destination].
 * 
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * 
 * @return the given [destination] with the transformed elements.
 */
public inline fun <R, C : MutableCollection<R>>
    ShortArray.mapTo(destination: C, transform: (Short) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}
MutableCollection<R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C

    for (item in this)
        destination.add(transform(item))
    return destination

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C

    for (item in this)
        destination.add(transform(item))
    return destination

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> DoubleArray.mapTo(destination: C, transform: (Double) -> R): C

    for (item in this)
        destination.add(transform(item))
    return destination

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean) -> R): C

    for (item in this)
        destination.add(transform(item))
    return destination

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> CharArray.mapTo(destination: C, transform: (Char) -> R): C

    for (item in this)
        destination.add(transform(item))
    return destination

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun <T> Array<out T>.withIndex(): Iterable<IndexedValue<T>>

    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun ByteArray.withIndex(): Iterable<IndexedValue<Byte>>

    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun ShortArray.withIndex(): Iterable<IndexedValue<Short>>

    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun IntArray.withIndex(): Iterable<IndexedValue<Int>>

    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun LongArray.withIndex(): Iterable<IndexedValue<Long>>

    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun FloatArray.withIndex(): Iterable<IndexedValue<Float>>

    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun DoubleArray.withIndex(): Iterable<IndexedValue<Double>>

    return IndexingIterable { iterator() }

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
public fun BooleanArray.withIndex(): Iterable<IndexedValue<Boolean>>

    return IndexingIterable { iterator() }

/**
 * Returns a list containing only distinct elements from the given array.
 * Among equal elements of the given array, only the first one will be present in the resulting list.
 */
public fun <T> Array<out T>.distinct(): List<T>

    return this.toMutableSet().toList()

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun ByteArray.distinct(): List<Byte>

    return this.toMutableSet().toList()

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun ShortArray.distinct(): List<Short>

    return this.toMutableSet().toList()

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun IntArray.distinct(): List<Int>

    return this.toMutableSet().toList()

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun LongArray.distinct(): List<Long>

    return this.toMutableSet().toList()

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun FloatArray.distinct(): List<Float>

    return this.toMutableSet().toList()

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun DoubleArray.distinct(): List<Double>

    return this.toMutableSet().toList()

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun BooleanArray.distinct(): List<Boolean>

    return this.toMutableSet().toList()
ShortArray.distinct(): List<Short> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun ShortArray.distinct(): List<Short> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun IntArray.distinct(): List<Int> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun LongArray.distinct(): List<Long> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun FloatArray.distinct(): List<Float> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun DoubleArray.distinct(): List<Double> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun BooleanArray.distinct(): List<Boolean> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only distinct elements from the given array.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public fun CharArray.distinct(): List<Char> {
    return this.toMutableSet().toList()
}

.ShortArray.distinct(): List<Short> {
    return this.toMutableSet().toList()
}
fun <K> LongArray.distinctBy(selector: (Long) -> K): List<Long> {
    val set = HashSet<K>()
    val list = ArrayList<Long>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

/**
 * Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they
 * were in the source array.
 */
public inline fun <K> FloatArray.distinctBy(selector: (Float) -> K): List<Float> {
    val set = HashSet<K>()
    val list = ArrayList<Float>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

/**
 * Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they
 * were in the source array.
 */
public inline fun <K> DoubleArray.distinctBy(selector: (Double) -> K): List[Double] {
    val set = HashSet<K>()
    val list = ArrayList<Double>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

/**
 * Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they
 * were in the source array.
 */
public inline fun <K> BooleanArray.distinctBy(selector: (Boolean) -> K): List<Boolean> {
    val set = HashSet<K>()
    val list = ArrayList<Boolean>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

/**
 * Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they
 * were in the source array.
 */
public inline fun <K> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = HashSet<K>()
    val list = ArrayList<Char>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}
order of the original array.

* Returns a set containing all elements that are contained at least in one of these collections use [union].

\*\public infix fun FloatArray.intersect(other: Iterable<Float>): Set<Float> {
\{\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n\}
\*\public infix fun DoubleArray.intersect(other: Iterable<Double>): Set<Double> {
\{\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n\}
\*\public infix fun BooleanArray.intersect(other: Iterable<Boolean>): Set<Boolean> {
\{\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n\}

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* Returns a set containing all elements that are contained by both this array and specified collection.

* \* \* Returns a set containing all elements that are contained by both this array and the specified collection.

* \* \* \* Returns a set containing all elements that are contained by both this array and the specified collection.

* \* To get a set containing all elements that are contained at least in one of these collections use [union].

* To get a set containing all elements that are contained at least in one of these collections use [union].

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* \* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* \* \* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* To get a set containing all elements that are contained at least in one of these collections use [union].

* To get a set containing all elements that are contained at least in one of these collections use [union].

* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* \* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* \* \* Returns a set containing all elements that are contained by this array and not contained by the specified collection.

* \* To get a set containing all elements that are contained at least in one of these collections use [union].

* To get a set containing all elements that are contained at least in one of these collections use [union].
preserves the element iteration order of the original array.

```kotlin
public fun ByteArray.toMutableSet(): MutableSet<Byte> {
    return toCollection(LinkedHashSet<Byte>(mapCapacity(size)))
}

/**
 * Returns a new [MutableSet] containing all distinct elements from the given array.
 * The returned set preserves the element iteration order of the original array.
 */
public fun ShortArray.toMutableSet(): MutableSet<Short> {
    return toCollection(LinkedHashSet<Short>(mapCapacity(size)))
}

/**
 * Returns a new [MutableSet] containing all distinct elements from the given array.
 * The returned set preserves the element iteration order of the original array.
 */
public fun IntArray.toMutableSet(): MutableSet<Int> {
    return toCollection(LinkedHashSet<Int>(mapCapacity(size)))
}

/**
 * Returns a new [MutableSet] containing all distinct elements from the given array.
 * The returned set preserves the element iteration order of the original array.
 */
public fun LongArray.toMutableSet(): MutableSet<Long> {
    return toCollection(LinkedHashSet<Long>(mapCapacity(size)))
}

/**
 * Returns a new [MutableSet] containing all distinct elements from the given array.
 * The returned set preserves the element iteration order of the original array.
 */
public fun FloatArray.toMutableSet(): MutableSet<Float> {
    return toCollection(LinkedHashSet<Float>(mapCapacity(size)))
}

/**
 * Returns a new [MutableSet] containing all distinct elements from the given array.
 * The returned set preserves the element iteration order of the original array.
 */
public fun DoubleArray.toMutableSet(): MutableSet<Double> {
    return toCollection(LinkedHashSet<Double>(mapCapacity(size)))
}

/**
 * Returns a set containing all distinct elements from both collections.
 * The returned set preserves the element iteration order of the original array.
 * Those elements of the [other] collection that are unique are iterated in the end
 * in the order of the [other] collection.
 * To get a set containing all elements that are contained in both collections use
 * [intersect].
 */
public infix fun <T> Array<out T>.union(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun ByteArray.union(other: Iterable<Byte>): Set<Byte> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

public fun ShortArray.union(other: Iterable<Short>): Set<Short> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun IntArray.union(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

public fun LongArray.union(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun FloatArray.union(other: Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

public fun DoubleArray.union(other: Iterable<Double>): Set<Double> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

public fun CharArray.union(other: Iterable<Char>): Set<Char> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun ByteArray.union(other: Iterable<Byte>): Set<Byte> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun ShortArray.union(other: Iterable<Short>): Set<Short> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun IntArray.union(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun LongArray.union(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun FloatArray.union(other: Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun DoubleArray.union(other: Iterable<Double>): Set<Double> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```

```kotlin
public fun CharArray.union(other: Iterable<Char>): Set<Char> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
```
unique are iterated in the end * in the order of the [other] collection. * To get a set containing all elements that are contained in both collections use [intersect]. */
public infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.
 * Those elements of the [other] collection that are unique are iterated in the end * in the order of the [other] collection. * To get a set containing all elements that are contained in both collections use [intersect]. */
public infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.
 * Those elements of the [other] collection that are unique are iterated in the end * in the order of the [other] collection. * To get a set containing all elements that are contained in both collections use [intersect]. */
public infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.
 * Those elements of the [other] collection that are unique are iterated in the end * in the order of the [other] collection. * To get a set containing all elements that are contained in both collections use [intersect]. */
public infix fun CharArray.union(other: Iterable<Character>): Set<Character> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}
samples.collections.Collections.Aggregates.any

//public fun IntArray.any(): Boolean {
\nreturn 
!isEmpty() \n* Returns `true` if array has at least one element.\n* \n* @sample 
samples.collections.Collections.Aggregates.any

//public fun LongArray.any(): Boolean {
\nreturn 
!isEmpty() \n* Returns `true` if array has at least one element.\n* \n* @sample 
samples.collections.Collections.Aggregates.any

//public fun FloatArray.any(): Boolean {
\nreturn 
!isEmpty() \n* Returns `true` if array has at least one element.\n* \n* @sample 
samples.collections.Collections.Aggregates.any

//public fun DoubleArray.any(): Boolean {
\nreturn 
!isEmpty() \n* Returns `true` if array has at least one element.\n* \n* @sample 
samples.collections.Collections.Aggregates.any

//public fun BooleanArray.any(): Boolean {
\nreturn 
!isEmpty() \n* Returns `true` if array has at least one element.\n* \n* @sample 
samples.collections.Collections.Aggregates.any

//public fun CharArray.any(): Boolean {
\nreturn 
!isEmpty() \n* Returns `true` if array has at least one element.\n* \n* @sample 
samples.collections.Collections.Aggregates.any

/** 
 * Returns `true` if array has at least one element.
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.any
 */
public fun IntArray.any(): Boolean {
    return 
!isEmpty()
}

/** 
 * Returns `true` if array has at least one element.
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.any
 */
public fun LongArray.any(): Boolean {
    return 
!isEmpty()
}

/** 
 * Returns `true` if array has at least one element.
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.any
 */
public fun FloatArray.any(): Boolean {
    return 
!isEmpty()
}

/** 
 * Returns `true` if array has at least one element.
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.any
 */
public fun DoubleArray.any(): Boolean {
    return 
!isEmpty()
}

/** 
 * Returns `true` if array has at least one element.
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.any
 */
public fun BooleanArray.any(): Boolean {
    return 
!isEmpty()
}

/** 
 * Returns `true` if array has at least one element.
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.any
 */
public fun CharArray.any(): Boolean {
    return 
!isEmpty()
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun <T> Array<out T>.any(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun ByteArray.any(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun ShortArray.any(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun IntArray.any(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun LongArray.any(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun FloatArray.any(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun DoubleArray.any(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun BooleanArray.any(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample 
 * samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun CharArray.any(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return 
false
}

/** 
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline 
fun <T> Array<out T>.count(): Int {
    return 
size
}

/** 
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.count(): Int {
    return 
size
}

/** 
 * Returns the 
number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.count(): Int {
    return 
size
}

/** 
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.count(): Int {
    return 
size
}

/** 
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.count(): Int {
    return 
size
}

/** 
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.count(): Int {
    return 
size
}

/** 
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.count(): Int {
    return 
size
}

/** 
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.count(): Int {
    return 
size
}

/** 
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.count(): Int {
    return 
size
}
public inline fun BooleanArray.count(): Int { var count = 0

    for (element in this) if (predicate(element)) ++count
    return count
}

public inline fun CharArray.count(): Int {
    return size
}

public inline fun <T> Array<out T>.count(predicate: (T) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

public inline fun ByteArray.count(predicate: (Byte) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

public inline fun ShortArray.count(predicate: (Short) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

public inline fun IntArray.count(predicate: (Int) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

public inline fun LongArray.count(predicate: (Long) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

public inline fun BooleanArray.count(predicate: (Boolean) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

public inline fun CharArray.count(predicate: (Char) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

public inline fun ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

public inline fun ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

public inline fun IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

public inline fun LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

public inline fun FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

public inline fun DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

public inline fun <R> Array<out T>.fold(initial: R, operation: (acc: R, T) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* /\n
public inline fun <R> FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {
    var accumulator = initial
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* /\n
public inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {
    var accumulator = initial
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {
    var accumulator = initial
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {
    var accumulator = initial
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {
    var accumulator = initial
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

* Returns the specified [initial] value if the array is empty.

* /\n
public inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

* Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
calculates the next accumulator value.

```java
@public inline fun <R> LongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): R {
    var index = 0
    var accumulator = initial
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right in * to current accumulator value and each element with its index in the original array.

```java
@public inline fun <R> LongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Accumulator) -> R): R {
    var index = 0
    var accumulator = initial
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right in * to current accumulator value and each element with its index in the original array.

```java
@public inline fun <R> CharArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this)
        accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right in * to current accumulator value and each element with its index in the original array.

```java
@public inline fun <R> BooleanArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this)
        accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right in * to current accumulator value and each element with its index in the original array.

```java
@public inline fun <R> DoubleArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this)
        accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right in * to current accumulator value and each element with its index in the original array.

```java
@public inline fun <R> FloatArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this)
        accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right in * to current accumulator value and each element with its index in the original array.

```java
@public inline fun <T> foldRight(initial: R, operation: (T, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    for (element in this)
        accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

Accumulates value starting with [initial] value and applying [operation] from left to right in * to current accumulator value and each element with its index in the original array.

```java
@public inline fun <R> IntArray.foldRight(initial: R, operation: (Int, acc: R) -> R): R
```
```kotlin
// Example Kotlin function to demonstrate foldRightIndexed

fun <T, R> Array<out T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @param [initial] value if the array is empty.
 * @return
 */
public inline fun <R> LongArray.foldRight(initial: R, operation: (Long, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

fun <R> CharArray.foldRight(initial: R, operation: (Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index--), accumulator)
    }
    return accumulator
}

fun <R> BooleanArray.foldRight(initial: R, operation: (Boolean, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

fun <R> DoubleArray.foldRight(initial: R, operation: (Double, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

fun <R> FloatArray.foldRight(initial: R, operation: (Float, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Open Source Used In DNAC Platform Antelope 9188
for (element in this) action(element) * Performs the given [action] on each element.

/*
public inline fun DoubleArray.forEach(action: (Double) -> Unit): Unit {
    for (element in this) action(element) * Performs the given [action] on each element.
}*/

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes the index of an element and the element itself
 * and performs the action on the element.
 */
public inline fun DoubleArray.forEachIndexed(action: (Int, Double) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes the index of an element and the element itself
 * and performs the action on the element.
 */
public inline fun BooleanArray.forEachIndexed(action: (Int, Boolean) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes the index of an element and the element itself
 * and performs the action on the element.
 */
public inline fun CharArray.forEachIndexed(action: (Int, Char) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun <T : Comparable<T>> Array<out T>.max(): T? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun ByteArray.max(): Byte? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun ShortArray.max(): Short? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun IntArray.max(): Int? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun LongArray.max(): Long? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun FloatArray.max(): Float? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun DoubleArray.max(): Double? {
    return maxOrNull()
}
null" if there are no elements.

```kotlin
@Sample
public inline fun <R : Comparable<R>> ShortArray.maxByOrNull(selector: (Short) -> R): Short? {
    if (isEmpty()) return null
    var maxElem = this[0] val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem) for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> IntArray.maxByOrNull(selector: (Int) -> R): Int? {
    if (isEmpty()) return null
    var maxElem = this[0] val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem) for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> LongArray.maxByOrNull(selector: (Long) -> R): Long? {
    if (isEmpty()) return null
    var maxElem = this[0] val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem) for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> FloatArray.maxByOrNull(selector: (Float) -> R): Float? {
    if (isEmpty()) return null
    var maxElem = this[0] val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem) for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> DoubleArray.maxByOrNull(selector: (Double) -> R): Double? {
    if (isEmpty()) return null
    var maxElem = this[0] val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem) for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> BooleanArray.maxByOrNull(selector: (Boolean) -> R): Boolean? {
    if (isEmpty()) return null
    var maxElem = this[0] val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem) for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}
```

```kotlin
/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array.
 * 
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.maxOf(selector: (T) -> Double): Double? {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0]) val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxE
Double): Double

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun FloatArray.maxOf(selector: (Float) -> Float): Float

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun IntArray.maxOf(selector: (Int) -> Int): Int

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun LongArray.maxOf(selector: (Long) -> Long): Long

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ByteArray.maxOf(selector: (Byte) -> Byte): Byte

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun CharArray.maxOf(selector: (Char) -> Char): Char

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ShortArray.maxOf(selector: (Short) -> Short): Short

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun IntArray.maxOf(selector: (Int) -> Int): Int

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ByteArray.maxOf(selector: (Byte) -> Byte): Byte

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun CharArray.maxOf(selector: (Char) -> Char): Char

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ShortArray.maxOf(selector: (Short) -> Short): Short

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun IntArray.maxOf(selector: (Int) -> Int): Int

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ByteArray.maxOf(selector: (Byte) -> Byte): Byte

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun CharArray.maxOf(selector: (Char) -> Char): Char

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ShortArray.maxOf(selector: (Short) -> Short): Short

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun IntArray.maxOf(selector: (Int) -> Int): Int

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ByteArray.maxOf(selector: (Byte) -> Byte): Byte

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun CharArray.maxOf(selector: (Char) -> Char): Char

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ShortArray.maxOf(selector: (Short) -> Short): Short

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun IntArray.maxOf(selector: (Int) -> Int): Int

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
    ByLambdaReturnType@n@kotlin.internal.InlineOnly\n    public inline fun ByteArray.maxOf(selector: (Byte) -> Byte): Byte

    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
        return maxValue
    }

    *@SinceKotlin("1.4")*@ OptIn(kotlin.experimental.ExperimentalTypeInference::class)*@OverloadResolution*
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector: (Float) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOf(selector: (Double) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOf(selector: (Boolean) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOf(selector: (Char) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxOf(selector: (T) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ByteArray.maxOf(selector: (Byte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ShortArray.maxOf(selector: (Short) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}
By LambdaReturnType

valArray.maxOf(selector: (Int) -> R): R

LongArray.maxOf(selector: (Long) -> R): R

FloatArray.maxOf(selector: (Float) -> R): R

DoubleArray.maxOf(selector: (Double) -> R): R

CharArray.maxOf(selector: (Char) -> R): R

Array<out T>.maxOfOrNull(selector: (T) -> Double): Double?

If any of values produced by [selector] function are `NaN`, the returned result is `NaN`.

*If any of values produced by [selector] function is `NaN`*, the returned result is `NaN`.

Open Source Used In DNAC Platform Antelope 9196
Open Source Used In DNAC Platform Antelope 9197

```kotlin
public inline fun ByteArray.maxOfOrNull(selector: (Byte) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

public inline fun ShortArray.maxOfOrNull(selector: (Short) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

public inline fun IntArray.maxOfOrNull(selector: (Int) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

public inline fun LongArray.maxOfOrNull(selector: (Long) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

public inline fun FloatArray.maxOfOrNull(selector: (Float) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

public inline fun DoubleArray.maxOfOrNull(selector: (Double) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

public inline fun BooleanArray.maxOfOrNull(selector: (Boolean) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```
is `NaN`, the returned result is `NaN`.

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun CharArray.maxOfOrNull(selector: (Char) -> Double?): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

* `NaN` * `NaN` * `NaN` * `NaN` *

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.maxOfOrNull(selector: (T) -> Float?): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

* `NaN` * `NaN` * `NaN` * `NaN` *

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun ByteArray.maxOfOrNull(selector: (Byte) -> Float?): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

* `NaN` * `NaN` * `NaN` * `NaN` *

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun ShortArray.maxOfOrNull(selector: (Short) -> Float?): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

* `NaN` * `NaN` * `NaN` * `NaN` *

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun IntArray.maxOfOrNull(selector: (Int) -> Float?): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

* `NaN` * `NaN` * `NaN` * `NaN` *

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun LongArray.maxOfOrNull(selector: (Long) -> Float?): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

* `NaN` * `NaN` * `NaN` * `NaN` *

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun FloatArray.maxOfOrNull(selector: (Float) -> Float?): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

* `NaN` * `NaN` * `NaN` * `NaN` *

Open Source Used In DNAC Platform Antelope 9198
if there are no elements.\n
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun DoubleArray.maxOfOrNull(selector: (Double) -> Float): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}

*\n
/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n *
 * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.\n
*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun BooleanArray.maxOfOrNull(selector: (Boolean) -> Float): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n *
 * If any of values produced by [selector] function\n * is `NaN`, the returned result is `NaN`.\n
*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun CharArray.maxOfOrNull(selector: (Char) -> Float): Float? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Array<out T>.maxOfOrNull(selector: (T) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ByteArray.maxOfOrNull(selector: (Byte) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ShortArray.maxOfOrNull(selector: (Short) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> IntArray.maxOfOrNull(selector: (Int) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}

*\n
Open Source Used In DNAC Platform Antelope 9199
LongArray.maxOfOrNull(selector: (Long) -> R): R? {  
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (maxValue < v) {  
            maxValue = v
        }  
    }  
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function * applied to each element in the array or ‘null’ if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> FloatArray.maxOfOrNull(selector: (Float) -> R): R? {  
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (maxValue < v) {  
            maxValue = v
        }  
    }  
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function * applied to each element in the array or ‘null’ if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> DoubleArray.maxOfOrNull(selector: (Double) -> R): R? {  
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (maxValue < v) {  
            maxValue = v
        }  
    }  
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function * applied to each element in the array or ‘null’ if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> BooleanArray.maxOfOrNull(selector: (Boolean) -> R): R? {  
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (maxValue < v) {  
            maxValue = v
        }  
    }  
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function * applied to each element in the array or ‘null’ if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> CharArray.maxOfOrNull(selector: (Char) -> R): R? {  
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (maxValue < v) {  
            maxValue = v
        }  
    }  
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function * applied to each element in the array or ‘null’ if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <T, R> Array<out T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {  
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {  
            maxValue = v
        }  
    }  
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function * applied to each element in the array or ‘null’ if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.maxOfWith(comparator: Comparator<in R>, selector: (Byte) -> R): R {  
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {  
            maxValue = v
        }  
    }  
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function * applied to each element in the array or ‘null’ if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R> ShortArray.maxOfWith(comparator: Comparator<in R>, selector: (Short) -> R): R {  
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {  
            maxValue = v
        }  
    }  
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R> IntArray.maxOfWith(comparator: Comparator<in R>, selector: (Int) -> R): R {  
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {  
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {  
            maxValue = v
        }  
    }  
    return maxValue
}
(comparator.compare(maxValue, v) < 0) \{ 
    maxVal = v
    } 
} return maxVal

* Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
* @throws NoSuchElementException if the array is empty.
* @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType<
    kotlin.internal.InlineOnly
    >

fun <R> IntArray.maxOfWith(comparator: Comparator<in R>, selector: (Int) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 * @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType<
    kotlin.internal.InlineOnly
    >

fun <R> LongArray.maxOfWith(comparator: Comparator<in R>, selector: (Long) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 * @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType<
    kotlin.internal.InlineOnly
    >

fun <R> FloatArray.maxOfWith(comparator: Comparator<in R>, selector: (Float) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 * @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType<
    kotlin.internal.InlineOnly
    >

fun <R> DoubleArray.maxOfWith(comparator: Comparator<in R>, selector: (Double) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 * @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType<
    kotlin.internal.InlineOnly
    >

fun <T, R> Array<out T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {
    if (isEmpty()) return null
    ...
var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector]
 * function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R>
    ByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {
        if (isEmpty()) return null
        var maxValue = selector(this[0])
        for (i in 1..lastIndex) {
            val v = selector(this[i])
            if (comparator.compare(maxValue, v) < 0) {
                maxValue = v
            }
        }
        return maxValue
    }

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector]
 * function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R>
    ShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {
        if (isEmpty()) return null
        var maxValue = selector(this[0])
        for (i in 1..lastIndex) {
            val v = selector(this[i])
            if (comparator.compare(maxValue, v) < 0) {
                maxValue = v
            }
        }
        return maxValue
    }

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector]
 * function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R>
    IntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {
        if (isEmpty()) return null
        var maxValue = selector(this[0])
        for (i in 1..lastIndex) {
            val v = selector(this[i])
            if (comparator.compare(maxValue, v) < 0) {
                maxValue = v
            }
        }
        return maxValue
    }

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector]
 * function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R>
    LongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {
        if (isEmpty()) return null
        var maxValue = selector(this[0])
        for (i in 1..lastIndex) {
            val v = selector(this[i])
            if (comparator.compare(maxValue, v) < 0) {
                maxValue = v
            }
        }
        return maxValue
    }

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector]
 * function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R>
    FloatArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {
        if (isEmpty()) return null
        var maxValue = selector(this[0])
        for (i in 1..lastIndex) {
            val v = selector(this[i])
            if (comparator.compare(maxValue, v) < 0) {
                maxValue = v
            }
        }
        return maxValue
    }

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector]
 * function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R>
    DoubleArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {
        if (isEmpty()) return null
        var maxValue = selector(this[0])
        for (i in 1..lastIndex) {
            val v = selector(this[i])
            if (comparator.compare(maxValue, v) < 0) {
                maxValue = v
            }
        }
        return maxValue
    }
BooleanArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? { if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) { val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    } return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector]
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R>
CharArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    } return maxValue
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun Array<out Double>.maxOrNull(): Double? { if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        max = maxOf(max, e)
    } return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun Array<out Float>.maxOrNull(): Float? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        max = maxOf(max, e)
    } return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun <T : Comparable<T>> Array<out T>.maxOrNull(): T? { if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    } return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun ByteArray.maxOrNull(): Byte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    } return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun ShortArray.maxOrNull(): Short? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    } return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun IntArray.maxOrNull(): Int? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    } return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun LongArray.maxOrNull(): Long? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    } return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun CharArray.maxOrNull(): Char? { if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    } return max
}

@Deprecated("Use maxWithOrNull instead.",
ReplacedWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun <T> Array<out T>.maxWith(comparator: Comparator<in T>): T? { if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    } return max
}

@Deprecated("Use maxWithOrNull instead.",
ReplacedWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun ByteArray.maxWith(comparator: Comparator<in Byte>): Byte? { if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    } return max
}

@Deprecated("Use maxWithOrNull instead.",
ReplacedWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun ShortArray.maxWith(comparator: Comparator<Short>): Short? { if (isEmpty()) return null
}
maxWithOrNull(comparator):\n\n@Deprecated("Use maxWithOrNull instead.",\nReplacementWith("maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun
IntArray.maxWith(comparator: Comparator<in Int>): Int? {\n  return
maxWithOrNull(comparator)\n\n@Deprecated("Use maxWithOrNull instead.",\nReplacementWith("maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun
LongArray.maxWith(comparator: Comparator<in Long>): Long? {\n  return
maxWithOrNull(comparator)\n\n@Deprecated("Use maxWithOrNull instead.",\nReplacementWith("maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun
FloatArray.maxWith(comparator: Comparator<in Float>): Float? {\n  return
maxWithOrNull(comparator)\n\n@Deprecated("Use maxWithOrNull instead.",\nReplacementWith("maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun
DoubleArray.maxWith(comparator: Comparator<in Double>): Double? {\n  return
maxWithOrNull(comparator)\n\n@Deprecated("Use maxWithOrNull instead.",\nReplacementWith("maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun
BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>): Boolean? {\n  return
maxWithOrNull(comparator)\n\n@Deprecated("Use maxWithOrNull instead.",\nReplacementWith("maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun
CharArray.maxWith(comparator: Comparator<in Char>): Char? {\n  return
maxWithOrNull(comparator)\n\n/**\n * Returns the first element having the largest value according to the
 * provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun <T> Array<out T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun ByteArray.maxWithOrNull(comparator: Comparator<in Byte>): Byte? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun ShortArray.maxWithOrNull(comparator: Comparator<in Short>): Short? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun IntArray.maxWithOrNull(comparator: Comparator<in Int>): Int? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun LongArray.maxWithOrNull(comparator: Comparator<in Long>): Long? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun FloatArray.maxWithOrNull(comparator: Comparator<in Float>): Float? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun DoubleArray.maxWithOrNull(comparator: Comparator<in Double>): Double? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>): Boolean? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun CharArray.maxWithOrNull(comparator: Comparator<in Char>): Char? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun IntArray.maxWithOrNull(comparator: Comparator<in Int>): Int? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun LongArray.maxWithOrNull(comparator: Comparator<in Long>): Long? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun FloatArray.maxWithOrNull(comparator: Comparator<in Float>): Float? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun DoubleArray.maxWithOrNull(comparator: Comparator<in Double>): Double? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\npublic fun BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>): Boolean? {\n  if (isEmpty()) return null\n  var max = this[0]\n  for (i in 1..lastIndex) {\n    val e = this[i]\n    if (comparator.compare(max, e) < 0) max = e\n  }\n  return max
*/
Boolean? { 
    if (isEmpty()) return null 
    var max = this[0] 
    for (i in 1..lastIndex) { 
        val e = this[i] 
        if (comparator.compare(max, e) < 0) max = e 
    } 
    return max 
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun CharArray.maxWithOrNull(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null 
    var max = this[0] 
    for (i in 1..lastIndex) { 
        val e = this[i] 
        if (comparator.compare(max, e) < 0) max = e 
    } 
    return max 
}

@Deprecated("Use minOrNull instead.", ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun ByteArray.min(): Byte? {
    return minOrNull() 
}

@Deprecated("Use minOrNull instead.", ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun ShortArray.min(): Short? {
    return minOrNull() 
}

@Deprecated("Use minOrNull instead.", ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun IntArray.min(): Int? {
    return minOrNull() 
}

@Deprecated("Use minOrNull instead.", ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun LongArray.min(): Long? {
    return minOrNull() 
}

@Deprecated("Use minOrNull instead.", ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun FloatArray.min(): Float? {
    return minOrNull() 
}

@Deprecated("Use minOrNull instead.", ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun DoubleArray.min(): Double? {
    return minOrNull() 
}

@Deprecated("Use minOrNull instead.", ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharArray.min(): Char? {
    return minOrNull() 
}
Comparable<R>> BooleanArray.minByOrNull(selector: (Boolean) -> R): Boolean? {
    if (isEmpty()) return null
    val minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minElem = this[i]
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <R> CharArray.minByOrNull(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    val minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minElem = this[i]
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun DoubleArray.minOf(selector: (Double) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun FloatArray.minOf(selector: (Float) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun IntArray.minOf(selector: (Int) -> Int): Int {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun LongArray.minOf(selector: (Long) -> Long): Long {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}
*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun FloatArray.minOf(selector: (Float) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun DoubleArray.minOf(selector: (Double) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun BooleanArray.minOf(selector: (Boolean) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun CharArray.minOf(selector: (Char) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun <T> Array<out T>.minOf(selector: (T) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun ByteArray.minOf(selector: (Byte) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun ShortArray.minOf(selector: (Short) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}
`NaN`.

* @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
**\@LambdaReturnType\@InlineOnly\)**

public inline fun IntArray.minOf(selector: (Int) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..<lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
**\@LambdaReturnType\@InlineOnly\)**

public inline fun LongArray.minOf(selector: (Long) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..<lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
**\@LambdaReturnType\@InlineOnly\)**

public inline fun FloatArray.minOf(selector: (Float) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..<lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
**\@LambdaReturnType\@InlineOnly\)**

public inline fun DoubleArray.minOf(selector: (Double) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..<lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
**\@LambdaReturnType\@InlineOnly\)**

public inline fun BooleanArray.minOf(selector: (Boolean) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..<lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throw

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
**\@LambdaReturnType\@InlineOnly\)**

public inline fun <T, R : Comparable<R>> Array<out T>.minOf(selector: (T) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..<lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throw

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
**\@LambdaReturnType\@InlineOnly\)**

public inline fun CharArray.minOf(selector: (Char) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..<lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.
 * @throw

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
**\@LambdaReturnType\@InlineOnly\)**

public inline fun <T, R : Comparable<R>> Array<out T>.minOf(selector: (T) -> R): R {
    if (isEmpty()) throw NoSuchFieldException()
    var minValue = selector(this[0])
    for (i in 1..<lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
array is empty.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ByteArray.minOf(selector: (Byte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ShortArray.minOf(selector: (Short) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> IntArray.minOf(selector: (Int) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> LongArray.minOf(selector: (Long) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> FloatArray.minOf(selector: (Float) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> DoubleArray.minOf(selector: (Double) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> BooleanArray.minOf(selector: (Boolean) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
produced by [selector] function

NoSuchElementException if the array is empty.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.experimental.InlineOnly

public inline fun <R : Comparable<R>>
CharArray.minOf(selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.experimental.InlineOnly

public inline fun <T> Array<out T>.minOfOrNull(selector: (T) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.experimental.InlineOnly

public inline fun ByteArray.minOfOrNull(selector: (Byte) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.experimental.InlineOnly

public inline fun ShortArray.minOfOrNull(selector: (Short) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.experimental.InlineOnly

public inline fun IntArray.minOfOrNull(selector: (Int) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.experimental.InlineOnly

public inline fun LongArray.minOfOrNull(selector: (Long) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.experimental.InlineOnly

public inline fun FloatArray.minOfOrNull(selector: (Float) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}
the smallest value among all values produced by [selector] function

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun DoubleArray.minOfOrNull(selector: (Double) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun BooleanArray.minOfOrNull(selector: (Boolean) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun CharArray.minOfOrNull(selector: (Char) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.minOfOrNull(selector: (T) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun ByteArray.minOfOrNull(selector: (Byte) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun ShortArray.minOfOrNull(selector: (Short) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun IntArray.minOfOrNull(selector: (Int) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {

val v = selector(this[i])

minValue = minOf(minValue, v)

return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
InlineOnly
public inline fun LongArray.minOfOrNull(selector: (Long) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
InlineOnly
public inline fun FloatArray.minOfOrNull(selector: (Float) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
InlineOnly
public inline fun DoubleArray.minOfOrNull(selector: (Double) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
InlineOnly
public inline fun BooleanArray.minOfOrNull(selector: (Boolean) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
InlineOnly
public inline fun CharArray.minOfOrNull(selector: (Char) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
InlineOnly
public inline fun <T, R : Comparable<R>> Array<out T>.minOfOrNull(selector: (T) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
InlineOnly
public inline fun <R : Comparable<R>> ByteArray.minOfOrNull(selector: (Byte) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
shortArray.minOfOrNull(selector: (Short) -> R): R?

```
\nminValue = v
for (i in 1..lastIndex) { 
    val v = selector(this[i])
    if (minValue > v) {
        minValue = v
    }
}
return minValue
```

public fun <R : Comparable<R>> ShortArray.minOfOrNull(selector: (Short) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n *
 * @return \n * Returns the smallest value among all values produced
 * by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n *
 * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public fun <T : Comparable<T>> Array<T>.minOfWith(comparator: Comparator<T>, selector: (T) -> T): T {
    if (isEmpty()) throw
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
NoSuchElementException()
    val minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.minOfWith(comparator: Comparator<in R>, selector: (Byte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> ShortArray.minOfWith(comparator: Comparator<in R>, selector: (Short) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> IntArray.minOfWith(comparator: Comparator<in R>, selector: (Int) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> LongArray.minOfWith(comparator: Comparator<in R>, selector: (Long) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> FloatArray.minOfWith(comparator: Comparator<in R>, selector: (Float) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> DoubleArray.minOfWith(comparator: Comparator<in R>, selector: (Double) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}
ByLambdaReturnType

@kotlin.internal.InlineOnly

public inline fun <R> BooleanArray.minOfWith(comparator: Comparator<in R>, selector: (Boolean) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator][n] among all values produced by [selector] function applied to each element in the array. If there are no elements, returns `null`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.internal.InlineOnly
public inline fun <R> CharArray.minOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator][n] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.internal.InlineOnly
public inline fun <T, R> Array<out T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator][n] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator][n] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.internal.InlineOnly
public inline fun <R> ShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator][n] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.internal.InlineOnly
public inline fun <R> IntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator][n] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.internal.InlineOnly
public inline fun <R> LongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator][n] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 */
FloatArray.minOfWithOrNull(comparator: Comparator<R>, selector: (Float) -> R): R? { \n    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) { \n        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) { \n            minValue = v
        }
    }
    return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

DoubleArray.minOfWithOrNull(comparator: Comparator<R>, selector: (Double) -> R): R? { \n    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) { \n        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) { \n            minValue = v
        }
    }
    return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

CharArray.minOfWithOrNull(comparator: Comparator<R>, selector: (Char) -> R): R? { \n    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) { \n        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) { \n            minValue = v
        }
    }
    return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

Array<out Double>.minOrNull(): Double? { \n    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) { \n        val e = this[i]
        min = minOf(min, e)
    }
    return min
}

Array<out Float>.minOrNull(): Float? { \n    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) { \n        val e = this[i]
        min = minOf(min, e)
    }
    return min
}

Array<out T : Comparable<T>>.minOrNull(): T? { \n    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) { \n        val e = this[i]
        if (min > e) min = e
    }
    return min
}

ByteArray.minOrNull(): Byte? { \n    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) { \n        val e = this[i]
        if (min > e) min = e
    }
    return min
}

ShortArray.minOrNull(): Short? { \n    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) { \n        val e = this[i]
        if (min > e) min = e
    }
    return min
}

IntArray.minOrNull(): Int? { \n    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) { \n        val e = this[i]
        if (min > e) min = e
    }
    return min
}

LongArray.minOrNull(): Long? { \n    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) { \n        val e = this[i]
        if (min > e) min = e
    }
    return min
}

FloatArray.minOrNull(): Float? { \n    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) { \n        val e = this[i]
        if (min > e) min = e
    }
    return min
}
Returns the smallest element or `null` if there are no elements.

`@SinceKotlin("1.4")` public fun minOrNull(): Double? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        min = minOf(min, e)
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * If any of elements is `NaN` returns `NaN`.
 */
@SinceKotlin("1.4")
public fun CharArray.minOrNull(): Char? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}

@Deprecated("Use minWithOrNull instead.\n", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun ByteArray.minWithOrNull(comparator: Comparator<in Byte>): Byte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun BooleanArray.minWithOrNull(comparator: Comparator<in Boolean>): Boolean? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@Deprecated("Use minWithOrNull instead.\n", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharArray.minWithOrNull(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun ShortArray.minWithOrNull(comparator: Comparator<in Short>): Short? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun IntArray.minWithOrNull(comparator: Comparator<in Int>): Int? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@Deprecated("Use minWithOrNull instead.\n", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun LongArray.minWithOrNull(comparator: Comparator<in Long>): Long? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun <T> Array<out T>.minWithOrNull(comparator: Comparator<in T>): T? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
isNotEmpty()) return null
var min = this[0]
for (i in 1..lastIndex) {
    val e = this[i]
    if (comparator.compare(min, e) > 0) min = e
}
return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun LongArray.minWithOrNull(comparator: Comparator<in Long>): Long? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun FloatArray.minWithOrNull(comparator: Comparator<in Float>): Float? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun DoubleArray.minWithOrNull(comparator: Comparator<in Double>): Double? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns `true` if the array has no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.none
 */
public fun <T> Array<out T>.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun <T> Array<out T>.none(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
Returns `true` if no elements match the given [predicate].

```kotlin
public inline fun IntArray.none(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun LongArray.none(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun FloatArray.none(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun DoubleArray.none(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
public inline fun CharArray.none(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.onEach(action: (T) -> Unit): Array<out T> {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ByteArray.onEach(action: (Byte) -> Unit): ByteArray {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ShortArray.onEach(action: (Short) -> Unit): ShortArray {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun IntArray.onEach(action: (Int) -> Unit): IntArray {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun LongArray.onEach(action: (Long) -> Unit): LongArray {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun FloatArray.onEach(action: (Float) -> Unit): FloatArray {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.onEach(action: (Double) -> Unit): DoubleArray {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun BooleanArray.onEach(action: (Boolean) -> Unit): BooleanArray {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun CharArray.onEach(action: (Char) -> Unit): CharArray {
    return apply { for (element in this) action(element) }
}
```

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.onEach(action: (T) -> Unit, context: Any): Array<out T> {
    return apply { for (element in this) action(element, context) }
}
```
accumulator.value
\n* \n* @sample samples.collections.Collections.Aggregates.reduce
\n*\n/unpublic inline fun

ByteArray.reduce(operation: (acc: Byte, Byte) -> Byte): Byte {
    if (isEmpty()) throw
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
        return accumulator
    }
}

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element.

Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

ShortArray.reduce(operation: (acc: Short, Short) -> Short): Short {
    if (isEmpty()) throw
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
        return accumulator
    }
}

IntArray.reduce(operation: (acc: Int, Int) -> Int): Int {
    if (isEmpty()) throw
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
        return accumulator
    }
}

LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {
    if (isEmpty()) throw
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
        return accumulator
    }
}

FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {
    if (isEmpty()) throw
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
        return accumulator
    }
}

DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {
    if (isEmpty()) throw
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
        return accumulator
    }
}
accumulator.value.* @sample samples.collections.Collections.Aggregates.reduce

//public inline fun BooleanArray.reduce(operation: (acc: Boolean, Boolean) -> Boolean): Boolean {
  if (isEmpty()) throw
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(accumulator, this[index])
  }
  return accumulator
}

//Accumulates value starting with the first element and applying [operation] from left to right
//to current accumulator value and each element.
//Throws an exception if this array is empty. If the array can be empty in an expected way,
//please use [reduceOrNullable] instead. It returns `null` when its receiver is empty.
//@param [operation] function that takes current accumulator value and an element,
//and calculates the next accumulator value.
//* @sample samples.Collections.Aggregates.reduce

//public inline fun CharArray.reduce(operation: (acc: Char, Char) -> Char): Char {
  if (isEmpty()) throw
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(accumulator, this[index])
  }
  return accumulator
}

//Accumulates value starting with the first element and applying [operation] from left to right
//to current accumulator value and each element with its index in the original array.
//Throws an exception if this array is empty. If the array can be empty in an expected way,
//please use [reduceIndexedOrNullable] instead. It returns `null` when its receiver is empty.
//@param [operation] function that takes the index of an element, current accumulator value and
//the element itself,
//and calculates the next accumulator value.
//@sample samples.collections.Collections.Aggregates.reduce

//public inline fun <S, T : S> Array<out T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {
  if (isEmpty()) throw
  var accumulator: S = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(index, accumulator, this[index])
  }
  return accumulator
}

//Accumulates value starting with the first element and applying [operation] from left to right
//to current accumulator value and each element with its index in the original array.
//Throws an exception if this array is empty. If the array can be empty in an expected way,
//please use [reduceIndexedOrNullable] instead. It returns `null` when its receiver is empty.
//@param [operation] function that takes the index of an element, current accumulator value and
//the element itself,
//and calculates the next accumulator value.
//@sample samples.collections.Collections.Aggregates.reduce

//public inline fun Array<Byte>.reduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte {
  if (isEmpty()) throw
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(index, accumulator, this[index])
  }
  return accumulator
}

//Accumulates value starting with the first element and applying [operation] from left to right
//to current accumulator value and each element with its index in the original array.
//Throws an exception if this array is empty. If the array can be empty in an expected way,
//please use [reduceIndexedOrNullable] instead. It returns `null` when its receiver is empty.
//@param [operation] function that takes the index of an element, current accumulator value and
//the element itself,
//and calculates the next accumulator value.
//@sample samples.collections.Collections.Aggregates.reduce

//public inline fun Array<Int>.reduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): Int {
  if (isEmpty()) throw
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(index, accumulator, this[index])
  }
  return accumulator
}

//Accumulates value starting with the first element and applying [operation] from left to right
//to current accumulator value and each element with its index in the original array.
//Throws an exception if this array is empty. If the array can be empty in an expected way,
//please use [reduceIndexedOrNullable] instead. It returns `null` when its receiver is empty.
//@param [operation] function that takes the index of an element, current accumulator value and
//the element itself,
//and calculates the next accumulator value.
//@sample samples.collections.Collections.Aggregates.reduce
Accumulates value starting with the first element and applying [operation] from left to right. 

- **LongArray**:
  ```kotlin
      if (isEmpty())
          throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
          accumulator = operation(index, accumulator, this[index])
      }
      return accumulator
  }
  ```

- **FloatArray**:
  ```kotlin
  public inline fun FloatArray.reduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): Float {
      if (isEmpty())
          throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
          accumulator = operation(index, accumulator, this[index])
      }
      return accumulator
  }
  ```

- **DoubleArray**:
  ```kotlin
  public inline fun DoubleArray.reduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): Double {
      if (isEmpty())
          throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
          accumulator = operation(index, accumulator, this[index])
      }
      return accumulator
  }
  ```

- **BooleanArray**:
  ```kotlin
  public inline fun BooleanArray.reduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean {
      if (isEmpty())
          throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
          accumulator = operation(index, accumulator, this[index])
      }
      return accumulator
  }
  ```

- **CharArray**:
  ```kotlin
  public inline fun CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {
      if (isEmpty())
          throw UnsupportedOperationException("Empty array can't be reduced.")
      var accumulator = this[0]
      for (index in 1..lastIndex) {
          accumulator = operation(index, accumulator, this[index])
      }
      return accumulator
  }
  ```

These functions accumulate values starting with the first element and applying the specified operation from left to right, accumulating the result for each element and its index in the original array. If the array is empty, they throw an exception. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead when the index of an element, current accumulator value, and the element itself are not available. The functions return `null` when the array is empty.
value and the element itself, \( n \) * and calculates the next accumulator value, \( n \) * \( n \) * @sample samples.collections.Collections.Aggregates.reduceOrNull

*\@SinceKotlin("1.4\") */

public inline fun <S, T : S> Array<out T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {
    if (isEmpty())
        return null
    var accumulator: S = this[0] for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/*
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, \( n \) * and calculates the next accumulator value, \( n \) * \( n \) * @sample samples.collections.Collections.Aggregates.reduceOrNull

*\@SinceKotlin("1.4\") */

public inline fun ByteArray.reduceIndexedOrNull(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/*
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, \( n \) * and calculates the next accumulator value, \( n \) * \( n \) * @sample samples.collections.Collections.Aggregates.reduceOrNull

*\@SinceKotlin("1.4\") */

public inline fun ShortArray.reduceIndexedOrNull(operation: (index: Int, acc: Short, Short) -> Short): Short? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/*
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, \( n \) * and calculates the next accumulator value, \( n \) * \( n \) * @sample samples.collections.Collections.Aggregates.reduceOrNull

*\@SinceKotlin("1.4\") */

public inline fun IntArray.reduceIndexedOrNull(operation: (index: Int, acc: Int, Int) -> Int): Int? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/*
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, \( n \) * and calculates the next accumulator value, \( n \) * \( n \) * @sample samples.collections.Collections.Aggregates.reduceOrNull

*\@SinceKotlin("1.4\") */

    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/*
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, \( n \) * and calculates the next accumulator value, \( n \) * \( n \) * @sample samples.collections.Collections.Aggregates.reduceOrNull

*\@SinceKotlin("1.4\") */

public inline fun FloatArray.reduceIndexedOrNull(operation: (index: Int, acc: Float, Float) -> Float): Float? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/*
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, \( n \) * and calculates the next accumulator value, \( n \) * \( n \) * @sample samples.collections.Collections.Aggregates.reduceOrNull

*\@SinceKotlin("1.4\") */

public inline fun DoubleArray.reduceIndexedOrNull(operation: (index: Int, acc: Double, Double) -> Double): Double? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * 
 * Returns `null` if the array is empty.
 * 
 * @param [operation] function that takes current accumulator value and an element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
public inline fun BooleanArray.reduceOrNull(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * 
 * Returns `null` if the array is empty.
 * 
 * @param [operation] function that takes current accumulator value and an element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
public inline fun CharArray.reduceOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * 
 * Returns `null` if the array is empty.
 * 
 * @param [operation] function that takes current accumulator value and an element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
public inline fun ByteArray.reduceOrNull(operation: (acc: Byte, Byte) -> Byte): Byte? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * 
 * Returns `null` if the array is empty.
 * 
 * @param [operation] function that takes current accumulator value and an element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
public inline fun ShortArray.reduceOrNull(operation: (acc: Short, Short) -> Short): Short? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * 
 * Returns `null` if the array is empty.
 * 
 * @param [operation] function that takes current accumulator value and an element.
 * 
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
public inline fun IntArray.reduceOrNull(operation: (acc: Int, Int) -> Int?): Int? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
---

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun FloatArray.reduceOrNull(operation: (acc: Float, Float) -> Float): Float? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun DoubleArray.reduceOrNull(operation: (acc: Double, Double) -> Double): Double? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun BooleanArray.reduceOrNull(operation: (acc: Boolean, Boolean) -> Boolean): Boolean? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to current accumulator value and each element.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
public inline fun <S, T : S> Array<out T>.reduceRight(operation: (T, acc: S) -> S): S {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * Throws an exception if this array is empty.
 * If the array can be empty in an expected way, please use [reduceRightOrNull] instead. It returns `null`
 * when its receiver is empty.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
public inline fun ByteArray.reduceRight(operation: (Byte, acc: Byte) -> Byte): Byte {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

---

Open Source Used In DNAC Platform Antelope 9227
throw UnsupportedOperationException("Empty array can't be reduced.");
var accumulator = get(index--)
while (index >= 0) {
    accumulator = operation(accumulator, get(index--))
} return accumulator

Accumulates value starting with the last element and applying [operation] from right to left and current accumulator value.

Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceRightOrNull] instead. It returns 'null' when its receiver is empty.

and calculates the next accumulator value.

@example

Open Source Used In DNAC Platform Antelope 9228
and calculates the next accumulator value.

```kotlin
public inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

And calculates the next accumulator value starting with the last element and applying [operation] from right to left.

```kotlin
public inline fun CharArray.reduceRight(operation: (Char, acc: Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with the last element and applying [operation] from right to left.

```kotlin
public inline fun <S, T : S> Array<out T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator: S = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

Accumulates value starting with the last element and applying [operation] from right to left.

```kotlin
public inline fun ByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```
Open Source Used In DNAC Platform Antelope

```kotlin
/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and
 * current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun BooleanArray.reduceRightIndexed(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
```

---

Open Source Used In DNAC Platform Antelope 9230
value, * and calculates the next accumulator value.

```kotlin
samples.collections.Collections.Aggregates.reduceRightIndexedOrNull
```

public inline fun CharArray.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left on each element with its index in the original array and current accumulator value.

* @param value

Retruns `null` if the array is empty.

```kotlin
samples.collections.Collections.Aggregates.reduceRightOrNullPointerException
```

public inline fun ByteArray.reduceRightIndexedOrNullPointerException(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

* Returns `null` if the array is empty.

```kotlin
samples.collections.Collections.Aggregates.reduceRightOrNull
```

public inline fun ShortArray.reduceRightIndexedOrNull(operation: (index: Int, Short, acc: Short) -> Short): Short? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

* @param value

```

Accumulates value starting with the last element and applying [operation] from right to left on each element with its index in the original array and current accumulator value.

```kotlin
samples.collections.Collections.Aggregates.reduceRightOrNullPointerException
```

public inline fun IntArray.reduceRightIndexedOrNullPointerException(operation: (index: Int, Int, acc: Int) -> Int): Int? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

* Returns `null` if the array is empty.

```kotlin
samples.collections.Collections.Aggregates.reduceRightOrNull
```

    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

* Returns `null` if the array is empty.

```kotlin
samples.collections.Collections.Aggregates.reduceRightOrNullPointerException
```

public inline fun Open Source Used In DNAC Platform Antelope 9231

public inline fun
* and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
public inline fun FloatArray.reduceRightIndexedOrNull(operation: (index: Int, Float, acc: Float) -> Float): Float? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
public inline fun DoubleArray.reduceRightIndexedOrNull(operation: (index: Int, Double, acc: Double) -> Double): Double? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
public inline fun BooleanArray.reduceRightIndexedOrNull(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> Array<out T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun ByteArray.reduceRightOrNull(operation: (Byte, acc: Byte) -> Byte): Byte? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```
samples.collections.Collections.Aggregates.reduceRightOrNull

@SinceKotlin("1.4").@WasExperimental(ExperimentalStdlibApi::class)public inline fun
ShortArray.reduceRightOrNull(operation: (Short, acc: Short) -> Short): Short? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, acc)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value.
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4").@WasExperimental(ExperimentalStdlibApi::class)public inline fun
IntArray.reduceRightOrNull(operation: (Int, acc: Int) -> Int): Int? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, acc)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value.
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4").@WasExperimental(ExperimentalStdlibApi::class)public inline fun
LongArray.reduceRightOrNull(operation: (Long, acc: Long) -> Long): Long? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, acc)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value.
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4").@WasExperimental(ExperimentalStdlibApi::class)public inline fun
FloatArray.reduceRightOrNull(operation: (Float, acc: Float) -> Float): Float? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, acc)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value.
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4").@WasExperimental(ExperimentalStdlibApi::class)public inline fun
DoubleArray.reduceRightOrNull(operation: (Double, acc: Double) -> Double): Double? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, acc)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value.
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4").@WasExperimental(ExperimentalStdlibApi::class)public inline fun
BooleanArray.reduceRightOrNull(operation: (Boolean, acc: Boolean) -> Boolean): Boolean? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, acc)
    }
    return accumulator
}
samples.collections.Collections.Aggregates.reduceRightOrNull

*\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun

CharArray.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {
    var index = lastIndex
    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n        return accumulator\n    }\n    return null
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right* to each element and current
 * accumulation value that starts with [initial] value.* Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.*\n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.*\n * Calculates the next accumulator value;\n * @param samples.collections.Collections.Aggregates.runningFold

*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n        return result\n    }\n    return null
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right* to each element and current
 * accumulation value that starts with [initial] value.* Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.*\n * Calculates the next accumulator value;\n * @param samples.collections.Collections.Aggregates.runningFold

*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.runningFold(initial: R, operation: (acc: R, Byte) -> R): List<R> {\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n        return result\n    }\n    return null
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right* to each element and current
 * accumulation value that starts with [initial] value.* Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.*\n * Calculates the next accumulator value;\n * @param samples.collections.Collections.Aggregates.runningFold

*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.runningFold(initial: R, operation: (acc: R, Short) -> R): List<R> {\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n        return result\n    }\n    return null
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right* to each element and current
 * accumulation value that starts with [initial] value.* Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.*\n * Calculates the next accumulator value;\n * @param samples.collections.Collections.Aggregates.runningFold

*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.runningFold(initial: R, operation: (acc: R, Int) -> R): List<R> {\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n        return result\n    }\n    return null
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right* to each element and current
 * accumulation value that starts with [initial] value.* Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.*\n * Calculates the next accumulator value;\n * @param samples.collections.Collections.Aggregates.runningFold

*\n@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.runningFold(initial: R, operation: (acc: R, Long) -> R): List<R> {\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n        return result\n    }\n    return null
}

Open Source Used In DNAC Platform Antelope 9234
+ 1).apply { add(initial) }\n    var accumulator = initial\n    for (element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n    return result\n} \n
@SinceKotlin("1.4")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.runningFoldIndexed(initial: T, operation: (index: Int, acc: T, T) -> T): List<T> {\n    \n    if (isEmpty()) return listOf(initial)\n    val result = ArrayList<T>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for (index in indices) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result
}
values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. The function takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

```kotlin
ByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial

    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

```kotlin
ShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial

    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

```kotlin
IntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial

    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

```kotlin
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial

    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

```kotlin
FloatArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
```
for (index in indices) {
    accumulator = operation(index, accumulator, this[index])
}

result.add(accumulator)
return result

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningFold

*\* @SinceKotlin("1.4")

DoubleArray.runningFoldIndexed(initial: R, operation: (Int, acc: Double, Double) -> R): List<Double> {
    if (isEmpty()) return listOf(initial).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningFold

*\* @SinceKotlin("1.4")

BooleanArray.runningFoldIndexed(initial: R, operation: (Int, acc: Boolean, Boolean) -> R): List<Boolean> {
    if (isEmpty()) return listOf(initial).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningFold

*\* @SinceKotlin("1.4")

CharArray.runningFoldIndexed(initial: R, operation: (Int, acc: Char, Char) -> R): List<Char> {
    if (isEmpty()) return listOf(initial).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

ByteArray.runningReduce(operation: (Byte, Byte) -> Byte): List<Byte> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

ShortArray.runningReduce(operation: (Short, Short) -> Short): List<Short> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

IntArray.runningReduce(operation: (Int, Int) -> Int): List<Int> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

LongArray.runningReduce(operation: (Long, Long) -> Long): List<Long> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

CharArray.runningReduce(operation: (Char, Char) -> Char): List<Char> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

ByteArray.runningReduce(operation: (Byte, Byte) -> Byte): List<Byte> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

ShortArray.runningReduce(operation: (Short, Short) -> Short): List<Short> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

IntArray.runningReduce(operation: (Int, Int) -> Int): List<Int> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\* @sample samples.collections.Aggregates.runningReduce

*\* @SinceKotlin("1.4")

LongArray.runningReduce(operation: (Long, Long) -> Long): List<Long> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

*\* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

*\* Note that ‘acc‘ value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

*\* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
operation(accumulator, this[index])
    result.add(accumulator)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4.1")
@kotlin.internal.InlineOnly
public inline fun ShortArray.runningReduce(operation: (acc: Short, Short) -> Short): List<Short> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Short>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4.1")
@kotlin.internal.InlineOnly
public inline fun IntArray.runningReduce(operation: (acc: Int, Int) -> Int): List<Int> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Int>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4.1")
@kotlin.internal.InlineOnly
public inline fun LongArray.runningReduce(operation: (acc: Long, Long) -> Long): List<Long> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Long>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4.1")
@kotlin.internal.InlineOnly
public inline fun FloatArray.runningReduce(operation: (acc: Float, Float) -> Float): List<Float> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Float>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4.1")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.runningReduce(operation: (acc: Double, Double) -> Double): List<Double> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Double>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4.1")
@kotlin.internal.InlineOnly
public inline fun BooleanArray.runningReduce(operation: (acc: Boolean, Boolean) -> Boolean): List<Boolean> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Boolean>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each
 * element and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator
 * value.
 * @sample sample.samples.collections.Collections.Aggregates.runningReduce
 * @SinceKotlin("1.4")
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Char>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each
 * element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates
 * the next accumulator value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect
 * the previous value in resulting list.
 * @sample sample.samples.collections.Collections.Aggregates.runningReduceIndexed
 * @SinceKotlin("1.4")
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {
    if (isEmpty()) return emptyList()
    var accumulator: Char = this[0]
    val result = ArrayList<Char>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each
 * element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates
 * the next accumulator value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect
 * the previous value in resulting list.
 * @sample sample.samples.collections.Collections.Aggregates.runningReduceIndexed
 * @SinceKotlin("1.4")
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.runningReduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): List<Byte> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Byte>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each
 * element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates
 * the next accumulator value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect
 * the previous value in resulting list.
 * @sample sample.samples.collections.Collections.Aggregates.runningReduceIndexed
 * @SinceKotlin("1.4")
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.runningReduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): List<Short> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Short>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each
 * element, its index in the original array and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates
 * the next accumulator value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect
 * the previous value in resulting list.
 * @sample sample.samples.collections.Collections.Aggregates.runningReduceIndexed
 * @SinceKotlin("1.4")
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.runningReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Int>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
takes the index of an element, current accumulator value, * and the element itself, and calculates the next accumulator value. *\n * *\n * @sample samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Long>(size).apply {
        add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array. *\n * * @param [operation] function that takes the index of an element, current accumulator value, * and the element itself, and calculates the next accumulator value. *\n * * * @sample samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun FloatArray.runningReduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): List<Float> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Float>(size).apply {
        add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array. *\n * * @param [operation] function that takes the index of an element, current accumulator value, * and the element itself, and calculates the next accumulator value. *\n * * * @sample samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.runningReduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): List<Double> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Double>(size).apply {
        add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array. *\n * * @param [operation] function that takes the index of an element, current accumulator value, * and the element itself, and calculates the next accumulator value. *\n * * * @sample samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun BooleanArray.runningReduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): List<Boolean> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Boolean>(size).apply {
        add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array. *\n * * @param [operation] function that takes the index of an element, current accumulator value, * and the element itself, and calculates the next accumulator value. *\n * * * @sample samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun CharArray.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Char>(size).apply {
        add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
samples.collections.Collections.Aggregates.scan
*

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <T, R>
Array<out T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {
    return runningFold(initial, operation)
}

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* Otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.scan
*

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.scan(initial: R, operation: (acc: R, Byte) -> R): List<R> {
    return runningFold(initial, operation)
}

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.scan
*

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> ShortArray.scan(initial: R, operation: (acc: R, Short) -> R): List<R> {
    return runningFold(initial, operation)
}

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.scan
*

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> IntArray.scan(initial: R, operation: (acc: R, Int) -> R): List<R> {
    return runningFold(initial, operation)
}

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.scan
*

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> LongArray.scan(initial: R, operation: (acc: R, Long) -> R): List<R> {
    return runningFold(initial, operation)
}

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.scan
*

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> FloatArray.scan(initial: R, operation: (acc: R, Float) -> R): List<R> {
    return runningFold(initial, operation)
}

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.scan
*

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> DoubleArray.scan(initial: R, operation: (acc: R, Double) -> R): List<R> {
    return
runningFold(initial, operation)\n\n* Returns a list containing successive accumulation values generated by applying [operation] from left to right\n* to each element and current accumulator value that starts with [initial] value.\n* Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n* @sample samples.collections.Collections.Aggregates.scan

```kotlin
/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n* to each element and current accumulator value that starts with [initial] value.\n* Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n* @sample samples.collections.Collections.Aggregates.scan

```
runningFoldIndexed(initial, operation)\n\n/**
 * Returns a list containing successive accumulation values
 generated by applying [operation] from left to right for each element, its index in the original array and current
 accumulation value that starts with [initial] value.\n * Note that `acc` value passed to [operation] function should
 not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation]
 * function that takes the index of an element, current accumulator value and the element itself, and calculates
 the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}

/**
 * Returns a list containing successive accumulation values
 generated by applying [operation] from left to right for each element, its index in the original array and current
 accumulation value that starts with [initial] value.\n * Note that `acc` value passed to [operation] function should
 not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation]
 * function that takes the index of an element, current accumulator value and the element itself, and calculates
 the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}

/**
 * Returns a list containing successive accumulation values
 generated by applying [operation] from left to right for each element, its index in the original array and current
 accumulation value that starts with [initial] value.\n * Note that `acc` value passed to [operation] function should
 not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation]
 * function that takes the index of an element, current accumulator value and the element itself, and calculates
 the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}

/**
 * Returns a list containing successive accumulation values
 generated by applying [operation] from left to right for each element, its index in the original array and current
 accumulation value that starts with [initial] value.\n * Note that `acc` value passed to [operation] function should
 not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation]
 * function that takes the index of an element, current accumulator value and the element itself, and calculates
 the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}

/**
 * Returns a list containing successive accumulation values
 generated by applying [operation] from left to right for each element, its index in the original array and current
 accumulation value that starts with [initial] value.\n * Note that `acc` value passed to [operation] function should
 not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation]
 * function that takes the index of an element, current accumulator value and the element itself, and calculates
 the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n    return runningFoldIndexed(initial, operation)\n}

@Deprecated("Use runningReduce instead.",
ReplaceWith("runningReduce(operation)"), level =
DeprecationLevel.ERROR)\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic inline fun <S, T : S> Array<out T>.scanReduce(operation: (acc: S, T) -> S): List<S> {\n    return runningReduce(operation)\n}

@Deprecated("Use runningReduce instead.",
ReplaceWith("runningReduce(operation)"), level =
DeprecationLevel.ERROR)\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi\npublic inline fun ByteArray.scanReduce(operation: (acc: Byte, Byte) -> Byte): List<Byte> {\n    return
return runningReduceIndexed(operation)@n}@n@Deprecated("Use runningReduceIndexed instead.",
ReplaceWith("runningReduceIndexed(operation)")", level =
DeprecationLevel.ERROR}@n]@SinceKotlin("1.3")}@n]@ExperimentalStdlibApi}@n]@kotlin.internal.InlineOnly}unpublished inline fun FloatArray.scanReduceIndexed(operation: (Int, acc: Float, Float) -> Float): List<Float> {
    \n return runningReduceIndexed(operation)@n}@n@Deprecated("Use runningReduceIndexed instead.",
ReplaceWith("runningReduceIndexed(operation)")", level =
DeprecationLevel.ERROR}@n]@SinceKotlin("1.3")}@n]@ExperimentalStdlibApi}@n]@kotlin.internal.InlineOnly}unpublished inline fun DoubleArray.scanReduceIndexed(operation: (Int, acc: Double, Double) -> Double): List<Double> {
    \n return runningReduceIndexed(operation)@n}@n@Deprecated("Use runningReduceIndexed instead.",
ReplaceWith("runningReduceIndexed(operation)")", level =
DeprecationLevel.ERROR}@n]@SinceKotlin("1.3")}@n]@ExperimentalStdlibApi}@n]@kotlin.internal.InlineOnly}unpublished inline fun BooleanArray.scanReduceIndexed(operation: (Int, acc: Boolean, Boolean) -> Boolean): List<Boolean> {
    \n return runningReduceIndexed(operation)@n}@n@Deprecated("Use runningReduceIndexed instead.",
ReplaceWith("runningReduceIndexed(operation)")", level =
DeprecationLevel.ERROR}@n]@SinceKotlin("1.3")}@n]@ExperimentalStdlibApi}@n]@kotlin.internal.InlineOnly}unpublished inline fun CharArray.scanReduceIndexed(operation: (Int, acc: Char, Char) -> Char): List<Char> {
    \n return runningReduceIndexed(operation)@n}@n@Deprecated("Use runningReduceIndexed instead.",
ReplaceWith("runningReduceIndexed(operation)")", level =
DeprecationLevel.ERROR}@n]@SinceKotlin("1.3")}@n]@ExperimentalStdlibApi}@n]@kotlin.internal.InlineOnly}unpublished inline fun <T> Array<out T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ByteArray.sumBy(selector: (Byte) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ShortArray.sumBy(selector: (Short) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun IntArray.sumBy(selector: (Int) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun LongArray.sumBy(selector: (Long) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun FloatArray.sumBy(selector: (Float) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun DoubleArray.sumBy(selector: (Double) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun BooleanArray.sumBy(selector: (Boolean) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun CharArray.sumBy(selector: (Char) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun <T> Array<out T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ByteArray.sumByDouble(selector: (Byte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ShortArray.sumByDouble(selector: (Short) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun IntArray.sumByDouble(selector: (Int) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun LongArray.sumByDouble(selector: (Long) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun FloatArray.sumByDouble(selector: (Float) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun DoubleArray.sumByDouble(selector: (Double) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun CharArray.sumByDouble(selector: (Char) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
element in the array.

*\public inline fun IntArray.sumByDouble(selector: (Int) -> Double): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n
Returns the sum of all values produced by [selector] function applied to each element in the array.

*\public inline fun LongArray.sumByDouble(selector: (Long) -> Double): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n
Returns the sum of all values produced by [selector] function applied to each element in the array.

*\public inline fun FloatArray.sumByDouble(selector: (Float) -> Double): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n
Returns the sum of all values produced by [selector] function applied to each element in the array.

*\public inline fun DoubleArray.sumByDouble(selector: (Double) -> Double): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n
Returns the sum of all values produced by [selector] function applied to each element in the array.

*\public inline fun BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n
Returns the sum of all values produced by [selector] function applied to each element in the array.

*\public inline fun CharArray.sumByDouble(selector: (Char) -> Double): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += selector(element)\n    }\n    return sum\n}\n
Returns the sum of all values produced by [selector] function applied to each element in the array.

*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\n\n\n\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\n\n\n\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\n\n\n\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\n\n\n\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\n\n\n\n
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfDouble")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.sumOf(selector: (Double) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfDouble")
@kotlin.internal.InlineOnly
public inline fun BooleanArray.sumOf(selector: (Boolean) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfDouble")
@kotlin.internal.InlineOnly
public inline fun CharArray.sumOf(selector: (Char) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.sumOf(selector: (T) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun ByteArray.sumOf(selector: (Byte) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun ShortArray.sumOf(selector: (Short) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun IntArray.sumOf(selector: (Int) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun LongArray.sumOf(selector: (Long) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun FloatArray.sumOf(selector: (Float) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType

@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.sumOf(selector: (Double) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
function applied to each element in the array.

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
CharArray.sumOf(selector: (Char) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
ByteArray.sumOf(selector: (Byte) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
ShortArray.sumOf(selector: (Short) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
IntArray.sumOf(selector: (Int) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
LongArray.sumOf(selector: (Long) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
FloatArray.sumOf(selector: (Float) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
DoubleArray.sumOf(selector: (Double) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
BooleanArray.sumOf(selector: (Boolean) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```
BooleanArray.sumOf(selector: (Boolean) -> Long): Long { return sum

  for (element in this) { sum += selector(element) } return sum }
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfUInt")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun CharArray.sumOf(selector: (Char) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun <T> Array<out T>.sumOf(selector: (T) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun ByteArray.sumOf(selector: (Byte) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun ShortArray.sumOf(selector: (Short) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun IntArray.sumOf(selector: (Int) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun LongArray.sumOf(selector: (Long) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun FloatArray.sumOf(selector: (Float) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun DoubleArray.sumOf(selector: (Double) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal
.InlineOnly
public inline fun BooleanArray.sumOf(selector: (Boolean) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

Open Source Used In DNAC Platform Antelope 9250
0.toULong() for (element in this) {\n    sum += selector(element)\n} return sum\n/**\n * Returns an original collection containing all the non-null elements, throwing an [IllegalArgumentExce
 @ Suppress("UNCHECKED_CAST")\n} return this as Array<T>\n
/**\n * Returns an original collection containing all the non-null elements, throwing an [IllegalArgumentException]\nif there are any null elements.\n */\npublic fun <T : Any> Array<Array<T?>>.requireNoNulls(): Array<Array<T>> {\n    for (element in this) {\n        if (element == null) {\n            throw IllegalArgumentExce
    @ Suppress("UNCHECKED_CAST")\n} return this as Array<T>\n
/**\n * Returns an original collection containing all the non-null elements, throwing an [IllegalArgumentException]\nif there are any null elements.\n */\npublic fun <T : Any> Array<T?>.requireNoNulls(): Array<T> {\n    for (element in this) {\n        if (element == null) {\n            throw IllegalArgumentException("null element found in $this.")\n    }\n}

/**\n * Returns an original collection containing all the non-null elements, throwing an [IllegalArgumentException]\nif there are any null elements.\n */\npublic fun <T : Any> Array<T>.requireNoNulls(): Array<T> {\n    for (element in this) {\n        if (element == null) {\n            throw IllegalArgumentException("null element found in $this.")\n    }\n}

/**\n * Splits the original array into pair of lists,\n */\npublic inline fun <T : Any> Array<T?>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n    val first = ArrayList<T>()\n    val second = ArrayList<T>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}

/**\n * Splits the original array into pair of lists,\n */\npublic inline fun <T : Any> ByteArray.partition(predicate: (Byte) -> Boolean): Pair<List<Byte>, List<Byte>> {\n    val first = ArrayList<Byte>()\n    val second = ArrayList<Byte>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}

/**\n * Splits the original array into pair of lists,\n */\npublic inline fun <T : Any> ShortArray.partition(predicate: (Short) -> Boolean): Pair<List<Short>, List<Short>> {\n    val first = ArrayList<Short>()\n    val second = ArrayList<Short>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}

/**\n * Splits the original array into pair of lists,\n */\npublic inline fun <T : Any> IntArray.partition(predicate: (Int) -> Boolean): Pair<List<Int>, List<Int>> {\n    val first = ArrayList<Int>()\n    val second = ArrayList<Int>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}

/**\n * Splits the original array into pair of lists,\n */\npublic inline fun <T : Any> LongArray.partition(predicate: (Long) -> Boolean): Pair<List<Long>, List<Long>> {\n    val first = ArrayList<Long>()\n    val second = ArrayList<Long>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}

/**\n * Splits the original array into pair of lists,\n */\npublic inline fun <T : Any> FloatArray.partition(predicate: (Float) -> Boolean): Pair<List<Float>, List<Float>> {\n    val first = ArrayList<Float>()\n    val second = ArrayList<Float>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}

/**\n * Splits the original array into pair of lists,\n */\npublic inline fun <T : Any> DoubleArray.partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>> {\n    val first = ArrayList<Double>()\n    val second = ArrayList<Double>()\n    for (element in this) {\n        if (predicate(element)) {\n            first.add(element)\n        } else {\n            second.add(element)\n        }\n    }\n    return Pair(first, second)\n}
Open Source Used In DNAC Platform Antelope

```kotlin
BooleanArray.partition(predicate: (Boolean) -> Boolean): Pair<List<Boolean>, List<Boolean>> {
    val first = ArrayList<Boolean>()
    val second = ArrayList<Boolean>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

CharArray.partition(predicate: (Char) -> Boolean): Pair<List<Char>, List<Char>> {
    val first = ArrayList<Character>()
    val second = ArrayList<Character>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}
```

- `List<Pair<Int, R>>` for Int
- `List<Pair<Double, R>>` for Double
- `List<Pair<Float, R>>` for Float
- `List<Pair<Short, R>>` for Short
- `List<Pair<Byte, R>>` for Byte
- `List<Pair<Char, R>>` for Char

These functions split an array into two lists based on a given predicate function. The first list contains elements for which the predicate returns `true`, and the second list contains elements for which the predicate returns `false`. The returned lists have the same index in the original array that was passed to the function.
list

\* Returns a list of values built from the elements of `this` array and the [other] array with the same index\n* using the provided [transform] function applied to each pair of elements.\n* The returned list has length of the shortest collection.\n* \n* @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
\*/
public inline fun <R, V> ByteArray.zip(other: Array<out R>, transform: (a: Byte, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

\*/
public inline fun <R, V> ShortArray.zip(other: Array<out R>, transform: (a: Short, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

\*/
public inline fun <R, V> IntArray.zip(other: Array<out R>, transform: (a: Int, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

\*/
public inline fun <R, V> LongArray.zip(other: Array<out R>, transform: (a: Long, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

\*/
public inline fun <R, V> FloatArray.zip(other: Array<out R>, transform: (a: Float, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

\*/
public inline fun <R, V> DoubleArray.zip(other: Array<out R>, transform: (a: Double, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

\*/
public inline fun <R, V> BooleanArray.zip(other: Array<out R>, transform: (a: Boolean, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

\*/
public inline fun <R, V> CharArray.zip(other: Array<out R>, transform: (a: Char, b: R) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
samples.collections.Iterables.Operations.zipIterable

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> Array<out T>.zip(other: Iterable<R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> ShortArray.zip(other: Iterable<R>): List<Pair<Short, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> IntArray.zip(other: Iterable<R>): List<Pair<Int, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> LongArray.zip(other: Iterable<R>): List<Pair<Long, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> FloatArray.zip(other: Iterable<R>): List<Pair<Float, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> DoubleArray.zip(other: Iterable<R>): List<Pair<Double, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> BooleanArray.zip(other: Iterable<R>): List<Pair<Boolean, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> CharArray.zip(other: Iterable<R>): List<Pair<Char, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <T, R, V> Array<out T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <R, V> ByteArray.zip(other: Iterable<R>, transform: (a: Byte, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <R, V> ShortArray.zip(other: Iterable<R>, transform: (a: Short, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <R, V> IntArray.zip(other: Iterable<R>, transform: (a: Int, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <R, V> FloatArray.zip(other: Iterable<R>, transform: (a: Float, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <R, V> DoubleArray.zip(other: Iterable<R>, transform: (a: Double, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <R, V> BooleanArray.zip(other: Iterable<R>, transform: (a: Boolean, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` collection and [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 */
public inline fun <R, V> CharArray.zip(other: Iterable<R>, transform: (a: Char, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}
list

* Returns a list of values built from the elements of `this` array and the [other] collection with the same index
* using the provided [transform] function applied to each pair of elements.

* The returned list has length of the shortest collection.

* @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform

* \/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform
samples.collections.Iterables.Operations.zipIterable

public infix fun IntArray.zip(other: IntArray):
    List<Pair<Int, Int>> {
        return zip(other) { t1, t2 -> t1 to t2 }
    }

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public infix fun LongArray.zip(other: LongArray):
    List<Pair<Long, Long>> {
        return zip(other) { t1, t2 -> t1 to t2 }
    }

/**
 * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index. The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public infix fun CharArray.zip(other: CharArray):
    List<Pair<Char, Char>> {
        return zip(other) { t1, t2 -> t1 to t2 }
    }

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest array.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public <V> ByteArray.zip(other: ByteArray, transform: (a: Byte, b: Byte) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest array.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public <V> ShortArray.zip(other: ShortArray, transform: (a: Short, b: Short) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest array.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public <V> IntArray.zip(other: IntArray, transform: (a: Int, b: Int) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest array.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public <V> LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index * using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest array.
 * @sample samples.collections.Iterables.Operations.zipIterableWithTransform
 */
public <V> FloatArray.zip(other: FloatArray, transform: (a: Float, b: Float) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
with the same index
* using the provided [transform] function applied to each pair of elements.
* The returned list has length of the shortest array.
* @sample

samples.collections.Iterables.Operations.zipIterableWithTransform\n */
public inline fun <V>
DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

DoubleArray.zip(other: DoubleArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

DoubleArray.zip(other: CharArray, transform: (a: Char, b: Char) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

ByteArray.zip(other: ByteArray, transform: (a: Byte, b: Byte) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

ShortArray.zip(other: ShortArray, transform: (a: Short, b: Short) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

ByteArray.zip(other: ByteArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

ShortArray.zip(other: ShortArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

ByteArray.zip(other: CharArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

ShortArray.zip(other: ShortArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}
def Appendable.joinTo(buffer: Appendable, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: (T) -> CharSequence? = null): Appendable {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

def List.joinTo(buffer: Appendable, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: (T) -> CharSequence? = null): Appendable {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}
/**
 * Appends the string from all the elements separated using [separator] and using the given 
 * [prefix] and [postfix] if supplied. If the collection could be huge, you can specify a non-negative value of 
 * [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which 
 * defaults to "..."), @sample samples.collections.Collections.Transformations.joinToString
 */
public fun <A : Appendable> CharArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "\", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((Char) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element)
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."), @sample samples.collections.Collections.Transformations.joinToString
 */
public fun <T> Array<out T>.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."), @sample samples.collections.Collections.Transformations.joinToString
 */
public fun ByteArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Byte) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."), @sample samples.collections.Collections.Transformations.joinToString
 */
public fun ShortArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Short) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."), @sample samples.collections.Collections.Transformations.joinToString
 */
public fun IntArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Int) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."), @sample samples.collections.Collections.Transformations.joinToString
 */
public fun LongArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Long) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}
samples.collections.Collections.Transformations.joinToString

    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}*{ *{ @sample samples.collections.Collections.Transformations.joinToString
}

    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

public fun BooleanArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Boolean) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

public fun CharArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Char) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

public fun <T> Array<out T>.asIterable(): Iterable<T> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

public fun ByteArray.asIterable(): Iterable<Byte> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

public fun ShortArray.asIterable(): Iterable<Short> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

public fun IntArray.asIterable(): Iterable<Int> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

public fun LongArray.asIterable(): Iterable<Long> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

public fun FloatArray.asIterable(): Iterable<Float> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

public fun DoubleArray.asIterable(): Iterable<Double> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

public fun BooleanArray.asIterable(): Iterable<Boolean> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

public fun CharArray.asIterable(): Iterable<Char> {
    if (isEmpty()) return emptyList()
    return this.iterator()
}

* {Open Source Used In DNAC Platform Antelope 9260}
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.

```kotlin
@sample samples.collections.Sequences.Building.sequenceFromArray
public fun ByteArray.asSequence():
    Sequence<Byte> {
        if (isEmpty()) return emptySequence()
        return Sequence { this.iterator() }
    }

@sample samples.collections.Sequences.Building.sequenceFromArray
public fun ShortArray.asSequence():
    Sequence<Short> {
        if (isEmpty()) return emptySequence()
        return Sequence { this.iterator() }
    }

@sample samples.collections.Sequences.Building.sequenceFromArray
public fun IntArray.asSequence():
    Sequence<Int> {
        if (isEmpty()) return emptySequence()
        return Sequence { this.iterator() }
    }

@sample samples.collections.Sequences.Building.sequenceFromArray
public fun LongArray.asSequence():
    Sequence<Long> {
        if (isEmpty()) return emptySequence()
        return Sequence { this.iterator() }
    }

@sample samples.collections.Sequences.Building.sequenceFromArray
public fun FloatArray.asSequence():
    Sequence<Float> {
        if (isEmpty()) return emptySequence()
        return Sequence { this.iterator() }
    }

@sample samples.collections.Sequences.Building.sequenceFromArray
public fun DoubleArray.asSequence():
    Sequence<Double> {
        if (isEmpty()) return emptySequence()
        return Sequence { this.iterator() }
    }

@sample samples.collections.Sequences.Building.sequenceFromArray
public fun BooleanArray.asSequence():
    Sequence<Boolean> {
        if (isEmpty()) return emptySequence()
        return Sequence { this.iterator() }
    }

@sample samples.collections.Sequences.Building.sequenceFromArray
public fun CharArray.asSequence():
    Sequence<Char> {
        if (isEmpty()) return emptySequence()
        return Sequence { this.iterator() }
    }

@kotlin.jvm.JvmName("averageOfByte")
public fun Array<out Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

@kotlin.jvm.JvmName("averageOfShort")
public fun Array<out Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

@kotlin.jvm.JvmName("averageOfInt")
public fun Array<out Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

@kotlin.jvm.JvmName("averageOfLong")
public fun Array<out Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

@kotlin.jvm.JvmName("averageOfFloat")
public fun Array<out Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

@kotlin.jvm.JvmName("averageOfDouble")
public fun Array<out Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}
```
elements in the array.
*/

public fun IntArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun LongArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun FloatArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
public fun DoubleArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun ByteArray.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun ShortArray.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun IntArray.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun LongArray.sum(): Long {
    var sum: Long = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun FloatArray.sum(): Float {
    var sum: Float = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun DoubleArray.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun IntArray.random(): Int {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun LongArray.random(): Long {
    return random(Random)
}

/**
 * Returns a random element from this range using the specified source of randomness.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun CharRange.random(): Char {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Random.random(): Random {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun MutableList.random(): MutableList {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun List.random(): List {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Set.random(): Set {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Array.random(): Array {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Array<out Any>.random(): Any {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Array<out Byte>.random(): Byte {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Array<out Short>.random(): Short {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Array<out Int>.random(): Int {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Array<out Long>.random(): Long {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Array<out Float>.random(): Float {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun Array<out Double>.random(): Double {
    return random(Random)
}
public fun IntRange.random(random: Random): Int {
    try {
        return random.nextInt(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the
 * specified source of randomness, or `null` if this range is empty.
 */
public fun LongRange.random(random: Random): Long {
    try {
        return random.nextLong(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the
 * specified source of randomness, or `null` if this range is empty.
 */
public fun CharRange.random(random: Random): Char {
    try {
        return random.nextInt(first.toInt(), last.toInt() + 1).toChar()
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range, or `null` if this
 * range is empty.
 */
public inline fun IntRange.randomOrNull(): Int? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this range, or `null` if this range is empty.
 */
public inline fun LongRange.randomOrNull(): Long? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this range, or `null` if this range is empty.
 */
public inline fun CharRange.randomOrNull(): Char? {
    return randomOrNull(Random)
}

/**
 * Returns `true` if this range contains the specified
 * [element].
 * Always returns `false` if the [element] is `null`.
 */
public operator fun IntRange.contains(element: Int?): Boolean {
    return element != null && contains(element)
}

/**
 * Returns `true` if this range contains
 * the specified [element].
 * Always returns `false` if the [element] is `null`.
 */
public operator fun LongRange.contains(element: Long?): Boolean {
    return element != null && contains(element)
}

/**
 * Returns `true` if this range contains
 * the specified [element].
 * Always returns `false` if the [element] is `null`.
 */
public operator fun CharRange.contains(element: Char?): Boolean {
    return element != null && contains(element)
}

/**
 * Checks if the specified [value] belongs to this range.
 */
public fun IntRange.contains(value: Byte): Boolean {
    return contains(value.toInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
public fun LongRange.contains(value: Long): Boolean {
    return contains(value.toLong())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
public fun CharRange.contains(value: Byte): Boolean {
    return contains(value.toChar())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
public fun ShortRange.contains(value: Byte): Boolean {
    return contains(value.toShort())
}
and floating point arguments has ambiguous semantics and is going to be removed.

```kotlin
@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4")
@kotlin.jvm.JvmName("doubleRangeContains")
public operator fun ClosedRange<Double>.contains(value: Byte): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.

@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4")
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Byte): Boolean {
    return contains(value.toFloat())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.

@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4")
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Double): Boolean {
    return value.toIntExactOrNull().let { if (it != null) contains(it) else false }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.

@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4")
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Double): Boolean {
    return value.toLongExactOrNull().let { if (it != null) contains(it) else false }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.

@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4")
@kotlin.jvm.JvmName("byteRangeContains")
public operator fun ClosedRange<Byte>.contains(value: Double): Boolean {
    return value.toByteExactOrNull().let { if (it != null) contains(it) else false }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.

@DeprecatedSinceKotlin(warningSince = "1.3", errorSince = "1.4")
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Double): Boolean {
    return value.toShortExactOrNull().let { if (it != null) contains(it) else false }
}
```

---

Open Source Used In DNAC Platform Antelope 9264
val kotlin.jvm.JvmName("shortRangeContains") operator fun ClosedRange<Short>.contains(value: Float): Boolean {
    return value.toFloatExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("doubleRangeContains") operator fun ClosedRange<Double>.contains(value: Float): Boolean {
    return value.toDoubleExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("longRangeContains") operator fun ClosedRange<Long>.contains(value: Short): Boolean {
    return value.toLongExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("byteRangeContains") operator fun ClosedRange<Byte>.contains(value: Int): Boolean {
    return value.toByteExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
    return value.toShortExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("doubleRangeContains") operator fun ClosedRange<Double>.contains(value: Int): Boolean {
    return value.toDoubleExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
    return value.toFloatExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("intRangeContains") operator fun ClosedRange<Int>.contains(value: Long): Boolean {
    return value.toLongExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("byteRangeContains") operator fun ClosedRange<Byte>.contains(value: Short): Boolean {
    return value.toByteExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
    return value.toShortExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("doubleRangeContains") operator fun ClosedRange<Double>.contains(value: Long): Boolean {
    return value.toDoubleExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
    return value.toFloatExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("intRangeContains") operator fun ClosedRange<Int>.contains(value: Short): Boolean {
    return value.toIntExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("byteRangeContains") operator fun ClosedRange<Byte>.contains(value: Short): Boolean {
    return value.toByteExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
    return value.toShortExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("doubleRangeContains") operator fun ClosedRange<Double>.contains(value: Long): Boolean {
    return value.toDoubleExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
    return value.toFloatExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("intRangeContains") operator fun ClosedRange<Int>.contains(value: Short): Boolean {
    return value.toIntExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}

/**
 * Checks if the specified [value] belongs to this range.
 */
val kotlin.jvm.JvmName("byteRangeContains") operator fun ClosedRange<Byte>.contains(value: Short): Boolean {
    return value.toByteExactOrNull().let {
        if (it != null) contains(it) else false
    }" + 
    if (value in startValue to endValue) {
        + "true"
    } else {
        + "false"
    }
}
public operator fun ClosedRange<Double>.contains(value: Short): Boolean {
    return contains(value.toDouble())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@Deprecated("This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.")
@DeprecatedSinceKotlin(warningSince = "1.3!", errorSince = "1.4")
public operator fun ClosedRange<Float>.contains(value: Short): Boolean {
    return contains(value.toFloat())
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this, to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Int): LongProgression {
    return LongProgression.fromClosedRange(this, to.toLong(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this, to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Int): LongProgression {
    return LongProgression.fromClosedRange(this, to.toLong(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Short.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Short.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Short.downTo(to: Long): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Char.downTo(to: Char): CharProgression {
    return CharProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Char.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Char.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Byte.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Byte.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Byte.downTo(to: Long): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Short.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Short.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Short.downTo(to: Long): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Char.downTo(to: Char): CharProgression {
    return CharProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Char.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Char.downTo(to: Long): IntProgression {
    return IntProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Int): LongProgression {
    return LongProgression.fromClosedRange(this, to.toLong(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to ‘this’ value.
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
public infix fun Int.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1)
}
than or equal to `this` value.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.

* If the [to] value is greater than `this` value the returned progression is empty.
Int.toShortExactOrNull(): Short? {
    return if (this in Short.MIN_VALUE.toInt()..Short.MAX_VALUE.toInt())
        this.toInt() else null
}

internal fun Long.toShortExactOrNull(): Short? {
    return if (this in
        Short.MIN_VALUE.toLong()..Short.MAX_VALUE.toLong()) this.toInt() else null
}

internal fun Double.toShortExactOrNull(): Short? {
    return if (this in
        Short.MIN_VALUE.toDouble()..Short.MAX_VALUE.toDouble()) this.toInt().toShort() else null
}

internal fun Float.toShortExactOrNull(): Short? {
    return if (this in
        Short.MIN_VALUE.toFloat()..Short.MAX_VALUE.toFloat()) this.toInt().toShort() else null
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Int.until(to: Byte): IntRange {
    return this .. (to.toInt() - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Long.until(to: Byte): LongRange {
    return this .. (to.toLong() - 1).toLong()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Byte.until(to: Byte): IntRange {
    return this.toInt() .. (to.toInt() - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Short.until(to: Byte): IntRange {
    return this.toInt() .. (to.toInt() - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Char.until(to: Char): CharRange {
    if (to <= '\u0000')
        return CharRange.EMPTY
    return this .. (to - 1).toChar()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Int.until(to: Int): IntRange {
    if (to <= Int.MIN_VALUE)
        return IntRange.EMPTY
    return this .. (to - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Long.until(to: Int): LongRange {
    if (to <= Long.MIN_VALUE)
        return LongRange.EMPTY
    return this .. (to - 1).toLong()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Byte.until(to: Int): IntRange {
    if (to <= Int.MIN_VALUE)
        return IntRange.EMPTY
    return this.toInt() .. (to - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Short.until(to: Int): IntRange {
    if (to <= Int.MIN_VALUE)
        return IntRange.EMPTY
    return this.toInt() .. (to - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Int.until(to: Long): LongRange {
    if (to <= Long.MIN_VALUE)
        return LongRange.EMPTY
    return this .. (to - 1).toLong()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Long.until(to: Long): LongRange {
    if (to <= Long.MIN_VALUE)
        return LongRange.EMPTY
    return this .. (to - 1).toLong()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Byte.until(to: Long): LongRange {
    if (to <= Long.MIN_VALUE)
        return LongRange.EMPTY
    return this .. (to - 1).toLong()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Short.until(to: Long): LongRange {
    if (to <= Long.MIN_VALUE)
        return LongRange.EMPTY
    return this .. (to - 1).toLong()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Int.until(to: Short): IntRange {
    return this .. (to.toInt() - 1).toInt()
}

/**
 * Returns a range from this value up to but excluding the specified `to` value.
 * If the `to` value is less than or equal to `this` value, then the returned range is empty.
 */
public infix fun Long.until(to: Short): LongRange {
    return this .. (to.toLong() - 1).toLong()
1).toLong() \n
\n
Returns a range from this value up to but excluding the specified [to] value :n * \n
* If the [to] value is less than or equal to ‘this’ value, then the returned range is empty: \n
* Anpublic infix fun Byte.until(\n
Short): IntRange \n
\n
* \n
* Returns a range from this value up to but excluding the specified [to] value :n * \n
* If the [to] value is less than or equal to ‘this’ value, then the returned range is empty: \n
* Anpublic infix fun Short.until(\n
Short): IntRange \n
\n
* \n
* Ensures that this value is not less than the specified [minimumValue]:n * \n
* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtLeastComparable \n
T.coerceAtLeast(minimumValue: T): T \n
\n
* \n
* Ensures that this value is not less than the specified [minimumValue]:n * \n
* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtLeast

Byte \n
\n
* \n
* Ensures that this value is not less than the specified [minimumValue]:n * \n
* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtLeast

Long \n
\n
* \n
* Ensures that this value is not less than the specified [minimumValue]:n * \n
* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtLeast

Float \n
\n
* \n
* Ensures that this value is not less than the specified [minimumValue]:n * \n
* @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtLeast

Double \n
\n
* \n
* Ensures that this value is not greater than the specified [maximumValue]:n * \n
* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtMost

Byte \n
\n
* \n
* Ensures that this value is not greater than the specified [maximumValue]:n * \n
* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtMost

Short \n
\n
* \n
* Ensures that this value is not greater than the specified [maximumValue]:n * \n
* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtMost

\n
\n
\n
\n
* Anpublic infix fun Byte.coerceAtMost(maximumValue: Byte): Byte \n
\n
* \n
* Ensures that this value is not greater than the specified [maximumValue]:n * \n
* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtMost

Short.coerceAtMost(maximumValue: Short): Short \n
\n
* \n
* Ensures that this value is not greater than the specified [maximumValue]:n * \n
* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtMost

Float.coerceAtMost(maximumValue: Float): Float \n
\n
* \n
* Ensures that this value is not greater than the specified [maximumValue]:n * \n
* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtMost

Double.coerceAtMost(maximumValue: Double): Double \n
\n
* \n
* Ensures that this value is not greater than the specified [maximumValue]:n * \n
* @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise:n * \n
* @sample samples.comparisons.ComparableOps.coerceAtMost
the [maximumValue] otherwise.\n* \n* @sample samples.comparisons.ComparableOps.coerceAtMost\n
*\npublic fun Long.coerceAtMost(maximumValue: Long): Long {\n    return if (this > maximumValue) maximumValue else this;\n}\n*\nEnsures that this value is not greater than the specified [maximumValue].\n* \n* @return this value if it’s less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n* \n* @sample samples.comparisons.ComparableOps.coerceAtMost\n
Float.coerceAtMost(maximumValue: Float): Float {\n    return if (this > maximumValue) maximumValue else this;\n}\n*\nEnsures that this value lies in the specified range [minimumValue].\n* \n* @return this value if it’s in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceAtMost\n
T.coerceIn(minimumValue: T?, maximumValue: T?): T {\n    if (minimumValue !== null && maximumValue !== null) {\n        if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n        if (this < minimumValue) return minimumValue\n        if (this > maximumValue) return maximumValue\n    } else {\n        if (minimumValue !== null && this < minimumValue) return minimumValue\n        if (maximumValue !== null && this > maximumValue) return maximumValue\n    }\n    return this\n}\n*\nEnsures that this value lies in the specified range [minimumValue].\n* \n* @return this value if it’s in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceIn\n
Byte.coerceIn(minimumValue: Byte, maximumValue: Byte): Byte {\n    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this\n}\n*\nEnsures that this value lies in the specified range [minimumValue].\n* \n* @return this value if it’s in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceIn\n
Short.coerceIn(minimumValue: Short, maximumValue: Short): Short {\n    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this\n}\n*\nEnsures that this value lies in the specified range [minimumValue].\n* \n* @return this value if it’s in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceIn\n
Int.coerceIn(minimumValue: Int, maximumValue: Int): Int {\n    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this\n}\n*\nEnsures that this value lies in the specified range [minimumValue].\n* \n* @return this value if it’s in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n* \n* @sample samples.comparisons.ComparableOps.coerceIn\n
Long.coerceIn(minimumValue: Long, maximumValue: Long): Long {\n    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n    if (this < minimumValue) return minimumValue\n    if (this > maximumValue) return maximumValue\n    return this\n}\n*\nEnsures that this value lies in the specified range [minimumValue].\n* \n* @return this value if it’s in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].

is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].

if (this < minimumValue) return minimumValue

if (this > maximumValue) return maximumValue

return this

* Ensures that this value lies in the specified range [minimumValue,..maximumValue]

* If [maximumValue] is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue]

* @sample samples.comparisons.ComparableOps.coerceIn

public fun Float.coerceIn(minimumValue: Float, maximumValue: Float): Float {
    if (minimumValue > maximumValue) throw
    IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
    minimum $minimumValue.")

    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue

    return this

* Ensures that this value lies in the specified range [minimumValue,..maximumValue]

* If [maximumValue] is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue]

* @sample samples.comparisons.ComparableOps.coerceIn

public fun Double.coerceIn(minimumValue: Double, maximumValue: Double): Double {
    if (minimumValue > maximumValue) throw
    IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than
    minimum $minimumValue.")

    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue

    return this

* Ensures that this value lies in the specified range [range]

* If this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`

* @sample samples.comparisons.ComparableOps.coerceInFloatingPointRange

public fun <T : Comparable<T>> T.coerceIn(range: ClosedFloatingPointRange<T>): T {
    if (range.isEmpty()) throw
    IllegalArgumentException("Cannot coerce value to an empty range: $range.")

    return when {
        range.lessThanOrEquals(this, range.start) && !range.lessThanOrEquals(range.start, this) -> range.start
        range.endInclusive && !range.lessThanOrEquals(this, range.endInclusive) -> range.endInclusive
        else -> this
    }

* Ensures that this value lies in the specified range [range]

* If this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`

* @sample samples.comparisons.ComparableOps.coerceInComparable

public fun <T : Comparable<T>> T.coerceIn(range: ClosedRange<T>): T {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<T>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")

    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }

* Ensures that this value lies in the specified range [range]

* If this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`

* @sample samples.comparisons.ComparableOps.coerceInComparable

public fun Int.coerceIn(range: ClosedRange<Int>): Int {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<Int>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")

    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }

* Ensures that this value lies in the specified range [range]

* If this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`

* @sample samples.comparisons.ComparableOps.coerceInComparable

public fun Long.coerceIn(range: ClosedRange<Long>): Long {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<Long>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")

    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else -> this
    }

* Ensures that this value lies in the specified range [range]

* If this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`

* @sample samples.comparisons.ComparableOps.coerceInComparable

* A constant holding the minimum value an instance of UByte can have.

@PublishedApi
@ExperimentalUnsignedTypes
@sinceKotlin("1.3")
public inline class UByte @PublishedApi internal constructor(@PublishedApi internal val data: Byte) : Comparable<UByte> {

    companion object {
        /**
         * A constant holding the minimum value an instance of UByte can have.
         */
        public const val
MIN_VALUE: UByte = UByte(0)  /** A constant holding the maximum value an instance of UByte can have. */

public const val MAX_VALUE: UByte = UByte(-1)  /** The number of bytes used to represent an instance of UByte in a binary form. */

public const val SIZE_BITS: Int = 8  /** The number of bits used to represent an instance of UByte in a binary form. */

public const val SIZE_BYTES: Int = 1  /** The number of bytes used to represent an instance of UByte in a binary form. */

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun minus(other: UShort): UInt = this.toUInt().minus(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun times(other: UShort): UInt = this.toUInt().times(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun times(other: ULong): ULong = this.toULong().times(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun div(other: UInt): UInt = this.toUInt().div(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun div(other: ULong): ULong = this.toULong().div(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other)

@Suppress("OVERRIDE_BY_INLINE")
public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)
** Inverts the bits in this value. */
@kotlin.internal.InlineOnly
public inline operator fun xor(other: UByte): UByte = UByte(this.data xor other.data)
/** Performs a bitwise XOR operation between the two values. */
@kotlin.internal.InlineOnly
public inline infix fun xor(other: UByte): UByte = UByte(this.data xor other.data)

    /** Performs a bitwise AND operation between the two values. */
    @kotlin.internal.InlineOnly
    public inline fun and(other: UByte): UByte = UByte(this.data and other.data)
/** Performs a bitwise OR operation between the two values. */
@kotlin.internal.InlineOnly
public inline infix fun or(other: UByte): UByte = UByte(this.data or other.data)

    /** Performs a bitwise OR operation between the two values. */
    @kotlin.internal.InlineOnly
    public inline fun or(other: UByte): UByte = UByte(this.data or other.data)
/** Performs a bitwise XOR operation between the two values. */
@kotlin.internal.InlineOnly
public inline function inv(): UByte = UByte(data.inv())

    /** Inverts the bits in this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline function rem(other: ULong): ULong = this.toULong().rem(other)
/** Increments this value. */
@kotlin.internal.InlineOnly
public inline function inc(): UByte = UByte(data.inc())
/** Decrements this value. */
@kotlin.internal.InlineOnly
public inline function dec(): UByte = UByte(data.dec())

    /** Creates a range from this value to the specified [other] value. */
    @kotlin.internal.InlineOnly
    public inline function rangeTo(other: UByte): UIntRange = UIntRange(this.toUInt(), other.toUInt())
/** Converts this [UByte] value to [Byte]. */
@kotlin.internal.InlineOnly
public inline function toByte(): Byte = data
/** Converts this [UByte] value to [Short]. */
@kotlin.internal.InlineOnly
public inline function toShort(): Short = data.toShort()
/** Converts this [UByte] value to [Int]. */
@kotlin.internal.InlineOnly
public inline function toInt(): Int = data.toInt() and 0xFF
/** Converts this [UByte] value to [UInt]. */
@kotlin.internal.InlineOnly
public inline function toUInt(): UInt = UInt(data.toInt() and 0xFF)
/** Converts this [UByte] value to [UByte]. Otherwise the result is negative. */
@kotlin.internal.InlineOnly
public inline function toUByte(): UByte = UByte(data.inv())
/** The least significant 8 bits of the resulting `Short` value are the same as the bits of this `UByte` value, whereas the most significant 8 bits are filled with zeros. */
@kotlin.internal.InlineOnly
public inline function toUShort(): UShort = UShort(data.toShort() and 0xFF)
/** The least significant 8 bits of the resulting `Int` value are the same as the bits of this `UByte` value, whereas the most significant 24 bits are filled with zeros. */
@kotlin.internal.InlineOnly
public inline function toLong(): Long = data.toLong() and 0xFF
/** The least significant 8 bits of the resulting `Bytes` are the same as the bits of this `UByte` value. */
@kotlin.internal.InlineOnly
public inline function toBytes(): Bytes = UBytes(data.toByteArray().toByteArray())
/** The least significant 8 bits of the resulting `Bytes` are the same as the bits of this `UByte` value. */
@kotlin.internal.InlineOnly
public inline function toULong(): ULong = data.toLong() and 0xFF
/** The least significant 8 bits of the resulting `Byte` value has the same binary representation as this `UByte`. Otherwise the result is negative. */
@kotlin.internal.InlineOnly
public inline function toByte(): Byte = UByte(data)
binary representation as this `Byte` value:

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Byte.toUByte(): UByte = UByte(this.toByte())
```

* If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `Short`. The resulting `UByte` value is represented by the least significant 8 bits of this `Short` value.

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Short.toUByte(): UByte = UByte(this.toByte())
```

* If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `Int` value. The resulting `UByte` value is represented by the least significant 8 bits of this `Int` value.

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Int.toUByte(): UByte = UByte(this.toByte())
```

* If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents the same numerical value as this `Long` value. The resulting `UByte` value is represented by the least significant 8 bits of this `Long` value.

```
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Long.toUByte(): UByte = UByte(this.toByte())
```

---

Open Source Used In DNAC Platform Antelope  9274
this.minus(other.toUInt())
    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(minus: UShort): UInt = this.minus(other.toUInt())
    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(minus: UInt): UInt = this.minus(other.toUInt())
    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(minus: ULong): ULong =

    this.times(other.toUInt())
    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(times: UByte): UInt = this.times(other.toUInt())
    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(times: UShort): UInt = this.times(other.toUInt())
    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(times: UInt): UInt = this.times(other.toUInt())
    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(times: ULong): ULong =

    this.div(other.toUInt())
    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(div: UByte): UInt = this.div(other.toUInt())
    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(div: UShort): UInt = this.div(other.toUInt())
    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(div: UInt): UInt = this.div(other.toUInt())
    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(div: ULong): ULong =

    this.rem(other.toUInt())
    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(rem: UByte): UInt = this.rem(other.toUInt())
    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(rem: UShort): UInt = this.rem(other.toUInt())
    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(rem: UInt): UInt =
    /** Calculates the remainder of dividing this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(rem: ULong): ULong =

    this.inc()
    /** Increments this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun inc(): UInt =

    this.dec()
    /** Decrements this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun dec(): UInt =

    this.rangeTo(other)
    /** Creates a range from this value to the specified [other] value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rangeTo(rangeTo: UInt): UIntRange =

    this.shl(bitCount)
    /** Shifts this value left by the [bitCount] number of bits. */
    @kotlin.internal.InlineOnly
    public inline infix fun shl(shl: Int): UInt =

    this.shr(bitCount)
    /** Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros. */
    @kotlin.internal.InlineOnly
    public inline infix fun shr(shr: Int): UInt =

    this.and(other)
    /** Performs a bitwise AND operation between the two values. */
    @kotlin.internal.InlineOnly
    public inline infix fun and(and: UInt): UInt =

    this.or(other)
    /** Performs a bitwise OR operation between the two values. */
    @kotlin.internal.InlineOnly
    public inline infix fun or(or: UInt): UInt =

    this.xor(other)
    /** Performs a bitwise XOR operation between the two values. */
    @kotlin.internal.InlineOnly
    public inline infix fun xor(xor: UInt): UInt =

    this.inv()
    /** Inverts the bits in this value. */
    @kotlin.internal.InlineOnly
    public inline fun inv(): UInt =

    this.toByte()
    /** Converts this [UInt] value to [Byte]. */
    @kotlin.internal.InlineOnly
    public inline fun toByte(): Byte =

    this.toShort()
    /** Converts this [UInt] value to [Short]. */
    @kotlin.internal.InlineOnly
    public inline fun toShort(): Short =

    this.toInt()
    /** Converts this [UInt] value to [Int]. */
    @kotlin.internal.InlineOnly
    public inline fun toInt(): Int =
*\n * The resulting `Int` value has the same binary representation as this `UInt` value.
 *\n @kotlin.internal.InlineOnly\n public inline fun toInt(): Int = data\n /**\n * Converts this `UInt` value to `Int`.
 *
 * The least significant 32 bits of the resulting `Long` value are the same as the bits of this `UInt` value,
 * whereas the most significant 32 bits are filled with zeros.
 */
 @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and 0xFFFF_FFFF\n /**\n * Converts this `UInt` value to `Long`.
 *
 * The resulting `Long` value represents the same numerical value as this `UInt` value.
 *\n * If this value is less than or equals to `UInt.MAX_VALUE`, the resulting `Long` value represents
 * the same numerical value as this `UInt` value.
 *\n * The resulting `Double` value is represented by the least significant 8 bits of this `UInt` value.
 */
 @kotlin.internal.InlineOnly\n public inline fun toFloat(): Float = this.toDouble().toFloat()\n /**\n * Converts this `UInt` value to `Double`.
 *
 * The resulting `Double` value represents the same numerical value as this `UInt` value.
 */
 @kotlin.internal.InlineOnly\n public inline fun toDouble(): Double = uintToDouble(data)\n
public override fun toString(): String = toLong().toString()
A constant holding the minimum value an instance of UShort can have.

A constant holding the maximum value an instance of UShort can have.

The number of bytes used to represent an instance of UShort in a binary form.

The number of bits used to represent an instance of UShort in a binary form.

Compares this value with the specified value for order.

Compares this value with the specified value for order.

Compares this value with the specified value for order.

Compares this value with the specified value for order.

Compares this value with the specified value for order.

Add the other value to this value.

Add the other value to this value.

Add the other value to this value.

Add the other value to this value.

Subtract the other value from this value.

Subtract the other value from this value.

Subtract the other value from this value.

Subtract the other value from this value.

Multiply this value by the other value.

Multiply this value by the other value.

Multiply this value by the other value.

Multiply this value by the other value.

Divide this value by the other value.

Divide this value by the other value.

Divide this value by the other value.

Divide this value by the other value.

Divide this value by the other value.
public inline operator fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())
/** Divides this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): UInt = this.toUInt().div(other)  // Divides this value by the other value.
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())
/** Divides this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UInt = this.toUInt().div(other)
/** Divides this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UByte): UByte = this.toUByte().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): UByte = this.toUShort().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): UByte = this.toUInt().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): UByte = this.toULong().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rangeTo(other: UShort): UIntRange = UIntRange(this.toUInt(), other.toUInt())
/** Creates a range from this value to the specified [other] value. */
@kotlin.internal.InlineOnly
public inline operator fun rangeTo(other: UByte): UByte = data.toUByte()  // Inverts the bits in this value.
@kotlin.internal.InlineOnly
public inline operator fun inv(): UShort = UShort(data.inv())
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun xor(other: UShort): UShort = UShort(this.data xor other.data)
/** Performs a bitwise XOR operation between the two values. */
@kotlin.internal.InlineOnly
public inline operator fun or(other: UShort): UShort = UShort(this.data or other.data)
/** Performs a bitwise OR operation between the two values. */
@kotlin.internal.InlineOnly
public inline operator fun and(other: UShort): UShort = UShort(this.data and other.data)
/** Performs a bitwise AND operation between the two values. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rangeTo(other: UShort): UShort = UShort(data.toUByte())
/** Returns this value. */
@kotlin.internal.InlineOnly
public inline operator fun inc(): UShort = UShort(data.inc())
/** Increments this value. */
@kotlin.internal.InlineOnly
public inline operator fun dec(): UShort = UShort(data.dec())
/** Decrements this value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): ULong = this.toULong().div(other)
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UInt = this.toUInt().div(other)
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UInt = this.toUInt().div(other)
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UShort): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UInt): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): UByte = data.toUByte()
/** Calculates the value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: UByte): UByte = data.toUByte()
The result of the `Float` value represents the same numerical value as this `UShort`. The most significant 48 bits are filled with zeros.

```kotlin
@kotlin.internal.InlineOnly
public inline fun toULong():
    ULong = ULong(data.toLong() and 0xFFFF)
```

```kotlin
/**
 * Converts this `UShort` value to `Float`. The resulting `Float` value represents the same numerical value as this `UShort`. The least significant 8 bits of the resulting `UShort` value are the same as the bits of this `Byte` value.
 */
@kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toInt().toFloat()
```

```kotlin
/**
 * Converts this `UShort` value to `Double`. The resulting `Double` value represents the same numerical value as this `UShort`. The resulting `UShort` value represents the same binary representation as this `UShort`. If this value is positive, the resulting `UShort` value represents the same numerical value as this `Int`. The resulting `UShort` value is represented by the least significant 16 bits of this `Int` value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double =
    this.toInt().toDouble()
```

```kotlin
/**
 * Converts this `Byte` value to `UShort`.
 *
 * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this `Byte` value. whereas the most significant 8 bits are filled with the sign bit of this value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Byte.toUShort(): UShort = UShort(this.toShort())
```

```kotlin
/**
 * Converts this `Short` value to `UShort`.
 *
 * The resulting `UShort` value has the same binary representation as this `Short` value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Short.toUShort(): UShort = UShort(this)
```

```kotlin
/**
 * Converts this `Int` value to `UShort`.
 *
 * If this value is positive and less than or equals to `UShort.MAX_VALUE`, the resulting `UShort` value represents the same numerical value as this `Int`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Int.toUShort(): UShort = UShort(this.toShort())
```

```kotlin
/**
 * Converts this `Long` value to `UShort`.
 *
 * If this value is positive and less than or equals to `UShort.MAX_VALUE`, the resulting `UShort` value represents the same numerical value as this `Long`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Long.toUShort(): UShort = UShort(this.toShort())
```
Open Source Used In DNAC Platform Antelope 9280

override fun toString(): String = "$first..$last"

    override fun toString(): String = "$first..$last"

    companion object {
        /** An empty range of values of type Long. */
        public val EMPTY: LongRange = LongRange(1, 0)
    }

/**
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CollectionsKt")
@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)

package kotlin.collections

import kotlin.contracts.*
import kotlin.random.Random

internal object EmptyIterator : ListIterator<Nothing> {
    override fun hasNext(): Boolean = false
    override fun hasPrevious(): Boolean = false
    override fun nextIndex(): Int = 0
    override fun previousIndex(): Int = -1
    override fun next(): Nothing = throw NoSuchElementException()
    override fun previous(): Nothing = throw NoSuchElementException()
}

internal object EmptyList : List<Nothing>, Serializable, RandomAccess {
    private const val serialVersionUID: Long = -7390468764508069838L

    override fun equals(other: Any?): Boolean = other is List<*>
    override fun hashCode(): Int = 1
    override fun toString(): String = "[]"

    override val size: Int get() = 0
    override fun isEmpty(): Boolean = true
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun get(index: Int): Nothing = throw IndexOutOfBoundsException("Empty list doesn't contain element at index $index.")
    override fun indexOf(element: Nothing): Int = -1
    override fun lastIndexOf(element: Nothing): Int = -1

    override fun iterator(): Iterator<Nothing> = EmptyIterator
    override fun listIterator(): ListIterator<Nothing> = EmptyIterator
    override fun listIterator(index: Int): ListIterator<Nothing> {
        if (index != 0) throw IndexOutOfBoundsException("Index: $index")
        return EmptyIterator
    }

    override fun subList(fromIndex: Int, toIndex: Int): List<Nothing> {
        if (fromIndex == 0 && toIndex == 0) return this
        throw IndexOutOfBoundsException("fromIndex: $fromIndex, toIndex: $toIndex")
    }

    private fun readResolve(): Any = EmptyList
}

/**
 * Returns an empty read-only list.  The returned list is serializable (JVM).
 * @sample samples.collections.Collections.Lists.emptyReadOnlyList
 */
public fun <T> emptyList(): List<T> = EmptyList

/**
 * Returns a new read-only list of given elements.  The returned list is serializable (JVM).
 * @sample samples.collections.Collections.Lists.readOnlyList
 */
public fun <T> listOf(vararg elements: T): List<T> = if (elements.size > 0) elements.asList() else emptyList()
public fun <T : Any> listOfNotNull(element: T?): List<T> = if (element != null) listOf(element) else emptyList()

/**
 * Returns a new read-only list only of those given elements, that are not null. The returned list is serializable (JVM).
 * @sample samples.collections.Collections.Lists.listOfNotNull
 */
public fun <T : Any> listOfNotNull(vararg elements: T?): List<T> = elements.filterNotNull()

/**
 * Creates a new read-only list with the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * The function [init] is called for each list element sequentially starting from the first one.
 * It should return the value for a list element given its index.
 * @sample samples.collections.Collections.Lists.readOnlyListFromInitializer
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public fun <T> List(size: Int, init: (index: Int) -> T): List<T> = MutableList(size, init)

/**
 * Creates a new mutable list with the specified [size], where each element is calculated by calling the specified
 * [init] function.
 * @sample samples.collections.Collections.Lists.mutableListFromInitializer
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <T> MutableList(size: Int, init: (index: Int) -> T): MutableList<T> {
    val list = ArrayList<T>(size)
    repeat(size) { index -> list.add(init(index)) }
    return list
}

/**
 * Builds a new read-only [List] by populating a [MutableList] using the given [builderAction] and returning a read-only
 * list with the same elements. The list passed as a receiver to the [builderAction] is valid only inside that function.
 * Using it outside of the function produces an unspecified behavior.
 * @sample samples.collections.Builders.Lists.buildListSample
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <E> buildList(@BuilderInference builderAction: MutableList<E>.() -> Unit): List<E> {
    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }
    return buildListInternal(builderAction)
}

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal expect inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit): List<E>

/**
 * Returns an [IntRange] of the valid indices for this collection.
 * @sample samples.collections.Collections.Collections.indicesOfCollection
 */
public val Collection<*>.indices: IntRange
    get() = 0..size - 1

/**
 * Returns the index of the last item in the list or -1 if the list is empty.
 * @sample samples.collections.Collections.Lists.lastIndexOfList
 */
public val <T> List<T>.lastIndex: Int
    get() = this.size - 1

/**
 * Returns `true` if this nullable collection is either null or empty.
 * @sample samples.collections.Collections.Collections.collectionIsNullOrEmpty
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Collection<T>?.isNullOrEmpty(): Boolean {
    contract {
        returns(false) implies (this@isNullOrEmpty != null)
    }

    return this == null || this.isEmpty()
}

/**
 * Returns this Collection if it's not `null` and the empty list otherwise.
 * @sample samples.collections.Collections.Collections.collectionOrEmpty
 */
@kotlin.internal.InlineOnly
public inline fun <T> Collection<T>?.orEmpty(): Collection<T> = this ?: emptyList()
the empty list otherwise.

    *\@sample samples.collections.Collections.Lists.listOrEmpty

*\@kotlin.internal.InlineOnly\public inline fun \T: List<\T>.orElseempty(): List<\T> = this ?: emptyList()
*\@sample samples.collections.Collections.Collections.emptyList

*\@kotlin.SinceKotlin("1.3")\@kotlin.internal.InlineOnly\public inline fun \C<, \R>: C.ifEmpty(\defaultValue: () -> \R): \R where \C : Collection<*, \R> : \R if (is(Empty()) default(\defaultValue) else this)\@sample samples.collections.Collections.Collections.emptyList

*\@kotlin.SinceKotlin("1.3")\@kotlin.internal.InlineOnly\public inline fun \T: List<\T>.orEmpty(): List<\T> = this ?: emptyList()

*\@kotlin.internal.InlineOnly
public inline fun <\T> List<\T>?.orEmpty(): List<\T> = this ?: emptyList()

/**
 * Returns this collection if it's not empty or the result of calling \[defaultValue\] function if the collection is empty.
 * @sample samples.collections.Collections.Collections.emptyList
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Collection<*, R> :
    if (is(Empty()) default(\defaultValue) else this)\@sample samples.collections.Collections.Collections.emptyList

/**
 * Checks if all elements in the specified collection are contained in this collection.\n * Allows to overcome type-safety restriction of \`containsAll\` that requires to pass a collection of type \`Collection<\E>\`.
 * @sample samples.collections.Collections.Collections.emptyList
 */
@Restricts<EXTENSION_SHADOWED_BY_MEMBER>() // false warning, extension takes precedence in some cases\@kotlin.internal.InlineOnly\public inline fun \<\@kotlin.internal.OnlyInputTypes T> Collection<\T>.containsAll\(elements: Collection<\T>: Boolean = this.containsAll\(elements)\)\@sample samples.collections.Collections.Collections.emptyList

/**
 * Returns a new list with the elements of this list randomly shuffled using the specified \[random\] instance as the source of randomness.\n *\@sample samples.collections.Collections.Collections.emptyList
 */
@SinceKotlin("1.3")
public fun <\T> Iterable<\T>.shuffled\(\random: Random\): List<\T> =
toMutableList().apply { shuffle\(\random\) }\@sample samples.collections.Collections.Collections.emptyList

/**
 * Searches this list or its range for the provided \[element\] using the binary search algorithm.\n * The list is expected to be sorted into ascending order according to the Comparable natural ordering of its elements,\n * otherwise the result is undefined.\n * If the list contains multiple elements equal to the specified \[element\], there is no guarantee which one will be found.\n * `null` value is considered to be less than any non-null value.\n *\@return the index of the element, if it is contained in the list within the specified range,\n * otherwise, the inverted insertion point `(insertion point - 1)`.\n * The insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified subrange of list) still remains sorted.\n * @sample samples.collections.Collections.Collections.emptyList
 */
public fun <\T : Comparable<\T>> List<\T?>.binarySearch\(element: \T?, fromIndex: Int = 0, toIndex: Int = size\): Int {
    rangeCheck\(size, fromIndex, toIndex\)
    var low = fromIndex
    var high = toIndex - 1\n    while (low <= high) {
        val mid = (low + high).ushr(1) // safe from overflows\n        val midVal = get\(mid\)
        val cmp = compareValues\(midVal, element\)

        if (cmp < 0) low = mid + 1
        else if (cmp > 0) high = mid - 1
        else return mid // key found
    }
    return -(low + 1) // key not found\@sample samples.collections.Collections.Collections.emptyList

/**
 * Searches this list or its range for the provided \[element\] using the binary search algorithm.\n * The list is expected to be sorted into ascending order according to the specified \[comparator\],\n * otherwise the result is undefined.\n * If the list contains multiple elements equal to the specified \[element\], there is no guarantee which one will be found.\n * `null` value is considered to be less than any non-null value.\n *\@return the index of the element, if it is contained in the list within the specified range,\n * otherwise, the inverted insertion point `(insertion point - 1)`.\n * The insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified subrange of list) still remains sorted according to the specified \[comparator\].\n * @sample samples.collections.Collections.Collections.emptyList
 */
public fun <\T> List<\T>.binarySearch\(element: \T, \comparator: Comparator<\in T>, fromIndex: Int = 0, toIndex: Int = size\): Int {
    rangeCheck\(size, fromIndex, toIndex\)
    var low = fromIndex
    var high = toIndex - 1\n    while (low <= high) {
        val mid = (low + high).ushr(1) // safe from overflows\n        val midVal = get\(mid\)
        val cmp = \comparator\(compare\(midVal, element\)\)

        if (cmp < 0) low = mid + 1
        else if (cmp > 0) high = mid - 1
        else return mid // key found
    }
    return -(low + 1) // key not found\@sample samples.collections.Collections.Collections.emptyList

/**
 * Searches this list or its range for an element having the key returned by the specified \[selector\] function equal to the provided \[key\] using the binary search algorithm.\n * The list is expected to be sorted into ascending order according to the Comparable natural ordering of keys of its elements,\n * otherwise the result is undefined.\n * If the list contains multiple elements with the specified \[key\], there is no guarantee which one will be found.\n * `null` value is considered to be less than any non-null value.\n *\@return the index of the element with the
 */
specified [key], if it is contained in the list within the specified range; * otherwise, the inverted insertion point `(-insertion point - 1)`.

* The insertion point is defined as the index at which the element should be inserted; * otherwise, the inverted insertion point `(-insertion point - 1)`.

* The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted.

* @sample samples.collections.Collections.Lists.binarySearchBy

```kotlin
List<T>.binarySearchBy

```
V> emptyMap(): Map<K, V> = @Suppress("UNCHECKED_CAST") (EmptyMap as Map<K, V>)

Returns a new read-only map with the specified contents, given as a list of pairs. If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs. Entries of the map are iterated in the order they were specified. The returned map is serializable (JVM).

V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> = @sample samples.collections.Maps.Instantiation.mapFromPairs

V> mapOf(): Map<K, V> = emptyMap() Returns an empty read-only map.

V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> = mapFromPairs(pairs).apply { putAll(pairs) }

V> mapOf(): Map<K, V> = emptyMap() Returns an empty new [MutableMap] with the specified contents, given as a list of pairs where the first component is the key and the second is the value. Entries of the map are iterated in the order they were specified. Entries of the created map are added in the order they were specified. The returned map is serializable (JVM).

V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> = LinkedHashMap(mapCapacity(pairs.size)).apply { putAll(pairs) }

V> mapOf(): Map<K, V> = emptyMap() Returns an empty new [HashMap] with the specified contents, given as a list of pairs where the first component is the key and the second is the value. Entries of the map are iterated in the order they were specified. Entries of the created map are added in the order they were specified. The returned map is serializable (JVM).

V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> = LinkedHashMap(mapCapacity(pairs.size)).apply { putAll(pairs) }

V> mapOf(): Map<K, V> = emptyMap() Returns an empty new [LinkedHashMap] with the specified contents, given as a list of pairs where the first component is the key and the second is the value. Entries of the map are iterated in the order they were specified. Entries of the created map are added in the order they were specified. The returned map is serializable (JVM).

V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> = LinkedHashMap(mapCapacity(pairs.size)).apply { putAll(pairs) }

V> buildMap(@BuilderInference builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {... contract { callInPlace(builderAction, InvocationKind.EXACTLY_ONCE) } }

V> buildMap(): Map<K, V> = emptyMap() Builds a new read-only [Map] by populating a [MutableMap] using the given [builderAction] and returning a read-only map with the same key-value pairs. If multiple pairs have the same key, The map passed as a receiver to the [builderAction] is valid only inside that function. Entries of the map are iterated in the order they were added by the [builderAction].

V> buildMap(): Map<K, V> = emptyMap() Builds a new read-only [Map] by populating a [MutableMap] using the given [builderAction] and returning a read-only map with the same key-value pairs. If multiple pairs have the same key, The map passed as a receiver to the [builderAction] is valid only inside that function. Entries of the map are iterated in the order they were added by the [builderAction].

IllegalArgumentException if the given [capacity] is negative.

* @sample samples.collections.Builders.Maps.buildMapSample

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun <K, V>
    buildMap(capacity: Int, @BuilderInference builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {
    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }
    return buildMapInternal(capacity, builderAction) as? @PublishedApi@SinceKotlin("1.3")@ExperimentalStdlibApi
    *inline
    *internal expect inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>.() -> Unit): Map<K, V>?
    * * @sample collections.Maps//* *usage.
    *MapIsNotEmpty
    * @PublishedApi@SinceKotlin("1.3")@ExperimentalStdlibApi
    *inline
    internal expect inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>.() ->
    Unit): Map<K, V>?
    * * @sample collections.Maps//* *usage.
    *MapIsNullOrEmpty
    * @PublishedApi
    *internal expect fun mapCapacity(expectedSize: Int): Int
    *Map
    *isNotEmpty(): Boolean = !isEmpty()
    *Map
    *OrNullEmpty
    * @PublishedApi
    *internal expect inline fun <K, V>
    *Map<out K, V>.isNullOrEmpty(): Boolean {
    *    contract {
    *        returns(false) implies (this@isNullOrEmpty != null)
    *    }
    *    return this == null || isEmpty()
    *}
    *Map
    *orElseEmpty(): Map<K, V> = this ?: emptyMap()
    *Map
    *ifEmpty(defaultValue: () -> R): R where M : Map<*, *>, M : R =
    *@Suppress("UNCHECKED_CAST") (this as Map<K, V>).ifEmpty()
    *Map
    *contains(key: K): Boolean = containsKey(key)
    *Map<out K,
    *V>.contains(key: K): Boolean = containsKey(key)
    *Map<K, V>.containsValue(value: V): Boolean =
    *this.containsKey(value)
    *Map<K,
    *V>.containsKey(key: K): Boolean =
    *@Suppress("UNCHECKED_CAST") (this as Map<K, V>).containsKey(key)
    *Map<K,
    *V>.remove(key: K): V? =
    *@Suppress("UNCHECKED_CAST") (this as MutableMap<K,
    *V>).remove(key)
    *Map.Entry<K, V>.component1(): K = key
    *Map.Entry<K, V>.component2(): V = value
    *Map.Entry<K, V>.isNotEmpty(): Boolean = !isEmpty()
    *Map.Entry<K, V>.isEmpty(): Boolean = isEmpty()
    *Map.Entry<K, V>.isNull(): Boolean = isNull()
    *Map.Entry<K, V>.isNullOrEmpty(): Boolean = isNullOrEmpty()
In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one. The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

Returns a map containing all key-value pairs with keys matching the given [predicate]. The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
public inline fun <K, V, R> mapKeys(transform: (Map.Entry<K, V>) -> R): Map<R, V> {
    return mapKeysTo(LinkedHashMap<R, V>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()
}
```

In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.

The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
    for (entry in this) {
        if (predicate(entry.key)) {
            destination.put(entry.key, entry.value)
        }
    }
    return destination
}
```

Returns a map containing all key-value pairs with values matching the given [predicate]. The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
public inline fun <K, V, R> mapKeysTo(destination: MutableMap<R, V>, itemToMap: (Map.Entry<K, V>) -> R): Map<R, V> {
    return mapKeysTo(LinkedHashMap<R, V>(mapCapacity(size)), itemToMap) // .optimizeReadOnlyMap()
}
```

In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.

The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
public inline fun <K, V, R> mapKeysTo(destination: MutableMap<R, V>, itemToMap: (Map.Entry<K, V>) -> R): Map<R, V> {
    for (entry in this) {
        if (predicate(entry.key)) {
            destination.put(entry.key, entry.value)
        }
    }
    return destination
}
```

In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.

The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
public inline fun <K, V, R> mapKeysTo(destination: MutableMap<R, V>, itemToMap: (Map.Entry<K, V>) -> R): Map<R, V> {
    for (entry in this) {
        if (predicate(entry.key)) {
            destination.put(entry.key, entry.value)
        }
    }
    return destination
}
```

In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one.

The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
public inline fun <K, V, R> mapKeysTo(destination: MutableMap<R, V>, itemToMap: (Map.Entry<K, V>) -> R): Map<R, V> {
    for (entry in this) {
        if (predicate(entry.key)) {
            destination.put(entry.key, entry.value)
        }
    }
    return destination
}
```

In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one. The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
public inline fun <K, V, R> mapKeysTo(destination: MutableMap<R, V>, itemToMap: (Map.Entry<K, V>) -> R): Map<R, V> {
    for (entry in this) {
        if (predicate(entry.key)) {
            destination.put(entry.key, entry.value)
        }
    }
    return destination
}
```

In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one. The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
public inline fun <K, V, R> mapKeysTo(destination: MutableMap<R, V>, itemToMap: (Map.Entry<K, V>) -> R): Map<R, V> {
    for (entry in this) {
        if (predicate(entry.key)) {
            destination.put(entry.key, entry.value)
        }
    }
    return destination
}
```

In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite the value associated with the former one. The returned map preserves the entry iteration order of the original map. * @sample samples.collections.Maps.Transformations.mapKeys

```kotlin
public inline fun <K, V, R> mapKeysTo(destination: MutableMap<R, V>, itemToMap: (Map.Entry<K, V>) -> R): Map<R, V> {
    for (entry in this) {
        if (predicate(entry.key)) {
            destination.put(entry.key, entry.value)
        }
    }
    return destination
}
```
the map.

```kotlin
public fun <K, V> Sequence<Pair<K, V>>.toMap(): Map<K, V> = toMap(LinkedHashMap<K, V>())....optimizeReadOnlyMap()... Returns a new read-only map containing all key-value pairs from the original map.

The returned map preserves the entry iteration order of the original map.

```
* the keys of which are contained in the given [keys] collection.
The returned map preserves the entry iteration order of the original map.

```kotlin
@SinceKotlin("1.1")
    this.toMutableMap().apply { minusAssign(keys) }.optimizeReadOnlyMap()

@SinceKotlin("1.1")
    this.toMutableMap().apply { minusAssign(keys) }.optimizeReadOnlyMap()

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline operator fun <K, V> MutableMap<K, V> minusAssign(key: K) {
    remove(key)
}

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline operator fun <K, V> MutableMap<K, V> minusAssign(keys: Iterable<K>) {
    this.keys.removeAll(keys)
}

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline operator fun <K, V> MutableMap<K, V> minusAssign(keys: Array<out K>) {
    this.keys.removeAll(keys)
}

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline operator fun <K, V> MutableMap<K, V> minusAssign(keys: Sequence<K>) {
    this.keys.removeAll(keys)
}
```

@PublishedApi
internal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) {
    0 -> emptyMap()
    1 -> toSingletonMapOrSelf()
    else -> this
}

// do not expose for now

@SinceKotlin("1.1")
    this.toMutableMap().apply { minusAssign(keys) }.optimizeReadOnlyMap()

@SinceKotlin("1.1")
    this.toMutableMap().apply { minusAssign(keys) }.optimizeReadOnlyMap()

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("\u6570\u95fb\u53f2\u79f0\u7ebf\u7f51\u70b9")
@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)
package kotlin.collections
import kotlin.contracts.*

internal object EmptySet : Set<Nothing>, Serializable {
    private const val serialVersionUID: Long = 3406603774387020532

    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()
    override fun hashCode(): Int = 0
    override fun toString(): String = "[]"
    override val size: Int get() = 0
    override fun isEmpty(): Boolean = true
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun iterator(): Iterator<Nothing> = EmptyIterator

    private fun readResolve(): Any = EmptySet
}

/**
 * Returns an empty read-only set. The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.emptyReadOnlySet
 */
public fun <T> emptySet(): Set<T> = EmptySet

/**
 * Returns a new read-only set with the given elements. The returned set is serializable (JVM).
 * @sample samples.collections.Collections.Sets.emptyReadOnlySet
 */
public fun <T> setOf(vararg elements: T): Set<T> = if (elements.size > 0) elements.toSet() else emptySet()

/**
 * Returns an empty read-only set. The returned set preserves the element iteration order.
 * @sample samples.collections.Collections.Sets.emptyReadOnlySet
 */
public fun <T> setOf(): Set<T> = emptySet()
fun <T> mutableSetOf(vararg elements: T): MutableSet<T> =
elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))

/** Returns a new [HashSet] with the given elements. */
@SinceKotlin("1.1")@kotlin.internal.InlineOnly
public inline fun <T> hashSetOf(vararg elements: T): HashSet<T> =
elements.toCollection(HashSet(mapCapacity(elements.size)))

/** Returns an empty new [LinkedHashSet]. */
@SinceKotlin("1.1")@kotlin.internal.InlineOnly
public inline fun <T> linkedSetOf(): LinkedHashSet<T> =
LinkedHashSet()

/** Returns a new [LinkedHashSet] with the given elements. */
public fun <T> linkedSetOf(vararg elements: T): LinkedHashSet<T> =
elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))

/**
 * Returns an empty new [LinkedHashSet].
 * @sample samples.collections.Collections.Sets.emptyLinkedHashSet
 */
@SinceKotlin("1.1")@kotlin.internal.InlineOnly
public inline fun <T> linkedSetOf(): LinkedHashSet<T> =
LinkedHashSet()

/** Returns a new read-only set only with those given elements, that are not null. */
Elements of the set are iterated in the order they were specified.
The returned set is serializable (JVM).

fun <T> Set<T>?.orEmpty(): Set<T> = this ?: emptySet()

internal fun <T> Set<T>.optimizeReadOnlySet() = when (size) {
    0 -> emptySet()
    1 -> setOf(iterator().next())
    else -> this
}

fun String.toByteOrNull(): Byte? = toByteOrNull(radix = 10)

/**
 * Parses the string as a signed [Byte] number and returns the result or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.1")@kotlin.internal.InlineOnly
public fun String.toByteOrNull(): Byte? = toByteOrNull(radix = 10)

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")
@file:Suppress("PLATFORM_CLASS_MAPPED_TO_KOTLIN")

package kotlin.text
the string as a signed [Byte] number and returns the result `null` or `null` if the string is not a valid representation of a number.

```kotlin
public fun String.toByteOrNull(radix: Int): Byte? { val int = this.toSignedIntOrNull(radix) ?: return null
    return int.toByte()
}
```

The above code snippet demonstrates how to parse a string into a Long number and return the result or `null` if the string is not a valid representation of a number.

---

The Open Source Used In DNAC Platform Antelope 9291

Kotlin Programming Language contributors.

*Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.*

---

Open Source Used In DNAC Platform Antelope 9291
kotlin
unn@PublishedApi
unn@internal fun uintCompare(v1: Int, v2: Int): Int = (v1 xor Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)
nunn@PublishedApi
unn@internal fun ulongCompare(v1: Long, v2: Long): Int = (v1.toLong() xor v2.toLong()).compareTo(v2.toLong())
nunn@PublishedApi
unn@internal fun uintDivide(v1: UInt, v2: UInt): UInt = (v1.toDouble() / v2.toDouble()).toUInt()
nunn@PublishedApi
unn@internal fun ulDiv(v1: ULong, v2: ULong): ULong = (v1.toDouble() / v2.toDouble()).toULong()
nunn@PublishedApi
unn@internal fun uintRemainder(v1: UInt, v2: UInt): UInt = (v1.toLong() % v2.toLong().toDouble()).toUInt()
// Division and remainder are based on Guava's UnsignedLongs implementation// Copyright 2011 The Guava Authors
unn@PublishedApi
unn@internal fun ulongDivide(v1: ULong, v2: ULong): ULong = (v1.toDouble() / v2.toDouble()).toULong()
nunn@PublishedApi
unn@internal fun ulongRemainder(v1: ULong, v2: ULong): ULong = (v1.toDouble() % v2.toDouble()).toULong()
unn@PublishedApi
unn@internal fun doubleToUInt(v: Double): UInt = when {
    v.isNaN() -> 0u
    v <= UInt.MIN_VALUE.toDouble() -> UInt.MIN_VALUE
    v >= UInt.MAX_VALUE.toDouble() -> UInt.MAX_VALUE
    v <= Int.MAX_VALUE.toDouble() -> v.toInt().toUInt()
    else -> (v - (v - Int.MAX_VALUE.toDouble() / v)).toDouble() * 2
}
nunn@PublishedApi
unn@internal fun doubleToULong(v: Double): ULong = when {
    v.isNaN() -> 0u
    v <= ULong.MIN_VALUE.toDouble() -> ULong.MIN_VALUE
    v >= ULong.MAX_VALUE.toDouble() -> ULong.MAX_VALUE
    v < Long.MAX_VALUE -> v.toLong().toULong()
    else -> (v - (v - Long.MAX_VALUE).toDouble()).toDouble() * 2048 + (v - Long.MAX_VALUE).toDouble() * 2
}
unn@internal fun uintToDouble(v: Int): Double = (v and Int.MAX_VALUE).toDouble() + (v ushr 31 shl 30).toDouble() * 2
unn@internal fun ulongToDouble(v: Long): Double = (v ushr 11).toDouble() * 2048 + (v and 2047).toDouble() * 2
unn@internal fun ulongToString(v: Long): String = ulongToString(v, 10)
unn@internal fun doubleToValue(v: Double) = when {
    v.isNaN() -> NaN
    v == Double.NEGATIVE_INFINITY -> Double.NEGATIVE_INFINITY
    v == Double.POSITIVE_INFINITY -> Double.POSITIVE_INFINITY
    v < 0 -> -doubleToValue(-v)
    v > 0 -> doubleToValue(v)
    v == 0L -> 0.0
    else -> 
}
unn@file:kotlin.jvm.JvmMultifileClass
unn@file:kotlin.jvm.JvmName("CollectionsKt")
nunn@package kotlin.collections
unn/***/unn* Given an [iterator] function constructs an [Iterable] instance that returns values through the [Iterator] * provided by that function. * @sample collections.Enumerables
unn@package kotlin
unn@contextual InlineOnly
unn@public inline fun <T> Iterable(crossinline iterator: () -> Iterator<T>): Iterable<T> = object : Iterable<T> {
    override fun iterator(): Iterator<T> = iterator()
}
unn@contextual
unn@internal class IndexingIterable<out T>(private val iteratorFactory: () -> Iterator<T>) : Iterable<IndexedValue<T>> {
    override fun iterator(): Iterator<IndexedValue<T>> = IndexingIterator(iteratorFactory())
}
unn@contextual
unn@PublishedApi
unn@internal fun collectionSizeOrNull(): Int? = if (this is Collection<*>) this.size else null
unn@contextual
unn@PublishedApi
unn@internal fun collectionSizeOrWarning(): Int? = if (this is Collection<*> && this.size == 0) 0 else warnAboutNullCollectionSize("")
Iterable<T>.collectionSizeOrDefault(default: Int) = if (this is Collection<*>) this.size else default

Converts this collection to a set, when it’s worth so and it doesn’t change contains method behavior. *

Iterable<T>.convertToSetForSetOperationWith(source: Iterable<T>): Collection<T> = when (this) {
    Set -> this
    Collection -> when (source)
        is Collection && source.size < 2 -> this
        else -> if (this.safeToConvertToSet()) toHashSet() else this
    }

Returns a single list of all elements from all collections in the given collection.

Iterable.Operations.flattenIterable(*\n
public fun <T> Iterable<T>.flatten(): List<T> {
    val result = ArrayList<T>()
    for (element in this) {
        result.addAll(element)
    }
    return result
}

Iterable.Operations.unzipIterable(*\n
public fun <T, R> Iterable<Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {
    val listT = ArrayList<T>(expectedSize)
    val listR = ArrayList<R>(expectedSize)
    for (pair in this) {
        listT.add(pair.first)
        listR.add(pair.second)
    }
    return listT to listR
}

SequencesKt

package kotlin.sequences

import kotlin.random.Random

/**
 * Given an [iterator] function constructs a [Sequence]
 * that returns values through the [Iterator]
 * provided by that function.
 * The values are evaluated lazily, and the
 * sequence is potentially infinite.

 * @sample samples.collections.Sequences.Building.sequenceFromIterator
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence(crossinline iterator: () -> Iterator<T>): Sequence<T> = object : Sequence<T> {
    override fun iterator(): Iterator<T> = iterator()
}

/**
 * Creates a
 * sequence that returns all elements from this iterator. The sequence is constrained to be iterated only once.

 * @sample samples.collections.Sequences.Usage.sequenceOrEmpty
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>?.orEmpty(): Sequence<T> = this ?: emptySequence()

/**
 * Returns a sequence that iterates through the elements either of
 * this sequence
 * or, if this sequence turns out to be empty, of the sequence returned by [defaultValue] function.

 * @sample samples.collections.Sequences.Transformations.flattenSequenceOfSequences
 */
public fun <T> Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }

Open Source Used In DNAC Platform Antelope 9293
elements from all iterables in this sequence. The operation is intermediate and stateless. Returns a sequence that yields elements of this sequence randomly shuffled. * Note that every iteration of the sequence returns elements in a different order. The operation is intermediate and stateful. A sequence that returns the values from the underlying sequence that either match or do not match the specified predicate. Returns a sequence that takes the index of the value in the underlying sequence along with the value itself. A sequence which combines values from the underlying sequence with their indices and returns them as IndexedValue objects. A sequence which combines values from the underlying sequence with their indices and returns them as IndexedValue objects. A sequence which combines values from the underlying sequence with their indices and returns them as IndexedValue objects.
var left = count
val iterator = sequence.iterator()
override fun next(): T {
    if (left == 0)
        throw NoSuchElementException()
    left--
    return iterator.next()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
}

override fun hasNext(): Boolean {
    return left > 0 && iterator.hasNext()
open

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("PreconditionsKt")

package kotlin

import kotlin.contracts.contract

/**
 * Throws an [IllegalArgumentException] if the [value] is false.
 * @sample samples.misc.Preconditions.failRequireWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun require(value: Boolean): Unit {
    contract {
        returns() implies value
    }
    require(value) {
        "Failed requirement."
    }
}

/**
 * Throws an [IllegalArgumentException] with the result of calling
 * [lazyMessage] if the [value] is null. Otherwise returns the not null value.
 * @sample samples.misc.Preconditions.failRequireWithLazyMessage
 */
@kotlin.internal.InlineOnly
    contract {
        returns() implies (value !=
        null)
    }
    if (value ==
null)
        throw
        IllegalArgumentException(lazyMessage()

            .toString())
    return
    value
}

/**
 * Returns a sequence defined by the function [seedFunction],
 * which is invoked to produce the starting value,
 * and the [nextFunction], which is invoked to calculate the next value based on
 * the previous one. The sequence produces values until it encounters first
 * `null` value. If [seedFunction] returns `null`, an empty sequence is produced.
 * The sequence can be iterated multiple times.
 * @see kotlin.sequences.sequence
 * @sample samples.collections.Sequences.Building.generateSequenceWithLazySeed
 */
public fun <T : Any>
    generateSequence(seedFunction: () -> T?,
    nextFunction: (T) -> T?): Sequence<T> =
    GeneratorSequence(seedFunction, nextFunction)

/**
 * Returns a sequence defined by
 * the function [seedFunction], which is invoked to produce the starting value,
 * and the function [nextFunction], which is invoked to calculate the next
 * value based on the previous one. The sequence produces values until it encounters first
 * `null` value. If [seedFunction] returns `null`, an empty sequence is produced.
 * The sequence can be iterated multiple times.
 * @see kotlin.sequences.sequence
 * @sample samples.collections.Sequences.Building.generateSequenceWithSeed
 */
@kotlin.internal.LowPriorityInOverloadResolution
public fun <T : Any>
generateSequence(seed: T?,
    nextFunction: (T) -> T?): Sequence<T> =
    GeneratorSequence({ seed },
    nextFunction)

/**
 * Returns a sequence which invokes the function to calculate the next value on each iteration until the function returns `null`
 * The returned sequence is constrained to be iterated only once.
 * @see kotlin.sequences.sequence
 * @sample samples.collections.Sequences.Building.generateSequence
 */
public fun <T : Any>
generateSequence(nextFunction: () -> T?)
    : Sequence<T> =
    GeneratorSequence(nextFunction, { nextFunction() }).constrainOnce()
```kotlin
null\n
    if (value == null) {\n        val message = lazyMessage()\n        throw IllegalArgumentException(message.toString())\n    } else {\n        return value\n    }
}

/**
 * Throws an [IllegalStateException] if the [value] is null. Otherwise returns the not null value.
 * @sample samples.misc.Preconditions.failWithLazyMessage\n * @sample samples.misc.Preconditions.failWithLazyMessage\n */
@kotlin.internal.InlineOnly\npublic inline fun check(value: Boolean, lazyMessage: () -> Any): Unit {\n    contract {\n        returns() implies value\n    }\n    if (!value) {\n        val message = lazyMessage()\n        throw IllegalStateException(message.toString())\n    } else {\n        return value\n    }
}

/**
 * Throws an [IllegalStateException] with the result of calling [lazyMessage] if the [value] is false.
 * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */
@kotlin.internal.InlineOnly\npublic inline fun checkNotNull(value: T?, lazyMessage: () -> Any): T {
    contract {\n        returns() implies (value != null)\n    }\n    if (value == null) {\n        val message = lazyMessage()\n        throw IllegalArgumentException(message.toString())\n    } else {\n        return value\n    }
}

/**
 * Throws an [IllegalStateException] with the given [message].
 * @sample samples.misc.Preconditions.failWithError\n */
@kotlin.internal.InlineOnly\npublic inline fun error(message: Any): Nothing = throw IllegalStateException(message.toString())
```

---

Open Source Used In DNAC Platform Antelope 9298
contains itself on any nesting level the behavior is undefined.

```kotlin
@SinceKotlin("1.1")
@kotlin.internal.LowPriorityInOverloadResolution
public actual fun <T> Array<out T>.contentDeepHashCode(): Int {
    return this.contentDeepHashCode()
}
```

```kotlin
@SinceKotlin("1.4")
@library("arrayDeepHashCode")
public actual fun <T> Array<out T>?.contentDeepHashCode(): Int {
    definedExternally
}
```

```kotlin
/**
 * Returns a hash code based on the contents of this array as if it is 
 * a [List]. * Nested arrays are treated as lists too. * If any of arrays 
 * contains itself on any nesting level the behavior is undefined.
 *
@SinceKotlin("1.4")
@library("arrayDeepHashCode")
public actual fun <T> Array<out T>?.contentDeepHashCode(): Int {
    definedExternally
}
```

```kotlin
/**
 * Returns a string representation of the contents of this array as if it is a [List]. * Nested arrays are treated as lists too. * If any of arrays contains itself on any nesting level the reference is rendered as `\"[...]\"` to prevent recursion. * @sample
samples.collections.Arrays.ContentOperations.contentDeepToString

@SinceKotlin("1.1")
@kotlin.internal.LowPriorityInOverloadResolution
public actual fun <T> Array<out T>.contentDeepToString(): String {
    return this.contentDeepToString()
}
```

```kotlin
@SinceKotlin("1.4")
@library("arrayDeepToString")
public actual fun <T> Array<out T>?.contentDeepToString(): String {
    definedExternally
}
```

```kotlin
/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the 
 * [equals][Any.equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. * @sample
"Use Kotlin compiler 1.4 to avoid deprecation warning." @ SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean {
    return this.contentEquals(other)
}
```

```kotlin
/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the 
 * [equals][Any.equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. * @sample
"Use Kotlin compiler 1.4 to avoid deprecation warning." @ SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual infix fun ByteArray.contentEquals(other: ByteArray): Boolean {
    return this.contentEquals(other)
}
```

```kotlin
/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the 
 * [equals][Any.equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. * @sample
"Use Kotlin compiler 1.4 to avoid deprecation warning." @ SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual infix fun ShortArray.contentEquals(other: ShortArray): Boolean {
    return this.contentEquals(other)
}
```

```kotlin
/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the 
 * [equals][Any.equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. * @sample
"Use Kotlin compiler 1.4 to avoid deprecation warning." @ SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual infix fun IntArray.contentEquals(other: IntArray): Boolean {
    return this.contentEquals(other)
}
```

```kotlin
/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,
 * i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the 
 * [equals][Any.equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. * @sample
"Use Kotlin compiler 1.4 to avoid deprecation warning." @ SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual infix fun LongArray.contentEquals(other: LongArray): Boolean {
    return this.contentEquals(other)
}
```
function.

* For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hideSince = "1.4")
public actual infix fun FloatArray.contentEquals(other: FloatArray): Boolean {
    return this.contentEquals(other)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

* The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hideSince = "1.4")
public actual infix fun DoubleArray.contentEquals(other: DoubleArray): Boolean {
    return this.contentEquals(other)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

* The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hideSince = "1.4")
public actual infix fun BooleanArray.contentEquals(other: BooleanArray): Boolean {
    return this.contentEquals(other)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

* The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hideSince = "1.4")
public actual infix fun CharArray.contentEquals(other: CharArray): Boolean {
    return this.contentEquals(other)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

* The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean {
    definedExternally
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

* The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun ByteArray?.contentEquals(other: ByteArray?): Boolean {
    definedExternally
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

* The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun ShortArray?.contentEquals(other: ShortArray?): Boolean {
    definedExternally
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

* The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun IntArray?.contentEquals(other: IntArray?): Boolean {
    definedExternally
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.

* The elements are compared for equality with the [equals][Any.equals] function.
 * For floating point numbers it means that ‘NaN’ is equal to itself and ‘-0.0’ is not equal to ‘0.0’.

@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun LongArray?.contentEquals(other: LongArray?): Boolean {
    definedExternally
}
* The elements are compared for equality with the `equals` function.  
* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```kotlin
@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun FloatArray?.contentEquals(other: FloatArray?): Boolean {
    definedExternally
}
```

```kotlin
/**
 * Returns `true` if the two specified arrays are
 * structurally equal to one another, i.e. contain the same
 * number of the same elements in the same order.
 * The elements are compared for equality with the `equals` function.  
* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```kotlin
@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean {
    definedExternally
}
```

```kotlin
/**
 * Returns `true` if the two specified arrays are
 * structurally equal to one another, i.e. contain the same
 * number of the same elements in the same order.
 * The elements are compared for equality with the `equals` function.  
* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```kotlin
@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean {
    definedExternally
}
```

```kotlin
/**
 * Returns `true` if the two specified arrays are
 * structurally equal to one another, i.e. contain the same
 * number of the same elements in the same order.
 * The elements are compared for equality with the `equals` function.  
* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

```kotlin
@SinceKotlin("1.4")
@library("arrayEquals")
public actual infix fun CharArray?.contentEquals(other: CharArray?): Boolean {
    definedExternally
}
```

```kotlin
/**
 * Returns a hash code based on the contents of this array
 * as if it is `List`.  
* Use Kotlin compiler 1.4 to avoid deprecation warning.

```kotlin
@SinceKotlin("1.1")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual fun <T> Array<out T>.contentHashCode(): Int {
    return this.contentHashCode()
}
```
the contents of this array as if it is [List].

fun ByteArray?.contentHashCode(): Int {
    return this.contentHashCode() // Returns a hash code based on the contents of this array as if it is [List].
}

fun ShortArray?.contentHashCode(): Int {
    return this.contentHashCode() // Returns a hash code based on the contents of this array as if it is [List].
}

fun IntArray?.contentHashCode(): Int {
    return this.contentHashCode() // Returns a hash code based on the contents of this array as if it is [List].
}

fun LongArray?.contentHashCode(): Int {
    return this.contentHashCode() // Returns a hash code based on the contents of this array as if it is [List].
}

fun FloatArray?.contentHashCode(): Int {
    return this.contentHashCode() // Returns a hash code based on the contents of this array as if it is [List].
}

fun DoubleArray?.contentHashCode(): Int {
    return this.contentHashCode() // Returns a hash code based on the contents of this array as if it is [List].
}

fun BooleanArray?.contentHashCode(): Int {
    return this.contentHashCode() // Returns a hash code based on the contents of this array as if it is [List].
}

fun CharArray?.contentHashCode(): Int {
    return this.contentHashCode() // Returns a hash code based on the contents of this array as if it is [List].
}

fun <T> Array<out T>.contentToString(): String {
    return this.contentToString() // Returns a string representation of the contents of the specified array as if it is [List].
}

fun ByteArray.contentToString(): String {
    return this.contentToString() // Returns a string representation of the contents of the specified array as if it is [List].
}

fun ShortArray.contentToString(): String {
    return this.contentToString() // Returns a string representation of the contents of the specified array as if it is [List].
}

fun IntArray.contentToString(): String {
    return this.contentToString() // Returns a string representation of the contents of the specified array as if it is [List].
}

fun LongArray.contentToString(): String {
    return this.contentToString() // Returns a string representation of the contents of the specified array as if it is [List].
}

fun FloatArray.contentToString(): String {
    return this.contentToString() // Returns a string representation of the contents of the specified array as if it is [List].
}

fun DoubleArray.contentToString(): String {
    return this.contentToString() // Returns a string representation of the contents of the specified array as if it is [List].
}

fun BooleanArray.contentToString(): String {
    return this.contentToString() // Returns a string representation of the contents of the specified array as if it is [List].
}
representation of the contents of the specified array as if it is [List].

@sample

*/
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.1")
@DepreciatedSinceKotlin(hiddenSince = "1.4")
public actual fun CharArray.contentToString(): String {
    return this.contentToString()
}

@sample

*/
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual fun CharArray.contentToString(): String {
    return this.contentToString()
}

@SinceKotlin("1.4")
@library("arrayToString")
public actual fun <T> BooleanArray?.contentToString(): String {
    definedExternally
}

@SinceKotlin("1.4")
@library("arrayToString")
public actual fun CharArray?.contentToString(): String {
    definedExternally
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it
 * overlaps with the destination range. It's also allowed to pass the same array in the [destination] and even specify the subrange so that it
 * overlaps with the destination range. It's also allowed to pass the same array in the [destination] and even specify the subrange so that it
 * overlaps with the destination range.
 * 
 * @param destination the array to copy to.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startInclusion the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endExclusion the end (exclusive) of the subrange to copy, size of this array by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startInclusion the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endExclusion the end (exclusive) of the subrange to copy, size of this array by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startInclusion the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endExclusion the end (exclusive) of the subrange to copy, size of this array by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startInclusion the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endExclusion the end (exclusive) of the subrange to copy, size of this array by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startInclusion the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endExclusion the end (exclusive) of the subrange to copy, size of this array by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startInclusion the beginning (inclusive) of the subrange to copy, 0 by default.
beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @param destination the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @param destination the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`
 * @param destination the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`
 * @param destination the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @param destination the position in the [destination] array to copy to, 0 by default.\n * @param destination the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @param destination the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
destination.unsafeCast<Array<Long>>()}, destinationOffset, startIndex, endIndex)
    return destination
}

/**
* Copies this array or its subrange into the [destination] array and returns that array.
* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
* @param destination the array to copy to.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset].
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.
*/
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray {
    arrayCopy(this.unsafeCast<Array<Float>>(), destination.unsafeCast<Array<Float>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
* Copies this array or its subrange into the [destination] array and returns that array.
* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
* @param destination the array to copy to.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset].
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.
*/
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray {
    arrayCopy(this.unsafeCast<Array<Double>>(), destination.unsafeCast<Array<Double>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
* Copies this array or its subrange into the [destination] array and returns that array.
* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
* @param destination the array to copy to.
* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset].
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.
*/
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {
    arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(), destinationOffset, startIndex, endIndex)
    return destination
}
starting at the specified [destinationOffset].
  * or when that index is out of the [destination] array indices range.
  *
  * @return the [destination] array.

*\n@SinceKotlin("1.3\")@kotlin.internal.InlineOnly\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n\npublic actual inline fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray {
  arrayCopy(this.unsafeCast<Array<Char>>(), destination.unsafeCast<Array<Char>>(), destinationOffset, startIndex, endIndex) return destination
}

/**
  * Returns new array which is a copy of the original array.
  * @sample samples.collections.Arrays.CopyOfOperations.copyOf
  */
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline fun <T> Array<out T>.copyOf(): Array<T> {
  return this.asDynamic().slice().\n}

/**
  * Returns new array which is a copy of the original array, resized to the given [newSize].
  * The copy is either truncated or padded at the end with zero values if necessary.
  * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
  */
public actual fun ByteArray.copyOf(newSize: Int): ByteArray {
  require(newSize >= 0) { "Invalid new array size: $newSize." }
  return fillFrom(this, ByteArray(newSize))
}

/**
  * Returns new array which is a copy of the original array, resized to the given [newSize].
  * The copy is either truncated or padded at the end with zero values if necessary.
  * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
  */
public actual fun ShortArray.copyOf(newSize: Int): ShortArray {
  require(newSize >= 0) { "Invalid new array size: $newSize." }
  return fillFrom(this, ShortArray(newSize))
}

/**
  * Returns new array which is a copy of the original array, resized to the given [newSize].
  * The copy is either truncated or padded at the end with zero values if necessary.
  * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
  */
public actual fun IntArray.copyOf(newSize: Int): IntArray {
  require(newSize >= 0) { "Invalid new array size: $newSize." }
  return fillFrom(this, IntArray(newSize))
}

/**
  * Returns new array which is a copy of the original array, resized to the given [newSize].
  * The copy is either truncated or padded at the end with zero values if necessary.
  * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
  */
public actual fun LongArray.copyOf(newSize: Int): LongArray {
  require(newSize >= 0) { "Invalid new array size: $newSize." }
  return fillFrom(this, LongArray(newSize))
}

/**
  * Returns new array which is a copy of the original array, resized to the given [newSize].
  * The copy is either truncated or padded at the end with zero values if necessary.
  * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
  */
public actual fun FloatArray.copyOf(newSize: Int): FloatArray {
  require(newSize >= 0) { "Invalid new array size: $newSize." }
  return fillFrom(this, FloatArray(newSize))
}

/**
  * Returns new array which is a copy of the original array, resized to the given [newSize].
  * The copy is either truncated or padded at the end with zero values if necessary.
  * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
  */
public actual fun DoubleArray.copyOf(newSize: Int): DoubleArray {
  require(newSize >= 0) { "Invalid new array size: $newSize." }
  return fillFrom(this, DoubleArray(newSize))
}

/**
  * Returns new array which is a copy of the original array, resized to the given [newSize].
  * The copy is either truncated or padded at the end with zero values if necessary.
  * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
  */
public actual fun BooleanArray.copyOf(newSize: Int): BooleanArray {
  require(newSize >= 0) { "Invalid new array size: $newSize." }
  return fillFrom(this, BooleanArray(newSize))
}

/**
  * Returns new array which is a copy of the original array, resized to the given [newSize].
  * The copy is either truncated or padded at the end with zero values if necessary.
  * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
  */
public actual fun CharArray.copyOf(newSize: Int): CharArray {
  require(newSize >= 0) { "Invalid new array size: $newSize." }
  return fillFrom(this, CharArray(newSize))
}

Open Source Used In DNAC Platform Antelope  9307
samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
public actual fun IntArray.copyOf(newSize: Int): IntArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, IntArray(newSize))
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public actual fun LongArray.copyOf(newSize: Int): LongArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("LongArray", arrayCopyResize(this, newSize, 0L))
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with zero values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public actual fun FloatArray.copyOf(newSize: Int): FloatArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, FloatArray(newSize))
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with null values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public actual fun CharArray.copyOf(newSize: Int): CharArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("CharArray", fillFrom(this, CharArray(newSize)))
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at the end with `false` values if necessary.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public actual fun BooleanArray.copyOf(newSize: Int): BooleanArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return withType("BooleanArray", arrayCopyResize(this, newSize, false))
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizingCopyOf
 */
@Suppress("ACTUAL_WITHOUT_EXPECT")
public actual fun <T> Array<out T>.copyOf(newSize: Int): Array<T?> {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return arrayCopyResize(this, newSize, (null))
}

/**
 * Returns new array which is a copy of the specified range of the original array.
 * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
 * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.
 * @sample samples.collections.Arrays.CopyOfOperations.resizingCopyOf
 */
@Suppress("ACTUAL_WITHOUT_EXPECT")
public actual fun <T> Array<out T>.copyOf(newSize: Int): Array<T?> {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return arrayCopyResize(this, newSize, (null))
}
T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<
    return this.asDynamic().slice(fromIndex, toIndex)
}

/*
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of the original array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/*
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of the original array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/*
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of the original array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/*
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of the original array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return withType("LongArray", this.asDynamic().slice(fromIndex, toIndex)
}

/*
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of the original array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/*
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of the original array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return this.asDynamic().slice(fromIndex, toIndex)
}

/*
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of the original array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return withType("BooleanArray", this.asDynamic().slice(fromIndex, toIndex)
}

/*
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of the original array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return withType("BooleanArray", this.asDynamic().slice(fromIndex, toIndex)
}
[fromIndex] is greater than [toIndex].

fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return withType("CharArray", this.asDynamic().slice(fromIndex, toIndex))
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return withType("CharArray", this.asDynamic().slice(fromIndex, toIndex))
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public actual fun <T> IntArray.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}
actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
    * Fills this array or its subrange with the specified [element] value.
    * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
    * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
    * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
    * @throws IndexOutOfBoundsException if [fromIndex] is greater than [toIndex].
}

actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
    * Fills this array or its subrange with the specified [element] value.
    * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
    * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
    * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
    * @throws IndexOutOfBoundsException if [fromIndex] is greater than [toIndex].
}

actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
    * Returns an array containing all elements of the original array and then the given [element].
}

actual fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
    * Returns an array containing all elements of the original array and then the given [element].
    * @return Array<Char> containing the concatenated elements.
}

@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
actual fun <T> Array<out T>.plus(element: T): Array<T> {
    return this.asDynamic().concat(arrayOf(element))
}

@SinceKotlin("1.3")
@Suppress("NOTHING_TO_INLINE")
actual inline operator fun ByteArray.plus(element: Byte): ByteArray {
    return plus(byteArrayOf(element))
}

@SinceKotlin("1.3")
@Suppress("NOTHING_TO_INLINE")
actual inline operator fun ShortArray.plus(element: Short): ShortArray {
    return plus(shortArrayOf(element))
}

@SinceKotlin("1.3")
@Suppress("NOTHING_TO_INLINE")
actual inline operator fun IntArray.plus(element: Int): IntArray {
    return plus(intArrayOf(element))
}

@SinceKotlin("1.3")
@Suppress("NOTHING_TO_INLINE")
actual inline operator fun LongArray.plus(element: Long): LongArray {
    return plus(longArrayOf(element))
}

@SinceKotlin("1.3")
@Suppress("NOTHING_TO_INLINE")
actual inline operator fun FloatArray.plus(element: Float): FloatArray {
    return plus(floatArrayOf(element))
}

@SinceKotlin("1.3")
@Suppress("NOTHING_TO_INLINE")
actual inline operator fun DoubleArray.plus(element: Double): DoubleArray {
    return plus(doubleArrayOf(element))
}

Open Source Used In DNAC Platform Antelope  9311
the given [element].

@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun CharArray.plus(element: Char): CharArray {
    return plus(charArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("ACTUAL_WITHOUT_EXPECT")
public actual operator fun <T> Array<out T>.plus(elements: Collection<T>): Array<T> {
    return arrayPlusCollection(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
public actual operator fun ShortArray.plus(elements: Collection<Short>): ShortArray {
    return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline operator fun <T> Array<out T>.plus(elements: Array<out T>): Array<T> {
    return this.asDynamic().concat(elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun CharArray.plus(elements: CharArray): CharArray {
    return plus(charArrayOf(elements))
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ByteArray.plus(elements: ByteArray): ByteArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ShortArray.plus(elements: ShortArray): ShortArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun IntArray.plus(elements: IntArray): IntArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun LongArray.plus(elements: LongArray): LongArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun CharArray.plus(elements: CharArray): CharArray {
    return from(charArrayOf(elements))
}
DoubleArray(): DoubleArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
public actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
public actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
public actual inline operator fun CharArray.plus(elements: CharArray): CharArray {
    return primitiveArrayConcat(this, elements)
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
public actual inline fun <T> Array<out T>.plusElement(element: T): Array<T> {
    return this.asDynamic().concat(arrayOf(element))
}

/**
 * Sorts the array in-place.
 */
@library("primitiveArraySort")
public actual fun IntArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 */
public actual fun LongArray.sort(): Unit {
    if (size > 1) sortArrayWith(this, a: Long, b: Long -> a.compareTo(b))
}

/**
 * Sorts the array in-place.
 */
@library("primitiveArraySort")
public actual fun ByteArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 */
@library("primitiveArraySort")
public actual fun ShortArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 */
@library("primitiveArraySort")
public actual fun DoubleArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 */
@library("primitiveArraySort")
public actual fun FloatArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 */
public actual fun <T : Comparable<T>> Array<out T>.sort(): Unit {
    if (size > 1) sortArrayWith(this)
}

/**
 * Sorts a range in the array in-place.
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    sortArrayWith(this, fromIndex, toIndex, naturalOrder())
}

/**
 * Sorts a range in the array in-place.
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val subarray =
this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ByteArray>().subarray.sort() @param fromIndex the start of the range (inclusive) to sort, 0 by default.
@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

/**
 * Sorts a range in the array in-place.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    // Implementation...
}

/**
 * Sorts a range in the array in-place.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    // Implementation...
}

/**
 * Sorts a range in the array in-place.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    // Implementation...
}

/**
 * Sorts a range in the array in-place.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    // Implementation...
}

/**
 * Sorts a range in the array in-place.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    // Implementation...
}

/**
 * Sorts a range in the array in-place.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    // Implementation...
}
Returns a *typed* object array containing all of the elements of this primitive array.

The functions are called sequentially, receive the given values [a] and [b] and return [Comparable] * objects. As soon as the [Comparable] instances returned by a function for [a] and [b] values do not compare as equal, the result of that comparison is returned.

The function is applied to the given values [a] and [b] and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The function is applied to the given values [a] and [b] and return objects of type K which are then being compared with the given [comparator].

Null is considered less than any value.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The function is applied to the given values [a] and [b] and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The function is applied to the given values [a] and [b] and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The function is applied to the given values [a] and [b] and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The function is applied to the given values [a] and [b] and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given [comparator].

The functions are called sequentially, receive the given values `a` and `b` and return objects of type K which are then being compared with the given `K`, and return [Comparable] objects. The result of comparison of these [Comparable] instances is returned.
samples.comparisons.Comparisons.compareByDescendingWithSingleSelector

*\*In@kotlin.internal.InlineOnly\*/public inline fun <T> compareByDescending(selector: (T) -> Comparable<*?>): Comparator<T> =
    Comparator { a, b -> compareValuesBy(b, a, selector) }\*\*/

Creates a descending comparator using the [selector] function to transform values being compared and then applying *\*In* to compare transformed values.\*\*In* Note that an order of [selector] is reversed by this wrapper.\*\*In* *@sample

samples.comparisons.Comparisons.compareByDescendingWithComparator

*\*In@kotlin.internal.InlineOnly\*/public inline fun <T, K> compareByDescending(comparator: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =
    Comparator { a, b -> compareValuesBy(b, a, comparator, selector) }\*\*/

Creates a comparator, comparing values after the primary comparator defined them equal. It uses\*\*In* the function to transform values comparing the primary comparator defined them equal. It uses\*\*In* the [selector] function to transform values being compared and then compares them with the given [comparator].\*\*In* *@sample

samples.comparisons.Comparisons.thenBy

Comparator<T>.thenBy(crossinline selector: (T) -> Comparable<*?>): Comparator<T> =
    Comparator { a, b ->\n        val previousCompare = this@thenBy.compare(a, b)\n        if (previousCompare != 0) previousCompare else compareValuesBy(a, b, selector)\n    }\*\*/

Creates a comparator comparing values after the primary comparator defined them equal. It uses\*\*In* the [selector] function to transform values being compared and then compares them with the given [comparator].\*\*In* *@sample

samples.comparisons.Comparisons.thenByWithComparator

Comparator<T>.thenByWithComparator<in K, K>: Comparator<T> =
    Comparator { a, b ->\n        val previousCompare = this@thenByWithComparator.compare(a, b)\n        if (previousCompare != 0) previousCompare else compareValuesBy(a, b, comparator, selector)\n    }\*\*/

Creates a descending comparator using the primary comparator and\*\*In* the function to transform values comparing the primary comparator defined them equal. It uses\*\*In* the [selector] function to transform values being compared and then compares them with the given [comparator].\*\*In* *@sample

samples.comparisons.Comparisons.thenByDescending

Comparator<T>.thenByDescending<in K, K>: Comparator<T> =
    Comparator { a, b ->\n        val previousCompare = this@thenByDescending.compare(a, b)\n        if (previousCompare != 0) previousCompare else compareValuesBy(a, b, comparator, selector)\n    }\*\*/

Creates a comparator using the primary comparator and function to calculate a result of comparison.\*\*In* *@sample

samples.comparisons.Comparisons.thenByWithComparatorDescending

Comparator<T>.thenByWithComparatorDescending<in K, K>: Comparator<T> =
    Comparator { a, b ->\n        val previousCompare = this@thenByWithComparator.compare(a, b)\n        if (previousCompare != 0) previousCompare else compareValuesBy(a, b, comparator, selector)\n    }\*\*/

Creates a descending comparator using the primary comparator and function to calculate a result of comparison.\*\*In* *@sample

samples.comparisons.Comparisons.thenComparator

Comparator<T>.thenComparator<in a, b, T>: Comparator<T> =
    Comparator { a, b ->\n        val previousCompare = this@thenComparator.compare(a, b)\n        if (previousCompare != 0) previousCompare else comparison(a, b)\n    }\*\*/

Extends the given [comparator] of non-nullable values to a comparator of nullable values\*\*In* considering ‘null’ value less than any other value.\*\*In* *@sample

Open Source Used In DNAC Platform Antelope  9317
samples.comparisons.Comparisons.nullsFirstLastWithComparator

/**
 * Provides a comparator of nullable [Comparable] values in * considering ‘null’ value less than any other value.
 */
public fun <T : Any>
    nullsFirst(comparator: Comparator<in T>): Comparator<T?> =
    Comparator { a, b ->
        when {
            a
            b == null -> -1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Provides a comparator of nullable [Comparable] values in * considering ‘null’ value greater than any other value.
 */
public fun <T : Any>
    nullsLast(comparator: Comparator<in T>): Comparator<T?> =
    Comparator { a, b ->
        when {
            a
            b == null -> 1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Returns a comparator that compares [Comparable] objects in natural order.
 */
public class NaturalOrderComparator : Comparator<Comparable<Any>> {
    override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)
    fun reversed(): Comparator<Comparable<Any>> = ReverseOrderComparator
}

/**
 * Returns a comparator that compares [Comparable] objects in reversed natural order.
 */
public class ReverseOrderComparator : Comparator<Comparable<Any>> {
    override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)
    fun reversed(): Comparator<Comparable<Any>> = NaturalOrderComparator
}

/**
 * An exception is thrown to indicate that a method body remains to be implemented.
 */
public class NotImplementedError(message: String = "An operation is not implemented.") : Error(message)

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(): Nothing = throw NotImplementedError()

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 * @param reason a string explaining why the implementation is missing.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(reason: String): Nothing = throw NotImplementedError("An operation is not implemented: $reason")

/**
 * Calls the specified function [block] and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <R> run(block: () -> R): R = block

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StandardKt")
package kotlin

import kotlin.contracts.*

/**
 * An exception is thrown to indicate that a method body remains to be implemented.
 */
public class NotImplementedError(message: String = "An operation is not implemented.") : Error(message)

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(): Nothing = throw NotImplementedError()

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 * @param reason a string explaining why the implementation is missing.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(reason: String): Nothing = throw NotImplementedError("An operation is not implemented: $reason")

/**
 * Calls the specified function [block] and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <R> run(block: () -> R): R = block

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StandardKt")
package kotlin

import kotlin.contracts.*

/**
 * An exception is thrown to indicate that a method body remains to be implemented.
 */
public class NotImplementedError(message: String = "An operation is not implemented.") : Error(message)

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(): Nothing = throw NotImplementedError()

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 * @param reason a string explaining why the implementation is missing.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(reason: String): Nothing = throw NotImplementedError("An operation is not implemented: $reason")

/**
 * Calls the specified function [block] and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <R> run(block: () -> R): R = block

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
callsInPlace(block, InvocationKind.EXACTLY_ONCE)
   return block()
}/*
 * Calls the specified function [block] with 'this' value as its receiver and returns its result.
 * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#apply).
 */
@kotlin.internal.InlineOnly
public actual inline fun <T, R> T.apply(block: (T) -> R): R {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return block()
}

/**
 * Calls the specified function [block] with the given [receiver] as its receiver and returns its result.
 * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#apply).
 */
@kotlin.internal.InlineOnly
public actual inline fun <T, R> T.apply(receiver: T, block: (T) -> R): R {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return receiver.block()
}

/**
 * Calls the specified function [block] with 'this' value as its argument and returns 'this' value.
 * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#apply).
 */
@kotlin.internal.InlineOnly
public actual inline fun <T> T.let(block: (T) -> Unit): T {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    return block(this)
}

/**
 * Returns `this` value if it satisfies the given [predicate] or `null`, if it doesn't.
 * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeIf).
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
public actual inline fun <T> T.takeIf(predicate: (T) -> Boolean): T? {
    contract {
        callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)
    }
    return if (predicate(this)) this else null
}

/**
 * Returns `this` value if it _does not_ satisfy the given [predicate] or `null`, if it does.
 * A zero-based index of current iteration is passed as a parameter to [action].
 * For sample usage see [ControlFlow.repeat](https://kotlinlang.org/docs/reference/scope-functions.html#repeat).
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
@Suppress("DEPRECATION_ERROR")
public actual inline fun <T, R> T.repeat(times: Int, action: (Int) -> Unit) {
    contract {
        callsInPlace(action) (null)
    }

    for (index in 0 until times) {
        action(index)
    }
}

/**
 * Returns the greater of two values.
 * If values are equal, returns the first one.
 * For sample usage see [ControlFlow.repeat](https://kotlinlang.org/docs/reference/scope-functions.html#repeat).
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
public actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@kotlin.internal.Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Byte, b: Byte): Byte {
    return Math.max(a.toInt(), b.toInt()).unsafeCast<Byte>()
}

/**
 * Returns the greater of two values.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
@kotlin.internal.Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Short, b: Short): Short {
    return Math.max(a.toInt(), b.toInt()).unsafeCast<Short>()
}

/**
 * Returns the greater of two values.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
public actual inline fun maxOf(a: Int, b: Int): Int {
    return Math.max(a, b)
}

/**
 * Returns the greater of two values.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
@kotlin.internal.Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Byte, b: Byte): Byte {
    return Math.max(a.toInt(), b.toInt()).unsafeCast<Byte>()
}

/**
 * Returns the greater of two values.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
@kotlin.internal.Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Short, b: Short): Short {
    return Math.max(a.toInt(), b.toInt()).unsafeCast<Short>()
}

/**
 * Returns the greater of two values.
 */
@kotlin.internal.InlineOnly
@SinceKotlin("1.1")
@kotlin.internal.Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Int, b: Int): Int {
    return Math.max(a, b)
}

}
public actual inline fun maxOf(a: Long, b: Long): Long {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual inline fun maxOf(a: Float, b: Float): Float {
    return Math.max(a, b)
}

/**
 * Returns the greater of two values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual inline fun maxOf(a: Double, b: Double): Double {
    return Math.max(a, b)
}

/**
 * Returns the greater of three values.
 * 
 * If there are multiple equal maximal values, returns the first of them.
 */
public actual inline fun maxOf(a: Byte, b: Byte, c: Byte): Byte {
    return Math.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()
}

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual inline fun maxOf(a: Short, b: Short, c: Short): Short {
    return Math.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()
}

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual inline fun maxOf(a: Int, b: Int, c: Int): Int {
    return Math.max(a, b, c)
}

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual inline fun maxOf(a: Long, b: Long, c: Long): Long {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual inline fun maxOf(a: Float, b: Float, c: Float): Float {
    return Math.max(a, b, c)
}

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual inline fun maxOf(a: Double, b: Double, c: Double): Double {
    return Math.max(a, b, c)
}

/**
 * Returns the greater of the given values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual fun <T : Comparable<T>> maxOf(a: T, vararg other: T): T {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual fun maxOf(a: Byte, vararg other: Byte): Byte {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual fun maxOf(a: Short, vararg other: Short): Short {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual fun maxOf(a: Int, vararg other: Int): Int {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual fun maxOf(a: Long, vararg other: Long): Long {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual fun maxOf(a: Float, vararg other: Float): Float {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual fun maxOf(a: Double, vararg other: Double): Double {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
public actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Long, b: Long): Long {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Float, b: Float): Float {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Double, b: Double): Double {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Byte, b: Byte): Byte {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Short, b: Short): Short {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Int, b: Int): Int {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Long, b: Long): Long {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Float, b: Float): Float {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Double, b: Double): Double {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Byte, b: Byte): Byte {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Short, b: Short): Short {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Int, b: Int): Int {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual fun minOf(a: Long, b: Long): Long {
    return Math.min(a, b)
}
public actual inline fun minOf(a: Short, b: Short): Short {
    return Math.min(a.toInt(), b.toInt()).unsafeCast<Short>()
}

/**
 * Returns the smaller of two values.
 */
public actual inline fun minOf(a: Int, b: Int): Int {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 * If either value is `NaN`, returns `NaN`.
 */
public actual inline fun minOf(a: Long, b: Long): Long {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 * If there are multiple equal minimal values, returns the first of them.
 */
public actual inline fun minOf(a: Float, b: Float): Float {
    return Math.min(a, b)
}

/**
 * Returns the smaller of two values.
 */
public actual inline fun minOf(a: Double, b: Double): Double {
    return Math.min(a, b)
}

/**
 * Returns the smaller of three values.
 * If there are multiple equal minimal values, returns the first of them.
 */
public actual fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the smaller of three values.
 */
public actual fun minOf(a: Byte, vararg other: Byte): Byte {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of three values.
 */
public actual fun minOf(a: Short, vararg other: Short): Short {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of three values.
 */
public actual fun minOf(a: Int, vararg other: Int): Int {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of three values.
 */
public actual fun minOf(a: Long, vararg other: Long): Long {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of three values.
 */
public actual fun minOf(a: Float, vararg other: Float): Float {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of three values.
 */
public actual fun minOf(a: Double, vararg other: Double): Double {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

// Auto-generated file. DO NOT EDIT!

package kotlin

import kotlin.experimental.*

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class ULong @PublishedApi internal constructor(ival data: Long) : Comparable<ULong> {

    companion object {
        /**
         * A constant holding the minimum value an instance of ULong can have.
         */
        public const val MIN_VALUE: ULong = ULong(0)

        /**
         * A constant holding the maximum value an instance of ULong can have.
         */
        public const val MAX_VALUE: ULong = ULong(-1)

        /**
         * The number of bytes used to represent an instance of ULong in a binary form.
         */
        public const val SIZE_BYTES: Int = 8

        /**
         * The number of bits used to represent an instance of ULong in a binary form.
         */
        public const val SIZE_BITS: Int = 64
    }

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other, or a positive number if it's greater than other.
     */
    public inline operator fun compareTo(other: UByte): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other, or a positive number if it's greater than other.
     */
    public inline operator fun compareTo(other: UShort): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other, or a positive number if it's greater than other.
     */
    public inline operator fun compareTo(other: UInt): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other, or a positive number if it's greater than other.
     */
    public inline operator fun compareTo(other: ULong): Int = ulongCompare(this.data, other.data)

    /** Adds the other value to this value. */
    public inline operator fun plus(other: UByte): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    public inline operator fun plus(other: UShort): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    public inline operator fun plus(other: UInt): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    public inline operator fun plus(other: ULong): ULong = ULong(this.data.plus(other.data))

    /** Subtracts the other value from this value. */
    public inline operator fun minus(other: UByte): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    public inline operator fun minus(other: UShort): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    public inline operator fun minus(other: UInt): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    public inline operator fun minus(other: ULong): ULong = ULong(this.data.minus(other.data))

    /** Multiplies this value by the other value. */
    public inline operator fun times(other: UByte): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    public inline operator fun times(other: UShort): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    public inline operator fun times(other: UInt): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))

    /** Divides this value by the other value. */
    public inline operator fun div(other: UByte): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    public inline operator fun div(other: UShort): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    public inline operator fun div(other: UInt): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    public inline operator fun div(other: ULong): ULong = ULong(this.data.times(other.data))

    /** Multiplies this value by the other value. */
    public inline operator fun times(other: UByte): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    public inline operator fun times(other: UShort): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    public inline operator fun times(other: UInt): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))

    /** Divides this value by the other value. */
    public inline operator fun div(other: UByte): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    public inline operator fun div(other: UShort): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    public inline operator fun div(other: UInt): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    public inline operator fun div(other: ULong): ULong = ULong(this.data.times(other.data))
/** Divides this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun div(other: ULong): ULong = ulongDivide(this, other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UByte): ULong = this.rem(other).toULong()
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UShort): ULong = this.rem(other).toULong()
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: UInt): ULong = this.rem(other).toULong()
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: Ulong): ULong = ulongRemainder(this, other)
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = this.rem(other).toULong()
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = this.rem(other).toULong()
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = this.rem(other).toULong()
/** Calculates the remainder of dividing this value by the other value. */
@kotlin.internal.InlineOnly
public inline operator fun rem(other: ULong): ULong = this.rem(other).toULong()
* * The resulting `UByte` value is represented by the least significant 32 bits of this `ULong` value.

```kotlin
public inline fun toULong(): ULong = this
```

* * **Converts this [ULong] value to [Double].**

```kotlin
public inline fun toDouble(): Double = ulongToDouble(data)
```

/**
 * Converts this 
 * value to 
 * value. 
 *
 * The resulting 
 * value is exactly between two `Double`'s, the one with zero at least significant bit of mantissa is selected.

```kotlin
@kotlin.internal.InlineOnly
public inline fun toDouble(): Double = ulongToDouble(data)
```

* * **Converts this [Byte] value to [ULong].**

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Byte.toULong(): ULong = ULong(this.toLong())
```

* * **Converts this [Short] value to [ULong].**

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Short.toULong(): ULong = ULong(this.toLong())
```

* * **Converts this [Int] value to [ULong].**

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Int.toULong(): ULong = ULong(this.toLong())
```

* * **Converts this [Long] value to [ULong].**

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Long.toULong(): ULong = ULong(this)
```

* * **Converts this [Float] value to [ULong].**

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Float.toULong(): ULong = doubleToULong(this.toDouble())
```

* * **Converts this [Double] value to [ULong].**

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun Double.toULong(): ULong = doubleToULong(this)
```

* * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Byte`.

* * **The least significant 8 bits of the resulting `ULong` value are the same as the bits of this `Byte` value.**

* * Whereas the most significant 56 bits are filled with the sign bit of this value.

* * Whereas this list contains exactly 56 bits.

* * **The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `Short` value.**

* * Whereas the most significant 48 bits are filled with the sign bit of this value.

* * **The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `Int` value.**

* * Whereas the most significant 32 bits are filled with the sign bit of this value.

* * **The least significant 64 bits of the resulting `ULong` value are the same as the bits of this `Long` value.**

* * **The fractional part, if any, is rounded down towards zero.**

* * **Returns zero if this `Float` value is negative or `NaN`,`ULong.MAX_VALUE` if it’s bigger than `ULong.MAX_VALUE`.**

* * **Returns 1st *element* from the list.**

* * **Returns 2nd *element* from the list.**

* * **Returns 3rd *element* from the list.**

* * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* * @file: kotlin.jvm.JvmMultifileClass

* * @package kotlin.collections

* * @JvmName("CollectionsKt")

* * @unpackage kotlin.collections

* * @NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt

* * See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

* * import kotlin.random.

* * import kotlin.ranges.contains

* * import kotlin.ranges.reversed

* * import kotlin.ranges.

* * **Throws an IndexOutOfBoundsException if the size of this list is less than 1.**

* * **Throws an IndexOutOfBoundsException if the size of this list is less than 2.**

* * **Throws an IndexOutOfBoundsException if the size of this list is less than 3.**

* * **Throws an IndexOutOfBoundsException if the size of this list is less than 4.**
`IndexOutOfBoundsException` if the size of this list is less than 3.

```kotlin
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component3(): T {
    return get(2)
}

/**
 * Returns 4th *element* from the list.
 * Throws an `IndexOutOfBoundsException` if the size of this list is less than 4.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component4(): T {
    return get(3)
}

/**
 * Returns 5th *element* from the list.
 * Throws an `IndexOutOfBoundsException` if the size of this list is less than 5.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component5(): T {
    return get(4)
}

/**
 * Returns the first *element* matching the given [predicate], or throws a
 * `NoSuchElementException` if the collection is empty.
 */
public fun <T> Iterable<T>.findFirst(predicate: (T) -> Boolean): T? {
    return find(predicate) ?: throw NoSuchElementException("Collection is empty.")
}
```

```kotlin
/**
 * Returns the last *element* matching the given [predicate], or throws a
 * `NoSuchElementException` if the collection is empty.
 */
public fun <T> Iterable<T>.findLast(predicate: (T) -> Boolean): T? {
    return findLast(predicate) ?: throw NoSuchElementException("Collection is empty.")
}
```

```kotlin
/**
 * Returns the last *element* matching the given [predicate], or throws a
 * `NoSuchElementException` if the collection is empty.
 */
public fun <T> List<T>.findLast(predicate: (T) -> Boolean): T? {
    return findLast(predicate) ?: throw NoSuchElementException("Collection is empty.")
}
```

```kotlin
/**
 * Returns the first *element* matching the given [predicate], or throws a
 * `NoSuchElementException` if the collection is empty.
 */
public fun <T> List<T>.findFirst(predicate: (T) -> Boolean): T? {
    return findFirst(predicate) ?: throw NoSuchElementException("Collection is empty.")
}
```

```kotlin
/**
 * Returns the first *element* matching the given [predicate], or throws a
 * `NoSuchElementException` if the collection is empty.
 */
public fun <T> List<T>.findFirstOrNull(predicate: (T) -> Boolean): T? {
    return findFirstOrNull(predicate) ?: throw NoSuchElementException("Collection is empty.")
}
```

```kotlin
/**
 * Returns the last *element* matching the given [predicate], or throws a
 * `NoSuchElementException` if the collection is empty.
 */
public fun <T> List<T>.findLastOrNull(predicate: (T) -> Boolean): T? {
    return findLastOrNull(predicate) ?: throw NoSuchElementException("Collection is empty.")
}
```
if the list is empty.

```kotlin
@throws [NoSuchElementException] if there is no element matching the given [predicate].

* * @throws [NoSuchElementException] if the given [predicate] is not found.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.

* * @throws [NoSuchElementException] if the list contains no element matching the predicate.

* * @throws [NoSuchElementException] if the list is empty.
Returns a list containing all elements that are instances of specified type parameter R.

Iterables.filterIsInstance<T>(predicate: (T) -> Boolean): List<T> { 
    val list = ArrayList<T>(size)
    if (this is RandomAccess) {
        for (index in size until size)
            list.add(this[index])
    } else {
        for (value in listIterator(size - n))
            list.add(value)
    }
    return list
}

Returns a list containing all elements that are not `null`.

Iterables.filterNotNullTo(destination: C, predicate: (T) -> Boolean): C {
    var count = 0
    val list = ArrayList<T>(n)
    for (value in this) {
        if (value != null) {
            list.add(value)
            if (++count == n)
                break
        }
    }
    return list.optimizeReadOnlyList()
}

Returns a list containing elements at specified indices.

Iterables.filterWhile(test: (T) -> Boolean): List<T> {
    var count = 0
    val list = ArrayList<T>(n)
    for (value in this) {
        if (test(value)) {
            list.add(value)
            if (++count == n)
                break
        }
    }
    return list.optimizeReadOnlyList()
}

Returns a list containing first elements satisfying the given predicate.

Iterables.filterLastWhile(test: (T) -> Boolean): List<T> {
    var count = 0
    val list = ArrayList<T>(size - n)
    for (value in this) {
        if (test(value)) {
            list.add(value)
            if (++count == n)
                break
        }
    }
    return list.optimizeReadOnlyList()
}

Returns a list containing all elements that are instances of specified type parameter R.

Iterables.filterAll<T>(predicate: (T) -> Boolean): List<T> {
    var count = 0
    val list = ArrayList<T>(size)
    for (value in this) {
        if (test(value)) {
            list.add(value)
            if (++count == n)
                break
        }
    }
    return list.optimizeReadOnlyList()
}

Returns a list containing all elements that are instances of specified type parameter R.

Iterables.filterWithDefault<T>(defaultValue: T): List<T> {
    var count = 0
    val list = ArrayList<T>(size)
    for (value in this) {
        if (test(value)) {
            list.add(value)
            if (++count == n)
                break
        }
    }
    return list.optimizeReadOnlyList()
}

Returns a list containing all elements that are instances of specified type parameter R.

Iterables.filterWithDefault<T>(defaultValue: T): List<T> {
    var count = 0
    val list = ArrayList<T>(size)
    for (value in this) {
        if (test(value)) {
            list.add(value)
            if (++count == n)
                break
        }
    }
    return list.optimizeReadOnlyList()
}

Returns a list containing all elements that are instances of specified type parameter R.

Iterables.filterWithDefault<T>(defaultValue: T): List<T> {
    var count = 0
    val list = ArrayList<T>(size)
    for (value in this) {
        if (test(value)) {
            list.add(value)
            if (++count == n)
                break
        }
    }
    return list.optimizeReadOnlyList()
}

Returns a list containing all elements that are instances of specified type parameter R.

Iterables.filterWithDefault<T>(defaultValue: T): List<T> {
    var count = 0
    val list = ArrayList<T>(size)
    for (value in this) {
        if (test(value)) {
            list.add(value)
            if (++count == n)
                break
        }
    }
    return list.optimizeReadOnlyList()
Open Source Used In DNAC Platform Antelope 9330

(T -> Boolean): List<T> { val list = ArrayList<T>() for (item in this) { if (!predicate(item)) list.add(item) } return list }/* Reverses elements in the list in-place. */

/* Reverses elements in the list in-place. */
Iterable<T>.reversed(): List<T> { for (item in this) { if (is Collection && size <= 1) return toList() val list = toMutableList() list.reversed() return list } /* Randomly shuffles elements in this list in-place using the specified [random] instance as the source of randomness. */

https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm

/* Returns a list of all elements sorted according to the specified [comparator]. */
Iterable<T>.sortedWith(comparator: Comparator<in T>): List<T> { for (item in this) { if (is Collection) { if (size <= 1) return this.toList() as Array<T>.apply { sort() } asList() } return toMutableList().apply { sort() } } return toTypedArray<Comparable<T>>().asList() } /* Returns an array of Boolean containing all of the elements of this collection. */

Collection<Boolean>.toBooleanArray(): BooleanArray { val result = BooleanArray(size) var index = 0 for (element in this) { result[index++] = element } return result } /* Returns an array of Byte containing all of the elements of this collection. */

Collection<Byte>.toByteArray(): ByteArray { val result = ByteArray(size) var index = 0 for (element in this) { result[index++] = element } return result } /* Returns an array of Char containing all of the elements of this collection. */

Collection<Char>.toCharArray(): CharArray { val result = CharArray(size) var index = 0 for (element in this) { result[index++] = element } return result } /* Returns an array of Byte containing all of the elements of this collection. */

Collection<Byte>.toByteArray(): ByteArray { val result = ByteArray(size) var index = 0 for (element in this) { result[index++] = element } return result } /* Returns an array of Char containing all of the elements of this collection. */

Collection<Char>.toCharArray(): CharArray { val result = CharArray(size) var index = 0 for (element in this) { result[index++] = element } return result }
result
}

/**
 * Returns an array of Double containing all of the elements of this collection.
 */
public fun Collection<Double>.toDoubleArray(): DoubleArray {
    val result = DoubleArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Float containing all of the elements of this collection.
 */
public fun Collection<Float>.toFloatArray(): FloatArray {
    val result = FloatArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Int containing all of the elements of this collection.
 */
public fun Collection<Int>.toIntArray(): IntArray {
    val result = IntArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Long containing all of the elements of this collection.
 */
public fun Collection<Long>.toLongArray(): LongArray {
    val result = LongArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Short containing all of the elements of this collection.
 */
public fun Collection<Short>.toShortArray(): ShortArray {
    val result = ShortArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns a [Map] containing key-value pairs provided by [transform] function applied to elements of the given collection.
 */
public inline fun <T, K, V> Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

/**
 * Returns a [Map] containing the elements from the given collection indexed by the key returned from [keySelector] function applied to each element.
 */
public inline fun <T, K> Iterable<T>.associateBy(keySelector: (T) -> K): Map<K, T> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given collection.
 */
public inline fun <T, K, V> Iterable<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given collection.
 */
public inline fun <T, K, M : MutableMap<in K, in T>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs provided by [transform] function applied to each element of the given collection.
 */
public inline fun <T, K, M : MutableMap<in K, in V>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

Open Source Used In DNAC Platform Antelope 9331
samples.collections.Collections.Transformations.associateTo

@public inline fun <T, K, V, M : MutableMap<K, V>> Iterable<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M { 
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Returns a [Map] where keys are elements from the given collection and values are produced by the [valueSelector] function applied to each element.
 * If any two elements are equal, the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 */
@sample samples.collections.Collections.Transformations.associateTo

Iterable<K>.associateTo(destination: M, valueSelector: (K) -> V): M { 
    for (item in this) {
        destination.put(item, valueSelector(item))
    }
    return destination
}

/**
 * Returns a [HashSet] of all elements.
 * The returned set preserves the element iteration order of the original collection.
 */
public fun <T> Iterable<T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>(mapCapacity(collectionSizeOrDefault(12))))
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original collection.
 * @sample samples.collections.Collections.Transformations.flatMap
 */
public inline fun <T, R> Iterable<T>.flatMap(transform: (T) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Iterable<T>.flatMap(transform: (T) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}
results of [transform] function being invoked on each element and its index in the original collection.

```kotlin
@sample samples.collections.Transformations.flatMapIndexed

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("flatMapIndexedSequence")@kotlin.internal.InlineOnly
public inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {
    return
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("flatMapIndexedIterableTo")@kotlin.internal.InlineOnly
public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(checkIndexOverflow(index++), element)
        destination.addAll(list)
    }
    return
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("flatMapIndexedSequenceTo")@kotlin.internal.InlineOnly
public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(checkIndexOverflow(index++), element)
        destination.addAll(list)
    }
    return
}

@sample samples.collections.Transformations.flatMapTo

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("flatMapSequenceTo")
public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return
}

@sample samples.collections.Transformations.groupBy

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("flatMapSequenceTo")@kotlin.internal.InlineOnly
public inline fun <T, K> Iterable<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<T>() }
        destination.setOrPut(key, list)
    }
    return
}
```

```kotlin
@sample samples.collections.Transformations.groupBy

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("flatMapSequenceTo")@kotlin.internal.InlineOnly
public inline fun <T, K, V> Iterable<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<T>() }
    }
    return
```
```kotlin
/**
 * Groups values returned by the [valueTransform] function applied to each element of the original collection
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Creates a [Grouping] source from a collection to be used later with one of group-and-fold operations
 * using a [keySelector] function to extract a key from each element,
 * and the element itself
 * returns the result of the transform applied to the element.
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * @sample samples.collections.Collections.Transformations.map
 */
public inline fun <T, R> Iterable<T>.map(transform: (T) -> R): List<R> {
    return mapIndexedTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

/**
 * Returns a list containing only the non-null results of applying the given 
 * [transform] function to each element in the original collection.
 * @param [transform] function that takes the index of an element and the element itself
 * and returns the result of the transform applied to the element.
 * @sample samples.collections.Collections.Transformations.mapNotNull
 */
public inline fun <T, R : Any> Iterable<T>.mapNotNull(transform: (T) -> R?): List<R> {
    return mapIndexedNotNullTo(ArrayList<R>(), transform)
}
```

---

**Open Source Used In DNAC Platform Antelope 9334**
destination

* Returns a lazy [Iterable] that wraps each element of the original collection into an [IndexedValue] containing the index of that element and the element itself.

* Returns a list containing only distinct elements from the given collection. * Among equal elements of the given collection, only the first one will be present in the resulting list. * The elements in the resulting list are in the same order as they were in the source collection.

** Returns a lazy [Iterable] that wraps each element of the original collection into an [IndexedValue] containing the index of that element and the element itself.

public fun <T> Iterable<T>.withIndex(): Iterable<IndexedValue<T>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a list containing only distinct elements from the given collection.
 * * Among equal elements of the given collection, only the first one will be present in the resulting list. * The elements in the resulting list are in the same order as they were in the source collection.

Iterable<T>.distinctAndDistinctBy

public fun <K> Iterable<T>.distinctBy(selector: (T) -> K): List<T> {
    val set = HashSet<K>()
    val list = ArrayList<T>()
    for (e in this) {
        val key = selector(e)
        if (set.add(key))
            list.add(e)
    }
    return list
}

Iterable<T>.intersect

public fun <T> Iterable<T>.intersect(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.retainAll(other)
    return set
}

Iterable<T>.subtract

public fun <T> Iterable<T>.subtract(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.removeAll(other)
    return set
}

Iterable<T>.toMutableSet

public fun <T> Iterable<T>.toMutableSet(): MutableSet<T> {
    return when (this) {
        is Collection<T> -> LinkedHashSet(this)
        else -> toCollection(LinkedHashSet<T>())
    }
}

Iterable<T>.union

public infix fun <T> Iterable<T>.union(other: Iterable<T>): Set<T> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

Iterable<T>.all

public inline fun <T> Iterable<T>.all(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return true
    for (element in this) if (!predicate(element)) return false
    return true
}

Iterable<T>.any

public fun <T> Iterable<T>.any(): Boolean {
    if (this is Collection) return !isEmpty()
    return iterator().hasNext()
}

Iterable<T>.any

public inline fun <T> Iterable<T>.any(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return false
    for (element in this) if (predicate(element)) return true
    return false
}

Iterable<T>.count

public fun <T> Iterable<T>.count(): Int {
    if (this is Collection) return size
    var count = 0
    for (element in this) checkCountOverflow(++count)
    return count
}

Iterable<T>.count

public inline fun <T> Collection<T>.count(): Int {
    return size
}

Iterable<T>.count

public inline fun <T> Iterable<T>.count(predicate: (T) -> Boolean): Int {
    if (this is Collection && isEmpty()) return 0
    var count = 0
    for (element in this) checkCountOverflow(++count)
    return count
}

Iterable<T>.count

public fun <T> Iterable<T>.count(predicate: (T) -> Boolean): Int {
    if (this is Collection && isEmpty()) return 0
    var count = 0
    for (element in this) if (predicate(element)) checkCountOverflow(++count)
    return count
}

Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element. * Returns the specified [initial] value if the collection is empty. * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
value.

/*
 * public inline fun <T, R> Iterable<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
 * var accumulator = initial
 * for (element in this) accumulator = operation(accumulator, element)
 * return accumulator
 *}
 */

public inline fun <T, R> Iterable<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original collection.
 * Returns the specified [initial] value if the collection is empty.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 */

/*
 * public inline fun <T, R> Iterable<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
 * var index = 0
 * var accumulator = initial
 * for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)
 * return accumulator
 *}
 */

public inline fun <T, R> Iterable<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * Returns the specified [initial] value if the list is empty.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 */

/*
 * public inline fun <T, R> List<T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {
 * var accumulator = initial
 * if (!isEmpty()) {
 * val iterator = listIterator(size)
 * while (iterator.hasPrevious()) {
 *    accumulator = operation(iterator.previous(), accumulator)
 * }
 * return accumulator
 *}
 */

public inline fun <T, R> List<T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {
    var accumulator = initial
    if (!isEmpty()) {
        val iterator = listIterator(size)
        while (iterator.hasPrevious()) {
            accumulator = operation(iterator.previous(), accumulator)
        }
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element with its index in the original list and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value, and performs the action on the element.
 */

/*
 * public fun <T, R> List<T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {
 * var index = 0
 * if (!isEmpty()) {
 * val iterator = listIterator(size)
 * while (iterator.hasPrevious()) {
 *    val index = iterator.previousIndex()
 *    accumulator = operation(index, iterator.previous(), accumulator)
 * }
 * return accumulator
 *}
 */

public fun <T, R> List<T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {
    var index = 0
    if (!isEmpty()) {
        val iterator = listIterator(size)
        while (iterator.hasPrevious()) {
            val index = iterator.previousIndex()
            accumulator = operation(index, iterator.previous(), accumulator)
        }
    }
    return accumulator
}

/**
 * Performs the given [action] on each element.
 */

@kotlin.internal.HidesMembers
public inline fun <T> Iterable<T>.forEach(action: (T) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes the index of an element and the element itself and performs the action on the element.
 */

/*
 * public inline fun <T> Iterable<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {
 * var index = 0
 * for (item in this) action(checkIndexOverflow(index++), item)
 */

public inline fun <T> Iterable<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {
    var index = 0
    for (item in this) action(checkIndexOverflow(index++), item)
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun Iterable<Double>.max(): Double? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun Iterable<Float>.max(): Float? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun <T : Comparable<T>> Iterable<T>.max(): T? {
    return maxOrNull()
}

@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun Iterable<T>.maxBy(selector: (T) -> R): T? {
    return maxByOrNull(selector)
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * @param [selector] function that takes each element in the collection and returns a value.
 */

/*
 * public fun <T, R : Comparable<R>> Iterable<T>.maxByOrNull(selector: (T) -> R): T? {
 * var maxElement = iterator.next()
 * var maxValue = selector(maxElement)
 * do {
 *    val elem = iterator.next()
 *    val v = selector(elem)
 *    if (maxValue < v) {
 *        maxElement = elem
 *        maxValue = v
 *    }
 * } while (iterator.hasNext())
 * return maxElement
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.maxOf(selector: (T) -> Double): Double {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the collection.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the collection is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.maxOf(selector: (T) -> Float): Float {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.
 * @throws NoSuchElementException if the collection is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>>
Iterable<T>.maxOf(selector: (T) -> R): R {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the collection.
 * @throws NoSuchElementException if the collection is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R> Iterable<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}
maxValue = v\n
}\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the collection or `null`
if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Iterable<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n val iterator =
iterator()\n if (!iterator.hasNext()) return null\n var maxValue = selector(iterator.next())\n while
(iterator.hasNext()) {\n
val v = selector(iterator.next())\n
if (comparator.compare(maxValue, v) < 0) {\n
maxValue = v\n
}\n }\n return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no
elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun
Iterable<Double>.maxOrNull(): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var
max = iterator.next()\n while (iterator.hasNext()) {\n
val e = iterator.next()\n
max = maxOf(max, e)\n
}\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of
elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Iterable<Float>.maxOrNull(): Float? {\n
val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while
(iterator.hasNext()) {\n
val e = iterator.next()\n
max = maxOf(max, e)\n }\n return max\n}\n\n/**\n *
Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T :
Comparable<T>> Iterable<T>.maxOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n
var max = iterator.next()\n while (iterator.hasNext()) {\n
val e = iterator.next()\n
if (max < e) max = e\n
}\n return max\n}\n\n@Deprecated(\"Use maxWithOrNull instead.\",
ReplaceWith(\"maxWithOrNull(comparator)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\")\npublic fun
<T> Iterable<T>.maxWith(comparator: Comparator<in T>): T? {\n return
maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first element having the largest value according to the
provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T>
Iterable<T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n val iterator = iterator()\n if
(!iterator.hasNext()) return null\n var max = iterator.next()\n while (iterator.hasNext()) {\n
val e =
iterator.next()\n
if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n@Deprecated(\"Use
minOrNull instead.\", ReplaceWith(\"minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince =
\"1.4\")\n@SinceKotlin(\"1.1\")\npublic fun Iterable<Double>.min(): Double? {\n return
minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\")\n@SinceKotlin(\"1.1\")\npublic
fun Iterable<Float>.min(): Float? {\n return minOrNull()\n}\n\n@Deprecated(\"Use minOrNull instead.\",
ReplaceWith(\"minOrNull()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\")\npublic fun <T :
Comparable<T>> Iterable<T>.min(): T? {\n return minOrNull()\n}\n\n@Deprecated(\"Use minByOrNull
instead.\", ReplaceWith(\"minByOrNull(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\")\npublic
inline fun <T, R : Comparable<R>> Iterable<T>.minBy(selector: (T) -> R): T? {\n return
minByOrNull(selector)\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or
`null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n
*/\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minByOrNull(selector: (T) ->
R): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minElem = iterator.next()\n if
(!iterator.hasNext()) return minElem\n var minValue = selector(minElem)\n do {\n
val e = iterator.next()\n
val v = selector(e)\n
if (minValue > v) {\n
minElem = e\n
minValue = v\n
}\n } while
(iterator.hasNext())\n return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by
[selector] function\n * applied to each element in the collection.\n * \n * If any of values produced by [selector]
function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the collection is
empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) ->

Open Source Used In DNAC Platform Antelope 9338


Double): Double {\n    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        \n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by \[selector\] function \n * applied to each element in the collection.\n * \n * If any of values produced by \[selector\] function is \texttt{NaN} \n * the returned result is \texttt{NaN}.\n * \n * @throws NoSuchElementException if the collection is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) -> Double): Double {
    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by \[selector\] function \n * applied to each element in the collection or \texttt{null} \n * if there are no elements.\n * \n * If any of values produced by \[selector\] function is \texttt{NaN}, the returned result is \texttt{NaN}.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T) -> Double): Double? {
    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by \[selector\] function \n * applied to each element in the collection or \texttt{null} \n * if there are no elements.\n * \n * If any of values produced by \[selector\] function is \texttt{NaN}, the returned result is \texttt{NaN}.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T) -> Float): Float? {
    val iterator = iterator()\n    if (!iterator.hasNext()) return null\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by \[selector\] function \n * applied to each element in the collection or \texttt{null} \n * if there are no elements.\n * \n * If any of values produced by \[selector\] function is \texttt{NaN}, the returned result is \texttt{NaN}.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minOfOrNull(selector: (T) -> R): R? {
    val iterator = iterator()\n    if (!iterator.hasNext()) throw NoSuchElementException()\n    var minValue = selector(iterator.next())\n    while (iterator.hasNext()) {\n        val v = selector(iterator.next())\n        if (minValue > v) {\n            minValue = v\n        }\n    }\n    return minValue
}
values produced by [selector] function applied to each element in the collection or `null` if there are no elements.

Iterable<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {
    val iterator = iterator() if (!iterator.hasNext()) return null
    var minValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val value = selector(iterator.next()) if (comparator.compare(minValue, value) > 0) {
            minValue = value
        }
    }
    return minValue
}

/**
 * Returns the first element having the smallest value according to the
 * provided [comparator] or `null` if there are no elements.
 */
public fun <T> Iterable<T>.minWith(comparator: Comparator<in T>): T? {
    return minWithOrNull(comparator)
}

/**
 * Returns `true` if the collection has no elements.
 */
public fun <T> Iterable<T>.none(): Boolean {
    if (this is Collection) return isEmpty()
    return !iterator().hasNext()
}

/**
 * Returns `true` if no elements match the given [predicate].
 */
public inline fun <T> Iterable<T>.none(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return true
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Performs the given [action] on each element and returns the collection itself afterwards.
 */
@SinceKotlin("1.1")
public inline fun <T, S : Iterable<T>> S.onEach(action: (T) -> Unit): S {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 */
@SinceKotlin("1.4")
public inline fun <T, C : Iterable<T>> C.onEachIndexed(action: (index: Int, T) -> Unit): C {
    return apply { forEachIndexed(action) }
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 */
public inline fun <S, T : S> Iterable<T>.reduce(operation: (acc: S, T) -> S): S {
    val iterator = this.iterator()
    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original collection.
 */
public inline fun <S, T : S> Iterable<T>.reduceIndexed(operation: (acc: S, Int, T) -> S): S {
    val iterator = this.iterator()
    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next(), iterator.next())
    }
    return accumulator
}
expected way,\n* please use [reduceIndexedOrNull] instead. It returns \texttt{null} when its receiver is empty.\n* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n* and calculates the next accumulator value.\n* @sample samples.collections.Collections.Aggregates.reduce
*\n
public inline fun <S, T : S> Iterable<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n    val iterator = this.iterator()\n    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty collection can't be reduced.")\n    var index = 0\n    var accumulator: S = iterator.next() while (iterator.hasNext()) {\n        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n    } \n    return accumulator\n}\n
\n
Accumulates value starting from the first element and applying [operation] from left to right.\n* to current accumulator value and each element with its index in the original collection.\n* \n* \n* @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n* and calculates the next accumulator value.\n* @sample samples.collections.Collections.Aggregates.reduce\n*\n
public inline fun <S, T : S> Iterable<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S?): S? {\n    val iterator = this.iterator()\n    if (!iterator.hasNext()) return null\n    var index = 0\n    var accumulator: S = iterator.next() while (iterator.hasNext()) {\n        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n    } \n    return accumulator\n}\n
Accumulates value starting with the last element and applying [operation] from right to left.\n* to current accumulator value and each element.\n* @param [operation] function that takes current accumulator value and an element,\n* and calculates the next accumulator value.\n* @sample samples.collections.Collections.Aggregates.reduceRight\n*\n
public inline fun <S, T : S> List<T>.reduceRight(operation: (T, acc: S) -> S): S {\n    val iterator = listIterator(size)\n    if (!iterator.hasPrevious())\n        throw UnsupportedOperationException("Empty list can't be reduced.")\n    var accumulator: S = iterator.previous()\n    while (iterator.hasPrevious()) {\n        accumulator = operation(iterator.previous(), accumulator)\n    } \n    return accumulator\n}\n
Accumulates value starting with the last element and applying [operation] from right to left.\n* to each element and current accumulator value.\n* @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n* and calculates the next accumulator value.\n* @sample samples.collections.Collections.Aggregates.reduceRightIndexed\n*\n
public inline fun <S, T : S> List<T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n    val iterator = listIterator(size)\n    if (!iterator.hasPrevious())\n        throw UnsupportedOperationException("Empty list can't be reduced.")\n    var index = 0\n    var accumulator: S = iterator.previous()\n    while (iterator.hasPrevious()) {\n        index = iterator.previousIndex()\n        accumulator = operation(index, iterator.previous(), accumulator)\n    } \n    return accumulator\n}\n
Accumulates value starting with the last element and applying [operation] from right to left.\n* to each element with its index in the original list and current accumulator value.\n* @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n* and calculates the next accumulator value.\n* @sample samples.collections.Collections.Aggregates.reduceRightIndexedOrNull\n*\n
public inline fun <S, T : S> List<T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S?): S? {\n    val iterator = listIterator(size)\n    if (!iterator.hasPrevious())\n        throw UnsupportedOperationException("Empty list can't be reduced.")\n    var index = 0\n    var accumulator: S = iterator.previous()\n    while (iterator.hasPrevious()) {\n        index = iterator.previousIndex()\n        accumulator = operation(index, iterator.previous(), accumulator)\n    } \n    return accumulator\n}
listIterator(size)
if (!iterator.hasPrevious())
    return null
var accumulator: S = iterator.previous()
while (iterator.hasPrevious()) {
    val index = iterator.previousIndex()
    accumulator = operation(index, iterator.previous(), accumulator)
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value.
 * Returns `null` if the list is empty.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class) public inline fun <S, T : S> List<T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {
    val iterator = listIterator(size)
    if (!iterator.hasPrevious())
        return null
    var accumulator: S = iterator.previous()
    while (iterator.hasPrevious()) {
        accumulator = operation(iterator.previous(), accumulator)
    }
    return accumulator
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")
public inline fun <T, R> Iterable<T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {
    val estimatedSize = collectionSizeOrDefault(9)
    if (estimatedSize == 0) return listOf(initial)
    val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original collection and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFoldIndexed
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class) public inline fun <T, R> Iterable<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {
    val estimatedSize = collectionSizeOrDefault(9)
    if (estimatedSize == 0) return listOf(initial)
    val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }
    var index = 0
    var accumulator = initial
    for (element in this) {
        accumulator = operation(index++, accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this collection.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class) public inline fun <S, T : S> Iterable<T>.runningReduce(operation: (acc: S, T) -> S): List<S> {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return emptyList()
    var accumulator: S = iterator.next()
    val result = ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}
Since Kotlin 1.4
@SinceKotlin("1.4")
public inline fun <T> Iterable<T>.runningReduceIndexed(operation: (index: Int, acc: T, T) -> T): List<T> {
    var accumulator = this.first()
    val result = ArrayList<T>(2)

    var index = 0
    while (index < this.size) {
        accumulator = operation(index++, accumulator, this[index++])
        result.add(accumulator)
        index = 0
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element in the collection.
 * 
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * 
 * @see samples.collections.Collections.Aggregates.sumOf
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("runningReduceIndexed")
public inline fun <T> Iterable<T>.runningReduceIndexed(operation: (index: Int, acc: T, T) -> T): List<T> {
    var accumulator = this.first()
    val result = ArrayList<T>(2)

    var index = 0
    while (index < this.size) {
        accumulator = operation(index++, accumulator, this[index++])
        result.add(accumulator)
        index = 0
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element in the collection.
 * 
 * @param [operation] function that takes an element and current accumulator value that starts with [initial] value.
 * 
 * @see samples.collections.Collections.Aggregates.sumBy
 */
@SinceKotlin("1.4")
public inline fun <T> Iterable<T>.sumBy(selector: (T) -> Int): List<T> {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the collection.
 * 
 * @param [selector] function that takes an element and current accumulator value that starts with [initial] value.
 * 
 * @see samples.collections.Collections.Aggregates.sumOf
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("runningReduce")
public inline fun <T> Iterable<T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the collection.
 * 
 * @param [selector] function that takes an element and current accumulator value that starts with [initial] value.
 * 
 * @see samples.collections.Collections.Aggregates.sumOf
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("runningReduce")
public inline fun <T> Iterable<T>.sumOf(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the collection.
 * 
 * @param [selector] function that takes an element and current accumulator value that starts with [initial] value.
 * 
 * @see samples.collections.Collections.Aggregates.sumOf
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("runningReduce")
public inline fun <T> Iterable<T>.sumOfLong(selector: (T) -> Long): Long {
    var sum: Long = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

@Deprecated("Use runningReduceIndexed instead.", level = DeprecationLevel.WARN, ReplaceWith("runningReduceIndexed(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public inline fun <T> Iterable<T>.scanIndexed(initial: T, operation: (index: Int, acc: T, T) -> T): List<T> {
    var accumulator = initial
    val result = ArrayList<T>(2)

    var index = 0
    while (index < this.size) {
        accumulator = operation(index++, accumulator, this[index++])
        result.add(accumulator)
        index = 0
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element in the collection.
 * 
 * @param [operation] function that takes an element and current accumulator value that starts with [initial] value.
 * 
 * @see samples.collections.Collections.Aggregates.sumBy
 */
@SinceKotlin("1.4")
public inline fun <T> Iterable<T>.scan(initial: T, operation: (acc: T, T) -> T): List<T> {
    var accumulator = initial
    val result = ArrayList<T>(2)

    while (accumulator != null) {
        result.add(accumulator)
        accumulator = operation(accumulator, this[0])
        this.removeAt(0)
    }
    return result
}

@Deprecated("Use scanIndexed instead.", level = DeprecationLevel.WARN, ReplaceWith("scanIndexed(initial, operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public inline fun <T> Iterable<T>.scan(initial: T, operation: (acc: T, T) -> T): List<T> {
    var accumulator = initial
    val result = ArrayList<T>(2)

    while (accumulator != null) {
        result.add(accumulator)
        accumulator = operation(accumulator, this[0])
        this.removeAt(0)
    }
    return result
}

@Deprecated("Use scanIndexed instead.", level = DeprecationLevel.WARN, ReplaceWith("scanIndexed(initial, operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public inline fun <T> Iterable<T>.scanIndexed(initial: T, operation: (index: Int, acc: T, T) -> T): List<T> {
    var accumulator = initial
    val result = ArrayList<T>(2)

    var index = 0
    while (accumulator != null) {
        result.add(accumulator)
        accumulator = operation(index++, accumulator, this[0])
        this.removeAt(0)
    }
    return result
}

@Deprecated("Use scanReduceIndexed instead.", level = DeprecationLevel.WARN, ReplaceWith("scanReduceIndexed(initial, operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public inline fun <T> Iterable<T>.scanReduce(initial: T, operation: (acc: T, T) -> T): List<T> {
    var accumulator = initial
    val result = ArrayList<T>(2)

    while (accumulator != null) {
        result.add(accumulator)
        accumulator = operation(accumulator, this[0])
        this.removeAt(0)
    }
    return result
}

@Deprecated("Use scanReduceIndexed instead.", level = DeprecationLevel.WARN, ReplaceWith("scanReduceIndexed(initial, operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
public inline fun <T> Iterable<T>.scanReduceIndexed(initial: T, operation: (index: Int, acc: T, T) -> T): List<T> {
    var accumulator = initial
    val result = ArrayList<T>(2)

    var index = 0
    while (accumulator != null) {
        result.add(accumulator)
        accumulator = operation(index++, accumulator, this[0])
        this.removeAt(0)
    }
    return result
}
sum += selector(element)

return sum

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.

* Returns the sum of all values produced by [selector] function applied to each element in the collection.
HashSet to speed up the operation, thus the elements are required to have a correct and stable implementation of `hashCode()` that doesn’t change between successive invocations.

```kotlin
public operator fun <T> Iterable<T>.minus(elements: Sequence<T>): List<T> {
    val other = elements.toHashSet()
    if (other.isEmpty())
        return this.toList()
    return this.filterNot { it in other }
}

/**
 * Returns a list containing all elements of the original collection without the first occurrence of the given [element].
 */
public operator fun <T> Iterable<T>.plus(element: T): List<T> {
    if (this is Collection) return this.plus(element)
    val result = ArrayList<T>()
    result.addAll(this)
    result.add(element)
    return result
}

/**
 * Returns a list containing all elements of the original collection and then the given [element].
 */
public operator fun <T> Collection<T>.plus(element: T): List<T> {
    val result = ArrayList<T>(size + 1)
    result.addAll(this)
    result.add(element)
    return result
}

/**
 * Returns a list of snapshots of the window of the given [size]
 * sliding along this collection with the given [step], where each snapshot is a list.
 */
public operator fun <T> Collection<T>.plus(elements: Sequence<T>): List<T> {
    val result = ArrayList<T>(size + 10)
    result.addAll(this)
    result.addAll(elements)
    return result
}
```

Open Source Used In DNAC Platform Antelope 9345
samples.collections.Sequences.Transformations.takeWindows

```kotlin
{ n
checkWindowSizeStep(size, step) if (this is RandomAccess & & this is List) { n val thisSize = this.size
val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1 n val result = ArrayList<T>(resultCapacity) n var index = 0 n while (index in 0 until thisSize) { n val windowSize = size.coerceAtMost(thisSize - index) n if (windowSize < size & & partialWindows) break n result.add(WindowList(windowSize, this[index+index]) n index += step} n return result
} n return this
}

/**
 * Returns a list of results of applying the given [transform] function to
 * an each list representing a view over the window of the given [size] sliding
 * along this collection with the given [step].
 * Note that the list passed to the [transform] function is ephemeral
 * and is valid only inside that function. You should not store it or allow it to escape in some way, unless you made
 * a snapshot of it. Several last lists may have less elements than the given [size].
 * Both [size] and [step] must be positive and can be greater than the number of elements in this collection.
 * @param size the number of elements to take in each window
 * @param step the number of elements to move the window forward by on each step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * * by default `false` which means partial windows won't be preserved
 * * @sample
 * */
{ n checkWindowSizeStep(size, step)
if (this is RandomAccess & & this is List) { n val thisSize = this.size
val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1 n val result = ArrayList<R>(resultCapacity) n val windowSize = size.coerceAtMost(thisSize - index) n if (windowSize < size & & partialWindows) break n result.add(transform(WindowList(windowSize, this[index+index]))) n index += step}
return result
} n result = ArrayList<R>()
val result = windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = false).forEach {
result.add(transform(it))
}
return result
}

/**
 * Returns a list of pairs built from the elements of `this` collection and the [other] array with the same index.
 * The returned list has length of the shortest collection. The returned list has length of the shortest collection.
 * @sample
 * */
public infix fun <T, R> Iterable<T>.zip(other: Array<out R>): List<Pair<T, R>>
{ n return this.zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of values built from the elements of `this` collection and the [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample
 * */
{ n first = iterator() n var i = 0 n for (element in this) {
if (i >= arraySize) break
list.add(transform(element, other[i++]))}
return list
} n return this.zip(other) { t1, t2 -> t1 to t2 }

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] collection with the same index.
 * The returned list has length of the shortest collection.
 * @sample
 * */
public infix fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>>
{ n return this.zip(other) { t1, t2 -> t1 to t2 }
}
```

Open Source Used In DNAC Platform Antelope 9346
Open Source Used In DNAC Platform Antelope

```kotlin
public fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> {
    return zipWithNext { a, b -> a to b }
}

/**
 * Returns a list containing the results of applying the given [transform] function to an pair of two adjacent elements in this collection.
 * The returned list is empty if this collection contains less than two elements.
 */
public fun <T> Iterable<T>.zipWithNextToFindDeltas(): List<T> {
    return zipWithNext( { a, b -> a to b } )
}

/**
 * Returns a list containing the results of applying the given [transform] function to an each pair of two adjacent elements in this collection.
 * The returned list is empty if this collection contains less than two elements.
 */
public fun <T> Iterable<T>.zipWithNext(transform: (a: T, b: T) -> T): List<T> {
    val iterator = iterator()
    if (!iterator.hasNext()) return emptyList()
    val result = mutableListOf<T>()
    var current = iterator.next()
    while (iterator.hasNext()) {
        val next = iterator.next()
        result.add(transform(current, next))
        current = next
    }
    return result
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...")
 */
public fun <T, A : Appendable> Iterable<T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1) buffer.append(separator)
        if (limit < 0 || count <= limit) {
            buffer.appendElement(element, transform)
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}
```

```kotlin
/**
 * Returns a list containing the results of applying the given [transform] function to an each pair of two adjacent elements in this collection.
 * The returned list is empty if this collection contains less than two elements.
 */
public fun <T> Iterable<T>.zipWithNext(transform: (a: T, b: T) -> T): List<T> {
    val iterator = iterator()
    if (!iterator.hasNext()) return emptyList()
    val result = mutableListOf<T>()
    var current = iterator.next()
    while (iterator.hasNext()) {
        val next = iterator.next()
        result.add(transform(current, next))
        current = next
    }
    return result
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...")
 */
public fun <T, A : Appendable> Iterable<T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1) buffer.append(separator)
        if (limit < 0 || count <= limit) {
            buffer.appendElement(element, transform)
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}
```

```kotlin
/**
 * Returns an average value of elements in the collection.
 */
public fun Iterable<Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the collection.
 */
public fun Iterable<Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the collection.
 */
public fun Iterable<Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the collection.
 */
public fun Iterable<Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the collection.
 */
public fun Iterable<Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}
```
@kotlin.jvm.JvmName("averageOfDouble")
public fun Iterable<Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Iterable<Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Iterable<Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Iterable<Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Iterable<Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Iterable<Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Iterable<Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the
 * Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

import kotlin.comparisons.naturalOrder
import kotlin.random.Random

/** Returns the array if it's not `null`, or an
 * empty array otherwise. */
@kotlin.internal.InlineOnly
public actual inline fun <T> Array<out T>?.orEmpty(): Array<out T> = this ?: emptyArray<T>()

@kotlin.internal.InlineOnly
public actual inline fun <T> Collection<T>.toTypedArray(): Array<T> =
    copyToArray(this)

@JsName("copyToArray")
@PublishedApi
internal fun <T> copyToArray(collection: Collection<T>): Array<T> {
    return if (collection.asDynamic().toArray !== undefined)
            collection.asDynamic().toArray().unsafeCast<Array<T>>()
    else
            copyToArrayImpl(collection).unsafeCast<Array<T>>()
}

@JsName("copyToArrayImpl")
internal actual fun <T> copyToArrayImpl(collection: Collection<*>): Array<Any?> {
    val array = emptyArray<Any?>()
    val iterator = collection.iterator()
    while (iterator.hasNext())
        array.asDynamic().push(iterator.next())
    return array
}

@JsName("copyToExistingArrayImpl")
internal actual fun <T> copyToArrayImpl(collection: Collection<*>, array: Array<T>): Array<T> {
    if (array.size < collection.size)
        return copyToArrayImpl(collection).unsafeCast<Array<T>>()

    val iterator = collection.iterator()
    var index = 0
    while (iterator.hasNext()) {
        array[index++] = iterator.next().unsafeCast<T>()
    }
    if (index < array.size)
        { array[index] = null.unsafeCast<T>() }
    return array
}

/**
 * Returns an immutable list
 * containing only the specified object [element].
 */
public fun <T> listOf(element: T): List<T> =
    arrayListOf(element)

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal actual inline fun <E> buildListInternal(builderAction: MutableList<E>.() -> Unit): List<E> {
    return ArrayList<E>().apply(builderAction).build()
}

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal actual inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit): List<E> {
    return ArrayList<E>().apply(builderAction).build()
}

/**
 * Returns an immutable set containing only the
 * specified object [element].
 */
public fun <T> setOf(element: T): Set<T> =
    hashSetOf(element)

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal actual inline fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E> {
    return HashSet<E>().apply(builderAction).build()
}

/**
 * Returns an immutable set containing only the
 * specified object [element].
 */
public fun <T> setOf(element: T): Set<T> =
    hashSetOf(element)
MutableSet<E>(unit) -> Unit: Set<E> {
    return
}

/**
 * Returns an immutable map, mapping only the specified key to the
 * specified value.
 */
mapOf(pair: Pair<K, V>): Map<K, V> = hashMapOf(pair: K, V) =

MutableSet<E>(capacity).apply(builderAction).build()

/**
 * Returns an immutable map, mapping only the specified key to the
 * specified value.
 */
public fun <K, V> mapOf(pair: Pair<K, V>): Map<K, V>

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal actual inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {
    return LinkedHashMap<K, V>().apply(builderAction).build()
}

/**
 * Returns an immutable map, mapping only the specified key to the
 * specified value.
 */
public fun <K, V> mapOf(pair: Pair<K, V>): Map<K, V> =

MutableList<T>.fill(value: T): Unit {
    for (index in 0..lastIndex) {
        this[index] = value
    }
}

/**
 * Randomly shuffles elements in this list.
 */
public actual fun <T> MutableList<T>.shuffle(): Unit = shuffle(Random)

/**
 * Returns a new list with the elements of this list randomly shuffled.
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.shuffled(): List<T> = toMutableList().apply { shuffle() }

/**
 * Fills the list with the provided [value].
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.fill(value: T): Unit {
    for (index in 0..lastIndex) {
        this[index] = value
    }
}

/**
 * Sorts elements in the list in-place according to their natural sort order.
 */
@SinceKotlin("1.2")
public actual fun <T : Comparable<T>> MutableList<T>.sort(): Unit

/**
 * Sorts elements in the list in-place according to the order
 * specified with [comparator].
 */
@SinceKotlin("1.2")
public actual fun <T> MutableList<T>.sortWith(comparator: Comparator<in T>): Unit

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal actual inline fun <T> Array<out T>.copyToArrayOfAny(isVarargs: Boolean): Array<out Any?> =
    if (isVarargs)
    // no need to copy vararg
        this
    else
        this.copyOf()

/**
 * Checks a collection builder function capacity argument.
 */
@PublishedApi
internal actual fun checkIndexOverflow(index: Int) {
    if (index < 0) {
        throwIndexOverflow()
    }
    return index
}

/**
 * Checks a collection builder function capacity argument.
 */
@PublishedApi
internal actual fun checkCountOverflow(count: Int) {
    if (count < 0) {
        throwCountOverflow()
    }
    return count
}
openSourceUsedInDNACPlatformAntelope 9350

@SinceKotlin("1.3")
@ExperimentalStdlibApi
@PublishedApi
internal fun checkBuilderCapacity(capacity: Int) {
    require(capacity >= 0) { "capacity must be non-negative." }
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CollectionsKt")

package kotlin.collections

/**
 * Returns the given iterator itself. This allows to use an instance of iterator in a `for`
 * loop.
 * @param samples: A sample
 * @param iterator: An iterator
 * @return The iterator itself
 */
@kotlin.internal.InlineOnly
public operator fun <T> Iterator<T>.iterator(): Iterator<T> = this

/**
 * Returns an [IndexedValue] containing the index of that element and the element itself.
 * @param samples: A sample
 * @param iterator: An iterator
 * @param withIndex: A function
 * @return An [IndexedValue] containing the index of that element and the element itself
 */
public fun <T> Iterator<T>.withIndex(): Iterator<IndexedValue<T>> = IndexingIterator(this)

/**
 * Performs the given [operation] on each element of this [Iterator].
 * @param samples: A sample
 * @param iterator: An iterator
 * @param operation: A function
 * @return The result of applying [operation] to each element
 */
public inline fun <T> Iterator<T>.forEach(operation: (T) -> Unit): Unit {
    for (element in this) operation(element)
}

/**
 * An [Iterator] transforming original `iterator` into iterator of [IndexedValue], counting index from zero.
 * @param samples: A sample
 * @param iterator: An iterator
 * @return An [Iterator] counting index from zero
 */
private class IndexingIterator<out T>(private val iterator: Iterator<T>) : Iterator<IndexedValue<T>> {
    private var index = 0
    final override fun hasNext(): Boolean = iterator.hasNext()
    final override fun next(): IndexedValue<T> = IndexedValue(checkIndexOverflow(index++), iterator.next())
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("ComparisonsKt")

package kotlin.comparisons

import kotlin.random.*

/**
 * Returns the greater of two values.
 * If values are equal, returns the first one.
 */
@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> maxOf(a: T, b: T): T

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Byte, b: Byte): Byte

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Short, b: Short): Short

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Int, b: Int): Int

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Long, b: Long): Long

/**
 * Returns the greater of two values.
 * If either value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Float, b: Float): Float

/**
 * Returns the greater of two values.
 * If either value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Double, b: Double): Double

/**
 * Returns the greater of three values.
 * If there are multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.1")
public fun <T> maxOf(a: T, b: T, c: T): T {
    return
}
maxOf(a, maxOf(b, c, comparator), comparator)\n} \n\n\n* Returns the greater of two values according to the order specified by the given [comparator]. \n\n* If values are equal, returns the first one.\n* \n@Since Kotlin("1.1")\npublic fun <T> maxOf(a: T, b: T, comparator: Comparator\$T>\n): T \n{ \n    return if \n      (comparator.compare(a, b) \n      \n\n* Returns the greater of the given values. \n* \n* If there are multiple equal maximal values, returns the first of them. \n* \n@Since Kotlin("1.1")\npublic expect fun <T : Composable\$T>\n* maxOf(a: T, vararg other: T): T\n\n* Returns the greater of the given values. \n* \n* Returns the greater of the given values. \n* \n@Since Kotlin("1.1")\npublic expect fun maxOf(a: Byte, vararg other: Byte): Byte\n\n\n* Returns the greater of the given values. \n* \n@Since Kotlin("1.1")\npublic expect fun maxOf(a: Short, vararg other: Short): Short\n\n\n* Returns the greater of the given values. \n* \n@Since Kotlin("1.1")\npublic expect fun maxOf(a: Int, vararg other: Int): Int\n\n\n* Returns the greater of the given values. \n* \n@Since Kotlin("1.1")\npublic expect fun maxOf(a: Long, vararg other: Long): Long\n\n* Returns the greater of the given values. \n* \n* If any value is `NaN`, returns `NaN`. \n* \n@Since Kotlin("1.1")\npublic expect fun maxOf(a: Float, vararg other: Float): Float\n\n\n* Returns the greater of the given values. \n* \n@Since Kotlin("1.1")\npublic expect fun maxOf(a: Double, vararg other: Double): Double\n\n\n* Returns the greater of the given values according to the order specified by the given [comparator]. \n\n* \n@Since Kotlin("1.1")\npublic expect fun <T : Comparable\$T>\n* minOf(a: T, b: T): T\n\n* Returns the smaller of two values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Byte, b: Byte): Byte\n\n\n* Returns the smaller of two values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Short, b: Short): Short\n\n\n* Returns the smaller of two values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Int, b: Int): Int\n\n\n* Returns the smaller of two values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Long, b: Long): Long\n\n\n* Returns the smaller of two values. \n* \n* If either value is `NaN`, returns `NaN`. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Float, b: Float): Float\n\n\n* Returns the smaller of two values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Double, b: Double): Double\n\n\n* Returns the smaller of three values. \n* \n* If there are multiple equal minimal values, returns the first of them. \n* \n@Since Kotlin("1.1")\npublic expect fun <T : Composable\$T>\n* minOf(a: T, b: T, c: T): T\n\n* Returns the smaller of three values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Byte, b: Byte, c: Byte): Byte\n\n\n* Returns the smaller of three values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Short, b: Short, c: Short): Short\n\n\n* Returns the smaller of three values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Int, b: Int, c: Int): Int\n\n* Returns the smaller of three values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Long, b: Long, c: Long): Long\n\n* Returns the smaller of three values. \n* \n* If any value is `NaN`, returns `NaN`. \n* \n@Since Kotlin("1.1")\npublic expect inline expect fun minOf(a: Float, b: Float, c: Float): Float\n\n\n* Returns the smaller of three values. \n* \n@Since Kotlin("1.1")\npublic expect inline fun minOf(a: Double, b: Double, c: Double): Double\n\n* Returns the smaller of three values according to the order specified by the given [comparator]. \n* \n@Since Kotlin("1.1")\npublic expect fun <T : Composable\$T>\n* minOf(a: T, b: T, c: T, comparator: Comparator\$T>\n): T\n\n* Returns the smaller of two values according to the order specified by the given [comparator]. \n* \n@Since Kotlin("1.1")\npublic expect fun <T : Composable\$T>\n* minOf(a: T, b: T, comparator: Comparator\$T>\n): T\n\n* Returns the smaller of two values according to the order specified by the given [comparator]. \n* \n@Since Kotlin("1.1")\npublic expect fun <T : Composable\$T>\n* minOf(a: T, b: T, comparator: Comparator\$T>\n): T\n\n* Returns the smaller of two values according to the order specified by the given [comparator]. \n* \n@Since Kotlin("1.1")\npublic expect fun <T : Composable\$T>\n* minOf(a: T, b: T, comparator: Comparator\$T>\n): T
(comparator.compare(a, b) <= 0) a else b

/**
 * Returns the smaller of the given values.
 * If there are multiple equal minimal values, returns the first of them.
 */
@SinceKotlin("1.4")
public expect fun <T : Comparator<in T>> minOf(a: T, vararg other: T): T

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
public expect fun minOf(a: Byte, vararg other: Byte): Byte

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
public expect fun minOf(a: Short, vararg other: Short): Short

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
public expect fun minOf(a: Int, vararg other: Int): Int

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
public expect fun minOf(a: Long, vararg other: Long): Long

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
public expect fun minOf(a: Float, vararg other: Float): Float

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
public expect fun minOf(a: Double, vararg other: Double): Double

/**
 * Returns the smaller of the given values according to the order specified by the given [comparator].
 */
@SinceKotlin("1.4")
public fun <T> minOf(a: T, vararg other: T, comparator: Comparator<in T>): T {
    var min = a
    for (e in other) if (comparator.compare(min, e) > 0) min = e
    return min
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("MapsKt")

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns a [List] containing all key-value pairs.
 */
public fun <K, V> Map<out K, V>.toList(): List<Pair<K, V>> {
    if (size == 0)
        return emptyList()

    val iterator = entries.iterator()
    if (!iterator.hasNext())
        return emptyList()
    val first = iterator.next()
    if (!iterator.hasNext())
        return listOf(first.toPair())
    val result = ArrayList<Pair<K, V>>(size)

    result.add(first.toPair())
    do {
        result.add(iterator.next().toPair())
    } while (iterator.hasNext())
    return result
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each entry of original map.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("flatMapSequence")
public inline fun <K, V, R> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> Sequence<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each entry of original map, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapSequenceTo")
public inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) -> Sequence<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Returns a list containing the results of applying the given [transform] function being invoked on each entry of original map.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("mapToList")
public inline fun <K, V, R> Map<out K, V>.map(transform: (Map.Entry<K, V>) -> R): List<R> {
    return mapTo(ArrayList<R>(size), transform)
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("MapsKt")

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

*/
Returns a list containing only the non-null results of applying the given [transform] function to each entry in the original map.

public inline fun <K, V, R : Any> Map<out K, V>.mapNotNull(transform: (Map.Entry<K, V>) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(), transform)
}

Returns a list containing only the non-null results of applying the given [transform] function to each entry in the original map and appends only the non-null results to the given [destination].

Returns the largest value between the given [predicate] function and the largest value in the map.

Returns the number of entries matching the given [predicate] function.

Performing the given [action] on each entry.

maxOf(selector: (Map.Entry<K, V>) -> Double): Double

Returns the largest value among all values produced by [selector] function applied to each entry in the map.

maxOf(selector: (Map.Entry<K, V>) -> Float): Float

maxOf(selector: (Map.Entry<K, V>) -> Boolean): Double

maxOf(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>?
ByLambdaReturnType

@optIn(kotlin.experimental.ExperimentalTypeInference::class)
public inline fun <K, V> Map<K, V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Double?): Double? {
    return entries.maxOfOrNull(selector)
}

/**
 * Returns the largest value among all values produced by [selector]
 * function applied to each entry in the map or `null` if there are no entries.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@optIn(kotlin.experimental.ExperimentalTypeInference::class)
public inline fun <K, V> Map<K, V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Float?): Float? {
    return entries.maxOfOrNull(selector)
}

/**
 * Returns the largest value among all values produced by [selector]
 * function applied to each entry in the map or `null` if there are no entries.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@optIn(kotlin.experimental.ExperimentalTypeInference::class)
public inline fun <K, V> Map<K, V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {
    return entries.maxOfOrNull(selector)
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector]
 * function applied to each entry in the map. If there are no entries, the returned result is `null`.
 */
@optIn(kotlin.experimental.ExperimentalTypeInference::class)
    return entries.maxWith(comparator)
}

/**
 * Returns the smallest value among all values produced by [selector]
 * function applied to each entry in the map or `null` if there are no entries.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@optIn(kotlin.experimental.ExperimentalTypeInference::class)
public inline fun <K, V> Map<K, V>.minOf(selector: (Map.Entry<K, V>) -> R?): R? {
    return entries.minOf(selector)
}

/**
 * Returns the smallest value among all values produced by [selector]
 * function applied to each entry in the map. If there are no entries, the returned result is `null`.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@optIn(kotlin.experimental.ExperimentalTypeInference::class)
    return entries.minOf(selector)
}

/**
 * Returns the first entry having the smallest value among all values produced
 * by [selector] function applied to each entry in the map.
 * If any of values produced by [selector] function is `NaN`, the returned result is `null`.
 */
@optIn(kotlin.experimental.ExperimentalTypeInference::class)
    return entries.minBy(selector)
}

/**
 * Returns the first entry having the smallest value among all values produced
 * by [selector] function applied to each entry in the map.
 * If any of values produced by [selector] function is `NaN`, the returned result is `null`.
 * If any of values produced by [selector] function is `NaN`, the returned result is `null`.
 */
@optIn(kotlin.experimental.ExperimentalTypeInference::class)
public inline fun <K, V> Map<K, V>.minByOrNull(selector: (Map.Entry<K, V>) -> R?): R? {
    return entries.minByOrNull(selector)
}
/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each entry in the map.
 * 
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * 
 * @throws NoSuchElementException if the map is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun <K, V> Map<out K, V>.minOf(selector: (Map.Entry<K, V>) -> Double): Double
    = entries.minOf(selector)

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each entry in the map.
 * 
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * 
 * @throws NoSuchElementException if the map is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun <K, V> Map<out K, V>.minOf(selector: (Map.Entry<K, V>) -> Float): Float
    = entries.minOf(selector)

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each entry in the map.
 * 
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * 
 * @throws NoSuchElementException if the map is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun <K, V> Map<out K, V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double?
    = entries.minOfOrNull(selector)

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each entry in the map.
 * 
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * 
 * @throws NoSuchElementException if the map is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun <K, V, R : Comparable<R>> Map<out K, V>.minOfOrNull(selector: (Map.Entry<K, V>) -> R): R?
    = entries.minOfOrNull(selector)

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each entry in the map.
 * 
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * 
 * @throws NoSuchElementException if the map is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
    = minWithOrNull(comparator)

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each entry in the map.
 * 
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * 
 * @throws NoSuchElementException if the map is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
    = minWithWithOrNull(comparator, selector)

/**
 * Returns the first entry having the smallest value according to the provided [comparator]
 * or `null` if there are no entries.
 */
public fun <K, V> Map<out K, V>.none(): Boolean
    = minWithOrNull(comparator)
public inline fun <T> Sequence<T>.first(predicate: (T) -> Boolean): T {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Sequence contains no element matching the predicate.")
}

/**
* Returns the first element, or `null` if the sequence is empty.
* The operation is _terminal_.
*/
public fun <T> Sequence<T>.firstOrNull(): T? {
    val iterator = iterator()
    if (!iterator.hasNext())
        return null
    return iterator.next()
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was
 * not found.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.firstOrNull(predicate: (T) -> Boolean): T? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns first index of [element], or -1 if the sequence does not contain [element].
 * The operation is _terminal_.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {
    var index = 0
    for (item in this) {
        if (item == element)
            return index
        index++
    }
    return -1
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the sequence does not
 * contain such element.
 * The operation is _terminal_.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.lastIndexOf(element: T): Int {
    var lastIndex = -1
    var index = 0
    for (item in this) {
        if (item == element)
            lastIndex = index
        index++
    }
    return lastIndex
}

/**
 * Returns the last element.
 * @throws NoSuchElementException if the sequence is empty.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.last(): T {
    val iterator = iterator()
    if (!iterator.hasNext())
        throw NoSuchElementException("Sequence is empty.")
    var last = iterator.next()
    while (iterator.hasNext())
        last = iterator.next()
    return last
}

/**
 * Returns the last element matching the given [predicate].
 * @throws NoSuchElementException if no such element is found.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.lastOrNull(predicate: (T) -> Boolean): T? {
    var last: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            last = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Sequence contains no element matching the predicate.")
    return last
}

/**
 * Returns the single element, or throws an exception if the sequence is empty or has more than one element.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.single(): T {
    val iterator = iterator()
    if (!iterator.hasNext())
        throw NoSuchElementException("Sequence is empty.")
    val single = iterator.next()
    if (iterator.hasNext())
        throw IllegalArgumentException("Sequence has more than one element.")
    return single
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching
 * element.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.singleOrNull(predicate: (T) -> Boolean): T? {
    var single: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Sequence contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Sequence contains no element matching the predicate.")
    return single
}
contains no element matching the predicate.

```kotlin
@Suppress("UNCHECKED_CAST")
return single as
```

@throws IllegalArgumentException if [n] is negative.

```kotlin
<@kotlin.internal.NoInfer R>
@Suppress("UNCHECKED_CAST")
define("DNAC")

```
Sequence<T>.filterNotNullTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this) if (element != null) destination.add(element)
    return destination
}

/**
 * Appends all elements that are not `null` to the given [destination].
 * The operation is _intermediate_ and _stateful_.
 * The operation is _intermediate_ and _stateful_.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.take(n: Int): Sequence<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return when {
        n == 0 -> emptySequence()
        this is DropTakeSequence -> this.take(n)
        else -> TakeSequence(this, n)
    }
}

/**
 * Returns a sequence containing first elements satisfying the given [predicate].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * The operation is _intermediate_ and _stateful_.
 * @sample samples.collections.Collections.Sorting.sortedBy
 */
public inline fun <T, R : Comparable<R>> Sequence<T>.sortedBy(crossinline selector: (T) -> R?): Sequence<T> {
    return sortedWith(compareBy(selector))
}

/**
 * Returns a sequence that yields elements of this sequence sorted according to the specified [comparator].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * The operation is _intermediate_ and _stateful_.
 * @sample samples.collections.Collections.Transformations.sort
 */
public fun <T : Comparable<T>> Sequence<T>.sortedDescending(): Sequence<T> {
    return sortedWith(reverseOrder())
}

/**
 * Returns a sequence containing first elements not matching the given [predicate].
 * The operation is _intermediate_ and _stateful_.
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)
}

/**
 * Returns a sequence containing first elements satisfying the given [predicate].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * The operation is _intermediate_ and _stateful_.
 */
public fun <T : Comparable<T>> Sequence<T>.sortedDescending(): Sequence<T> {
    return sortedWith(reverseOrder())
}

/**
 * Returns a sequence containing first elements not matching the given [predicate].
 * The operation is _intermediate_ and _stateful_.
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)
}

/**
 * Returns a sequence containing first elements not matching the given [predicate].
 * The operation is _intermediate_ and _stateful_.
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)
}

/**
 * Returns a sequence containing first elements not matching the given [predicate].
 * The operation is _intermediate_ and _stateful_.
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)
}

/**
 * Returns a sequence containing first elements not matching the given [predicate].
 * The operation is _intermediate_ and _stateful_.
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)
}

/**
 * Returns a sequence containing first elements not matching the given [predicate].
 * The operation is _intermediate_ and _stateful_.
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)
}
Sequence<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> { return associateTo(LinkedHashMap<K, V>(), transform) } // Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given sequence. The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_ and * @sample samples.collections.Collections.Transformations.associateBy

Sequence<T>.associateBy(keySelector: (T) -> K): Map<K, T> { return associateByTo(LinkedHashMap<K, V>(), keySelector) } // Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given sequence. The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_ and * @sample samples.collections.Collections.Transformations.associateByWithValueTransform

Sequence<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> { return associateByTo(LinkedHashMap<K, V>(), keySelector, valueTransform) } // Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given sequence. The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_ and * @sample samples.collections.Collections.Transformations.associateByTo

Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K): M { // destination.put(keySelector(element), element) return destination } // Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function applied to each element of the given sequence. The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_ and * @sample samples.collections.Collections.Transformations.associateByTo

Sequence<T>.associateByToWithValueTransform(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M { // destination.put(keySelector(element), valueTransform(element)) return destination } // Returns a [Map] where keys are elements from the given sequence and values are produced by the [valueSelector] function applied to each element. The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_ and * @sample samples.collections.Collections.Transformations.associateByToWithValueTransform

Sequence<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M { // transform() return destination += transform(element) } // Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to each element of the given sequence. The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_ and * @sample samples.collections.Collections.Transformations.associateTo

Sequence<T>.associateWith(valueSelector: (K) -> V): Map<K, V> { // valueSelector() return destination } // Returns a [Map] where key is the element itself and value is provided by the [valueSelector] function applied to that key. The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_ and * @sample samples.collections.Collections.Transformations.associateWith

Sequence<T>.associateWithTo(destination: M, valueSelector: (K) -> V): M { // valueSelector() return destination.put(element, valueSelector(element)) } // Returns a [Map] where keys are elements from the given sequence and values are produced by the [valueSelector] function applied to each element. The returned map preserves the entry iteration order of the original sequence. The operation is _terminal_ and * @sample samples.collections.Collections.Transformations.associateWithTo

Sequence<T>.toCollection(destination: C): C { // return destination } // Converts a sequence to a collection and returns it. The returned collection is _terminal_.

for (item in this) {
    destination.add(item)
} return destination

/**
 * Returns a new [HashSet] of all elements.
 * The operation is _terminal_ and _anpublic fun <T> Sequence<T>.toHashSet(): HashSet<T>
 * {
 * return toCollection(HashSet<T>())
 * }
 */
public fun <T> Sequence<T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>())
}

/**
 * Returns a [List] containing all elements.
 * The operation is _terminal_ and _anpublic fun <T> Sequence<T>.toList(): List<T>
 * {
 * return this.toList()
 * }
 */
public fun <T> Sequence<T>.toList(): List<T> {
    return this.toList()
}

/**
 * Returns a new [MutableList] filled with all elements of this sequence.
 * The operation is _terminal_ and _anpublic fun <T> Sequence<T>.toMutableList(): MutableList<T>
 * {
 * return this.toMutableList()
 * }
 */
public fun <T> Sequence<T>.toMutableList(): MutableList<T> {
    return this.toMutableList()
}

/**
 * Returns a [Set] containing all elements.
 * The operation is _terminal_ and _anpublic fun <T> Sequence<T>.toSet(): Set<T>
 * {
 * return this.toSet()
 * }
 */
public fun <T> Sequence<T>.toSet(): Set<T> {
    return this.toSet()
}

/**
 * Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.
 * The operation is _intermediate_ and _stateless_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIterable")
public fun <T, R>
    Sequence<T>.flatMap(transform: (T) -> Iterable<R>): Sequence<R> {
    return FlatteningSequence(this,
        transform, Iterable<R>::iterator)
}

/**
 * Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.
 * The operation is _intermediate_ and _stateless_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterable")
@kotlin.internal.InlineOnly
public inline fun <T, R, C : MutableCollection<in R>> Sequence<T>
    .flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(checkIndexOverflow(index++), element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element's index in the original sequence, to the given [destination].
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun <T, R, C : MutableCollection<in R>> Sequence<T>
    .flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(checkIndexOverflow(index++), element)
        destination.addAll(list)
    }
    return destination
}
destination\n\n/**
 * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given [destination].\n
 * The operation is _terminal_.\n
 * @since Kotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.jvm.JvmName("flatMapIterableTo")\n */
public inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each
element of original sequence, to the given [destination].\n
 * The operation is _terminal_.\n
 * @since Kotlin("1.4")\n * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n * @OverloadResolution
ByLambdaReturnType\n * @kotlin.jvm.JvmName("flatMapIterableTo")\n */
public inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups elements of the original sequence by the key returned by the given [keySelector]
function\n * applied to each element and returns a map where each group key is associated with a list of
* corresponding elements.\n * The returned map preserves the entry iteration order of the keys produced from the
original sequence.\n
 * The operation is _terminal_.\n
 * @sample samples.collections.Collections.Transformations.groupBy\n */
public inline fun <T, K, M : MutableMap<in K, MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<T>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups elements of the original sequence by the key returned by the given [keySelector]
 * function and puts to the [destination] map each group key associated with a list of corresponding elements.\n
 * The operation is _terminal_.\n
 * @sample samples.collections.Collections.Transformations.groupBy\n */
public inline fun <T, K, M : MutableMap<in K, MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<T>() }
        list.add(element)
    }
    return destination
}

/**
 * Creates a [Grouping] source from a sequence to be used later with one of group-and-fold
operations\n * using the specified [keySelector] function to extract a key from each element.\n
 * The operation is _intermediate_ and _stateless_.\n
 * @sample samples.collections.Grouping\n */
public inline fun <T, K> Sequence<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {
    return object : Grouping<T, K> {
        override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()
        override fun keyOf(element: T): K = keySelector(element)
    }
}

/**
 * Returns a sequence containing the results of applying the given [transform] function\n * to each element in the original sequence.\n
 * The operation is _intermediate_ and _stateless_.\n
 * @sample samples.collections.Transformations.map\n */
public fun <T, R> Sequence<T>.map(transform: (T) -> R): Sequence<R> {
    return TransformingSequence(this, transform)
}
the results of applying the given [transform] function * to each element and its index in the original sequence.

* @param [transform] function that takes the index of an element and the element itself * and returns the result of the transform applied to the element.

* The operation is _intermediate_ and _stateless_.

* @public fun <T, R> Sequence<T>.mapIndexed(transform: (index: Int, T) -> R): Sequence<R> {

    return TransformingIndexedSequence(this, transform) * Returns a sequence containing only the non-null results of applying the given [transform] function * to each element and its index in the original sequence.

* @param [transform] function that takes the index of an element and the element itself * and returns the result of the transform applied to the element.

* The operation is _intermediate_ and _stateless_.

* Public fun <T, R> Sequence<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): Sequence<R> {

    return TransformingIndexedSequence(this, transform).filterNotNull() * Applies the given [transform] function to each element and its index in the original sequence * and appends only the non-null results to the given

* [destination] * and returns the result of the transform applied to the element.

* The operation is _intermediate_ and _stateless_.

* Public fun <T, R : Any> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {

    return TransformingIndexedSequence(this, transform).filterNotNull() * Returns a sequence containing only the non-null results of applying the given [transform] function * to each element in the original sequence.

* The operation is _intermediate_ and _stateless_.

* Public fun <T, R : Any, C : MutableCollection<R>> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {

    return TransformingIndexedSequence(this, transform).filterNotNull() * Applies the given [transform] function to each element of the original sequence * and appends only the non-null results to the given [destination].

* The operation is _terminal_.

* Public fun <T, R, C : MutableCollection<R>> Sequence<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {

    return TransformingIndexedSequence(this, transform).filterNotNull() * Applies the given [transform] function to each element in the original sequence * and appends only the non-null results to the given [destination].

* The operation is _terminal_.

* Public fun <T, R : Any, C : MutableCollection<R>> Sequence<T>.mapTo(destination: C, transform: (T) -> R): C {

    return TransformingSequence(this, transform).filterNotNull() * Applies the given [transform] function to each element of the original sequence * and appends only the non-null results to the given [destination].

* The operation is _terminal_.

* Public fun <T, R : Any, C : MutableCollection<R>> Sequence<T>.withIndex():

    return IndexingSequence(this) * Returns a sequence containing only distinct elements from the given sequence.

* Among equal elements of the given sequence, only the first one will be present in the resulting sequence.

* The elements in the resulting sequence are in the same order as they were in the source sequence.

* The operation is _intermediate_ and _stateful_.

* Returns a sequence containing only distinct elements from the given sequence * having distinct keys returned by the given [selector] function.

* Among elements of the given sequence with equal keys, only the first one will be present in the resulting sequence.

* The elements in the resulting sequence are in the same order as they were in the source sequence.

* The operation is _intermediate_ and _stateful_.

* Returns a new [MutableSet] containing all distinct elements from the given sequence.
preserves the element iteration order of the original sequence.

The operation is terminal.

`public fun <T> Sequence<T>.toMutableSet(): MutableSet<T> { val set = LinkedHashSet<T>() for (item in this) set.add(item) return set }`

`public fun <T> Sequence<T>.count(): Int { var count = 0
for (item in this) count++
return count }`

`public fun <T> Sequence<T>.maxOrNull(): T? { val iterator = iterator()
if (!iterator.hasNext()) return null
var maxElem = iterator.next()`
maxValue = selector(maxElem)
    do {
        val e = iterator.next()
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    } while (iterator.hasNext())
    return maxElem
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the sequence.
 * If any values produced by [selector] function is 'NaN', the returned result
 * is 'NaN'.
 * @throws NoSuchElementException if the sequence is empty.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.maxOf(selector: (T) -> Double): Double {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the sequence or `null` if there are no elements.
 * If any values produced by [selector] function is 'NaN', the returned result
 * is 'NaN'.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.maxOfOrNull(selector: (T) -> Double): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the sequence or 'NaN' if the sequence is empty.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.maxOf(selector: (T) -> Float): Float {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the sequence or 'NaN' if the sequence is empty.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Sequence<T>.maxOf(selector: (T) -> R): R {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the sequence or 'NaN' if the sequence is empty.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.maxOfOrNull(selector: (T) -> Float): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
maxValue
} * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the sequence.

* @throws NoSuchElementException if the sequence is empty.

* The operation is _terminal_.

* Since Kotlin("1.4")

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the sequence or 'null' if there are no elements.

 * The operation is _terminal_.

 * Since Kotlin("1.4")

 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.experimental.InlineOnly
public inline fun <T, R>

Sequence<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R

if (!iterator.hasNext()) throw NoSuchElementException()

var maxValue = selector(iterator.next())

while (iterator.hasNext())

val v = selector(iterator.next())

if (comparator.compare(maxValue, v) < 0) {

maxValue = v
}


return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the sequence or 'null' if there are no elements.

 * The operation is _terminal_.

 * Since Kotlin("1.4")

 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.experimental.InlineOnly
public inline fun <T, R>

Sequence<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R?

if (!iterator.hasNext()) return null

var maxValue = selector(iterator.next())

while (iterator.hasNext())

val v = selector(iterator.next())

if (comparator.compare(maxValue, v) < 0) {

maxValue = v
}


return maxValue
}

/**
 * Returns the largest element or 'null' if there are no elements.

 * If any of elements is 'NaN' returns 'NaN'.

 * The operation is _terminal_.

 * Since Kotlin("1.4")

 */
@SinceKotlin("1.4")
public fun Sequence<Double>.maxOrNull(): Double? {

val iterator = iterator()

if (!iterator.hasNext()) return null

var max = iterator.next()

while (iterator.hasNext()) {

val e = iterator.next()

max = maxOf(max, e)

}

return max
}

/**
 * Returns the largest element or 'null' if there are no elements.

 * If any of elements is 'NaN' returns 'NaN'.

 * The operation is _terminal_.

 * Since Kotlin("1.4")

 */
@SinceKotlin("1.4")
public fun Sequence<Float>.maxOrNull(): Float? {

val iterator = iterator()

if (!iterator.hasNext()) return null

var max = iterator.next()

while (iterator.hasNext()) {

val e = iterator.next()

max = maxOf(max, e)

}

return max
}

/**
 * Returns the largest element or 'null' if there are no elements.

 * @sample samples.collections.Collections.Aggregates.maxWithOrNull

 */
@SinceKotlin("1.4")
public fun <T : Comparable<T>> Sequence<T>.maxOrNull(): T? {

val iterator = iterator()

if (!iterator.hasNext()) return null

var max = iterator.next()

while (iterator.hasNext()) {

val e = iterator.next()

if (max < e) max = e

}

return max
}

@Deprecated("Use maxWithOrNull instead.",

ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun <T> Sequence<T>.maxWithOrNull(comparator: Comparator<in T>): T? {

val iterator = iterator()

if (!iterator.hasNext()) throw NoSuchElementException()

var max = iterator.next()

while (iterator.hasNext())

val v = iterator.next()

if (comparator.compare(max, v) < 0) max = v

}

return max
}

@Deprecated("Use minOrNull instead.",

ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun Sequence<Double>.min(): Double? {

return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or 'null' if there are no elements.

 * The operation is _terminal_.

 * Since Kotlin("1.4")

 */
@SinceKotlin("1.4")
public inline fun <T, R : Comparable<R>> Sequence<T>.minBy(selector: (T) -> R): T? {

val iterator = iterator()

if (!iterator.hasNext()) throw NoSuchElementException()

var min = iterator.next()

while (iterator.hasNext())

val v = iterator.next()

if (comparator.compare(min, v) > 0) min = v

}

return min
}

@Deprecated("Use minByOrNull instead.",

ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public inline fun <T, R : Comparable<R>> Sequence<T>.minByOrNull(selector: (T) -> R): T? {

val iterator = iterator()

if (!iterator.hasNext()) throw NoSuchElementException()

var min = iterator.next()

while (iterator.hasNext())

val v = iterator.next()

if (comparator.compare(min, v) > 0) min = v

}

return min
}

@Deprecated("Use minWithoutOrNull instead.",

ReplaceWith("minWithoutOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public inline fun <T, R : Comparable<R>> Sequence<T>.minWithoutOrNull(selector: (T) -> R): T? {

val iterator = iterator()

if (!iterator.hasNext()) throw NoSuchElementException()

var min = iterator.next()

while (iterator.hasNext())

val v = iterator.next()

if (comparator.compare(min, v) > 0) min = v

}

return min
}

@Deprecated("Use minWithOrNull instead.",

ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public inline fun <T, R : Comparable<R>> Sequence<T>.minWithOrNull(comparator: Comparator<in R>): T? {

val iterator = iterator()

if (!iterator.hasNext()) throw NoSuchElementException()

var min = iterator.next()

while (iterator.hasNext())

val v = iterator.next()

if (comparator.compare(min, v) > 0) min = v

}

return min
}
Comparable<R> Sequence<T>.minByOrNull(selector: (T) -> R): T? {
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minByOrNull(selector: (T) -> Double): Double? {
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minByOrNull(selector: (T) -> Float): Float? {
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Sequence<T>.minByOrNull(selector: (T) -> R): R? {
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Double): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence, * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Float): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Sequence<T>.minOfOrNull(selector: (T) -> R): R? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Float): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Sequence<T>.minOfOrNull(selector: (T) -> R): R? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Double): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Float): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the sequence. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * The operation is _terminal_.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@kotlin.experimental.InlineOnly
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Float): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minE = iterator.next()
    minValue = selector(minE)
    do {
        var e = iterator.next()
        var v = selector(e)
        if (minValue > v) {
            minE = e
            minValue = v
        }
    } while (iterator.hasNext())
    return minE
}
null

val minValue = selector(iterator.next())
while (iterator.hasNext()) {
    val v = selector(iterator.next())
    if (minValue > v) {
        minValue = v
    }
}

return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the sequence.
 */
@throws
NoSuchElementException if the sequence is empty.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minOfWith(comparator: Comparator<in T>, selector: (T) -> T): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minValue
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the sequence or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minOfWithOrNull(comparator: Comparator<in T>, selector: (T) -> T): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minValue
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun Sequence<Double>.minOrNull(): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        min = minOf(min, e)
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun Sequence<Float>.minOrNull(): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        min = minOf(min, e)
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
public fun <T : Comparable<T>> Sequence<T>.minOrNull(): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (min > e) min = e
    }
    return min
}

@Deprecated("Use minWithOrNull instead.", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun <T> Sequence<T>.minWith(comparator: Comparator<in T>): T? {
    return minWithOrNull(comparator)
}

/**
 * Returns `true` if no elements match the given [predicate].
 */
public inline fun <T> Sequence<T>.none(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns a sequence which performs the given [action] on each element of the original sequence as they pass through it.
 */
@SinceKotlin("1.1")
public fun <T> Sequence<T>.onEach(action: (T) -> Unit): Sequence<T> {
    return map {
        action(it)
        it
    }
}
the \index of an element and the \element itself\n * and performs the action on the element.\n *@\n * The operation is \_intermediate_ and \_stateless_.\n *@SinceKotlin("1.4")\n public fun <T> Sequence<T>.onEachIndexed(action: (index: Int, T) -> Unit): Sequence<T> {
 \n return mapIndexed { index, element -> action(index, element)\n }\n }\n
 *@\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n *@\n * Throws an exception if this sequence is empty. If the sequence can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n *@\n * and calculates the next accumulator value.\n *@\n * The operation is \_terminal_.\n *@\n * @sample collections.Collections.Aggregates.reduce\n *\n * Sequence<T>.reduce(operation: (acc: S, T) -> S): S {
 \n val iterator = this.iterator()\n \n if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.")\n \n var accumulator = S = iterator.next()\n \n while (iterator.hasNext()) {\n \n accumulator = operation(accumulator, iterator.next())\n \n }\n \n return accumulator\n }\n
 *@\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original sequence.\n *@\n * Returns `null` if the sequence is empty. If the sequence can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n *@\n * and calculates the next accumulator value.\n *@\n * The operation is \_terminal_.\n *@\n * @sample samples.collections.Collections.Aggregates.reduceIndexed\n *\n * Sequence<T>.reduceIndexed(operation: (index: Int) -> Unit, acc: S, T) -> S: S {
 \n val iterator = this.iterator()\n \n if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.")\n \n var accumulator = operation(\n \n checkIndexOverflow(index++), accumulator, iterator.next())\n \n var index = 1\n \n while (iterator.hasNext()) {\n \n accumulator = operation(accumulator\n \n checkIndexOverflow(index++), accumulator, iterator.next())\n \n }\n \n return accumulator\n }\n
 *@\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n *@\n * Note that `acc` value passed to \[operation\] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n *@\n * The [initial] value should also be immutable (or should not be mutated)\n * as it may be passed to \[operation\] function later because of sequence's lazy nature.\n *@\n * Returns `null` if the sequence is empty.\n *@\n * The operation is \_intermediate_ and \_stateless_.\n *@\n * @sample samples.collections.Collections.Aggregates.runningFold\n *\n * Sequence<T>.runningFold(initial: R, operation: (acc: R, T) -> R):
Sequence<R> { return sequence {
    yield(initial)
    var accumulator = initial
    for (element in this@runningFold) {
        accumulator = operation(accumulator, element)
        yield(accumulator)
    }
}

/**
 * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original sequence and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting sequence.
 * The [initial] value should also be immutable (or should not be mutated) as it may be passed to [operation] function later because of sequence's lazy nature.
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulate value.
 * The operation is _intermediate_ and _stateless_.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")
public fun <T, R> Sequence<T>.runningFold(initial: R, operation: (T, R) -> R): Sequence<R> {
    return sequence {
        yield(initial)
        for (element in this@runningFold) {
            accumulator = operation(accumulator, element)
            yield(accumulator)
        }
    }
}

/**
 * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this sequence.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting sequence.
 * The operation is _intermediate_ and _stateless_.
 * @sample samples.collections.Collections.Aggregates.runningFoldIndexed
 */
@SinceKotlin("1.4")
public fun <T> Sequence<T>.runningFoldIndexed(initial: T, operation: (T, Int, R) -> R): Sequence<R> {
    return sequence {
        yield(initial)
        for (element in this@runningFoldIndexed) {
            accumulator = operation(checkIndexOverflow(index++), accumulator, element)
            yield(accumulator)
        }
    }
}

/**
 * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original sequence and current accumulator value that starts with the first element of this sequence.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting sequence.
 * The operation is _intermediate_ and _stateless_.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T, S> Sequence<T>.runningReduceIndexed(operation: (Int, T, S) -> S): Sequence<S> {
    return sequence {
        index = 1
        accumulator = initial
        while (iterator.hasNext()) {
            accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
            yield(accumulator)
            index++
        }
    }
}

/**
 * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting sequence.
 * The operation is _intermediate_ and _stateless_.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T, R> Sequence<T>.scan(initial: R, operation: (R, T) -> R): Sequence<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting sequence.
 * The operation is _intermediate_ and _stateless_.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T, R> Sequence<T>.scan(initial: R, operation: (R, T) -> R): Sequence<R> {
    return runningFold(initial, operation)
}
* otherwise it would affect the previous value in resulting sequence.\n* The [initial] value should also be immutable
(should not be mutated)\n* as it may be passed to [operation] function later because of sequence's lazy nature.\n* \n* @param [operation] function that takes the index of an element, current accumulator value\n* and the \n* element itself, and calculates the next accumulator value.\n* The operation is _intermediate_ and _stateless_.\n* \n* @sample samples.Collections.Aggregates.scan

/*@SinceKotlin("1.4")*/'@WasExperimental(ExperimentalStdlibApi::class)\n
    {\n    return runningFoldIndexed(initial, operation)\n    }\n
ReplaceWith("runningReduce(operation())"); level = DeprecationLevel.ERROR\n
/*@SinceKotlin("1.3")*/'@ExperimentalStdlibApi\n
public fun <S, T : S> Sequence<T>.scanReduce(operation: (acc: S, T) -> S): Sequence<S>
    {\n    return runningReduce(operation)\n    }\n
ReplaceWith("runningReduceIndexed(operation())"); level = DeprecationLevel.ERROR\n
/*@SinceKotlin("1.3")*/'@ExperimentalStdlibApi\n
@Deprecated("Use runningReduce instead.",\n    ReplaceWith("runningReduce(operation())"), level = DeprecationLevel.ERROR)\n
public fun <T, R>
    Sequence<T>.scanReduceIndexed(operation: (index: Int, acc: R, T) -> R): Sequence<R> \n    {
    return runningReduceIndexed(operation)\n    }\n
/**
 * Returns the sum of all values produced by [selector] function applied to each element in the sequence.\n * The operation is _terminal_.\n*/
public inline fun <T>
    Sequence<T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the sequence.\n * The operation is _terminal_.\n*/
public inline fun <T>
    Sequence<T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by \n*[selector] function applied to each element in the sequence.\n* The operation is _terminal_.\n*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfDouble")@kotlin.internal.InlineOnly\n
public inline fun <T> Sequence<T>.sumOf(selector: (T) -> Double)\n    : Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by \n*[selector] function applied to each element in the sequence.\n* The operation is _terminal_.\n*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfInt")@kotlin.internal.InlineOnly\n
public inline fun <T> Sequence<T>.sumOf(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by \n*[selector] function applied to each element in the sequence.\n* The operation is _terminal_.\n*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfLong")@kotlin.internal.InlineOnly\n
public inline fun <T> Sequence<T>.sumOf(selector: (T) -> Long): Long {
    var sum: Long = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by \n*[selector] function applied to each element in the sequence.\n* The operation is _terminal_.\n*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@kotlin.experimental.UnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun <T> Sequence<T>.sumOf(selector: (T) -> ULong): ULong {
    var sum: ULong = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by \n*[selector] function applied to each element in the sequence.\n* The operation is _terminal_.\n*/
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@kotlin.experimental.UnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun <T> Sequence<T>.sumOf(selector: (T) -> UInt): UInt {
    var sum: UInt = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
an original collection containing all the non-null elements, throwing an IllegalArgumentException if there are any null elements.

* The operation is _intermediate_ and _stateless_.

```kotlin
public fun <T : Any>
Sequence<T?>.requireNoNulls(): Sequence<T> {
    return map { it ?: throw IllegalArgumentException("null element found in $this.") }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
public fun <T, R>
Sequence<T>.chunked(size: Int, transform: (List<T>) -> R): Sequence<R> {
    return windowed(size, size, partialWindows = true).map { it.transform() }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Sequence<T>): Sequence<T> {
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            var removed = false
            return this@minus.filter { if (!removed && it == element) { removed = true; false } else true }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty()) return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Iterable<T>): Sequence<T> {
    if (isNotEmpty()) return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.convertToSetForSetOperation()
            if (other.isEmpty())
                return this@minus.filterNot { it in other }.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any?): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```

* The operation is _intermediate_ and _stateless_.

```kotlin
operator fun <T> Sequence<T>.minus(elements: Any): Sequence<T> {
    if (elements is Array<out T> || elements is Sequence<out T>)
        return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if (other.isEmpty())
                return this@minus.iterator()
            else
                return this@minus.filterNot { it in other }.iterator()
        }
    }
}
```
override fun iterator(): Iterator<T> {
    if (other.isEmpty())
        return this@minus.iterator()
    else
        return this@minus.filterNot { it in other }.iterator()
}

/**
 * Returns a sequence containing all the elements of the original sequence
 * without the first occurrence of the given [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.minusElement(element: T): Sequence<T> {
    return this@minus.filterNot { it in element }.iterator()
}

/**
 * Splits the original sequence into pairs of lists.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {
    val first = ArrayList<T>()
    val second = ArrayList<T>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Returns a sequence containing all elements of the original sequence and then
 * the given [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.plus(element: T): Sequence<T> {
    return plus(element)
}

/**
 * Returns a sequence containing all elements of the original sequence and then
 * all elements of the given [elements] array.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.plus(elements: Array<out T>): Sequence<T> {
    return plus(elements.asList())
}

/**
 * Returns a sequence containing all elements of the original sequence and then
 * all elements of the given [elements] collection.
 */
public operator fun <T> Sequence<T>.plus(elements: Iterable<T>): Sequence<T> {
    return sequenceOf(this, elements.asSequence()).flatten()
}

/**
 * Returns a sequence containing all elements of the original sequence and then
 * all elements of the given [elements] sequence.
 */
public operator fun <T> Sequence<T>.plus(elements: Sequence<T>): Sequence<T> {
    return sequenceOf(this, elements).flatten()
}

/**
 * Returns a sequence containing all elements of the original sequence and then
 * the given [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.plusElement(element: T): Sequence<T> {
    return plus(element)
}

/**
 * Returns a sequence of snapshots of the window of the given [size]
 * sliding along this sequence with the given [step], where each
 * snapshot is a list.
 */
@SinceKotlin("1.2")
    return windowedSequence(size, step, partialWindows, reuseBuffer = false) { it in this }.asSequence()
}

/**
 * Returns a sequence of results of applying the given [transform] function to
 * each list representing a view over the window of the given [size] sliding
 * along this sequence with the given [step], where each
 * snapshot is a list.
 */
    return windowedSequence(size, step, partialWindows, reuseBuffer = false) { it in this }.asSequence()
}

---

Open Source Used In DNAC Platform Antelope 9373
partialWindows controls whether or not to keep partial windows in the end if any. By default `false` which means partial windows won't be preserved.

```kotlin
@sample samples.collections.Sequences.Transformations.averageWindows
@sample @SinceKotlin("1.2")
unpublic fun <T, R> Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): Sequence<R> { /* Returns a sequence of values built from the elements of 'this' sequence and the [other] sequence with the same index. */ return windowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform) }

/**
 * Samples the given sequence and yield the sample every `size` elements.
 * @sample samples.collections.Sequences.Transformations.zip
 */
@SinceKotlin("1.2")
public fun <T> Sequence<T>.sample(size: Int): Sequence<T> {
    return Sequence { var current = iterator().next(); while (iterator.hasNext()) yield(transform(current, iterator.next()))
    } current = iterator().next();
}

/**
 * Returns a sequence containing the results of applying the given [transform] function to an each pair of two adjacent elements in this sequence.
 * The returned sequence is empty if this sequence contains less than two elements.
 */
public fun <T> Sequence<T>.zipWithNext(transform: (a: T, b: T) -> R): Sequence<R> {
    return sequence { var current = iterator().next(); while (iterator.hasNext()) yield(transform(current, iterator.next()))
    } return current;
}

/**
 * Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same index.
 * The resulting sequence ends as soon as the shortest input sequence ends.
 */
@sample samples.collections.Sequences.Transformations.zip
@public infix fun <T, R> Sequence<T>.zip(other: Sequence<R>): Sequence<Pair<T, R>> {
    return MergingSequence(this, other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same index.
 * The resulting sequence ends as soon as the shortest input sequence ends.
 */
@sample samples.collections.Sequences.Transformations.zip
@public fun <T, R> Sequence<T>.zipWithTransform(transform: (List<T>) -> R): Sequence<R> {
    return windowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform) }

/**
 * Returns a sequence of pairs of each two adjacent elements in this sequence.
 * The returned sequence is empty if this sequence contains less than two elements.
 */
@SinceKotlin("1.2")
public fun <T> Sequence<T>.zipWithNext(): Sequence<Pair<T, T>> {
    return zipWithNext { a, b -> a to b }
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * The operation is _terminal_.
 */
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {
    buffer.append(separator) if (limit < 0 || count <= limit) { buffer.appendElement(element, transform) } else break
    } return buffer.toString() + truncated + postfix
}
```

Open Source Used In DNAC Platform Antelope 9374
public fun Sequence<Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfShort")
public fun Sequence<Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfInt")
public fun Sequence<Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Sequence<Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Sequence<Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("averageOfDouble")
public fun Sequence<Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Sequence<Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Sequence<Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Sequence<Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Sequence<Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Sequence<Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Sequence<Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
== element) { removed = true; false } else true }
}

/**
 * Returns a set containing all elements of the original set except the elements contained in the given 
 * elements array.
 * The returned set preserves the element iteration order of the original set.
 * The elements array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> minus(elements: Array<T>): Set<T> {
    val result = LinkedHashSet<T>((this)
        .removeAll(elements)
        .toSet())
    if (other.isEmpty())
        return this.toSet()
    if (other is
    return other.filterNotTo(LinkedHashSet<T>(this)) { it in other }
    return result.addAll(other)
    return result
}

/**
 * Returns a set containing all elements of the original set except the elements contained in the given 
 * elements collection.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] collection may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] collection may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> minus(elements: Iterable<T>): Set<T> {
    val result = LinkedHashSet<T>(
        (this)
            .removeAll(elements)
            .toSet())
    if (other.isEmpty())
        return this.toSet()
    if (other is
    return result.addAll(other)
    return result
}

/**
 * Returns a set containing all elements of the original set except the elements contained in the given 
 * elements sequence.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] sequence may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set except the elements contained in the given [elements] sequence.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> minus(elements: Sequence<T>): Set<T> {
    val result = LinkedHashSet<T>(
        (this)
            .removeAll(elements)
            .toSet())
    if (other.isEmpty())
        return this.toSet()
    if (other is
    return result.addAll(other)
    return result
}

/**
 * Returns a set containing all elements of the original set except the given 
 * element.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> minusElement(element: T): Set<T> {
    return minus(element)
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElement(element: T): Set<T> {
    return plus(element)
}

/**
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElements(Sequence<T>): Set<T> {
    val result = LinkedHashSet<T>(
        (this)
            .addAll(elements)
            .toSet())
    if (other.isEmpty())
        return this.toSet()
    if (other is
    return result.addAll(other)
    return result
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElement(element: T): Set<T> {
    return plus(element)
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElements(Sequence<T>): Set<T> {
    val result = LinkedHashSet<T>(
        (this)
            .addAll(elements)
            .toSet())
    if (other.isEmpty())
        return this.toSet()
    if (other is
    return result.addAll(other)
    return result
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElement(element: T): Set<T> {
    return plus(element)
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElements(Sequence<T>): Set<T> {
    val result = LinkedHashSet<T>(
        (this)
            .addAll(elements)
            .toSet())
    if (other.isEmpty())
        return this.toSet()
    if (other is
    return result.addAll(other)
    return result
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElement(element: T): Set<T> {
    return plus(element)
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElements(Sequence<T>): Set<T> {
    val result = LinkedHashSet<T>(
        (this)
            .addAll(elements)
            .toSet())
    if (other.isEmpty())
        return this.toSet()
    if (other is
    return result.addAll(other)
    return result
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
 * thus the elements are required to have a correct and stable implementation of `hashCode()`
 * that doesn't change between successive invocations.
 * Returns a set containing all elements of the original set and the given 
 * elements array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 */
public <T> Set<T> plusElement(element: T): Set<T> {
    return plus(element)
}

/**
 * Returns a set containing all elements of the original set and then the given 
 * element if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * The [elements] array may be converted to a [HashSet] to speed up the operation,
CharSequence.elementAt(index: Int): Char

```kotlin
/**
 * Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.
 */
public inline fun CharSequence.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
/**
 * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.
 */
public fun CharSequence.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

```kotlin
/**
 * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.
 */
public fun CharSequence.getOrNull(index: Int): Char? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

```kotlin
/**
 * Returns index of the first character matching the given [predicate], or -1 if the char sequence does not contain such character.
 */
public inline fun CharSequence.indexOfFirst(predicate: (Char) -> Boolean): Int {
    for (index in indices) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}
```

```kotlin
/**
 * Returns index of the last character matching the given [predicate], or -1 if the char sequence does not contain such character.
 */
public inline fun CharSequence.indexOfLast(predicate: (Char) -> Boolean): Int {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return index
    }
    return -1
}
```

```kotlin
/**
 * Returns the last character.
 * @throws [NoSuchElementException] if the char sequence is empty.
 */
public fun CharSequence.last(): Char {
    if (isEmpty())
        throw NoSuchElementException("Char sequence is empty.")
    return this[lastIndex]
}
```

```kotlin
/**
 * Returns the last character matching the given [predicate].
 * @throws [NoSuchElementException] if no such character is found.
 */
public inline fun CharSequence.last(predicate: (Char) -> Boolean): Char {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Char sequence contains no character matching the predicate.")
}
```

```kotlin
/**
 * Returns the last character, or `null` if the [index] is out of bounds of this char sequence.
 */
public fun CharSequence.lastOrNull(): Char? {
    return if (isEmpty()) null else this[length - 1]
}
```

```kotlin
/**
 * Returns the last character matching the given [predicate], or `null` if no such character was found.
 * @throws [NoSuchElementException] if no such character is found.
 */
public inline fun CharSequence.lastOrNull(predicate: (Char) -> Boolean): Char? {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    return null
}
```

```kotlin
/**
 * Returns first character.
 * @throws [NoSuchElementException] if the char sequence is empty.
 */
public fun CharSequence.first(): Char {
    if (isEmpty())
        throw NoSuchElementException("Char sequence is empty.")
    return this[0]
}
```

```kotlin
/**
 * Returns the first character matching the given [predicate].
 * @throws [NoSuchElementException] if no such character is found.
 */
public inline fun CharSequence.first(predicate: (Char) -> Boolean): Char {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Char sequence contains no character matching the predicate.")
}
```

```kotlin
/**
 * Returns the first character, or `null` if the char sequence is empty.
 */
public fun CharSequence.firstOrNull(): Char? {
    return if (isEmpty()) null else this[0]
}
```

```kotlin
/**
 * Returns the first character matching the given [predicate], or `null` if no such character was found.
 */
public inline fun CharSequence.firstOrNull(predicate: (Char) -> Boolean): Char? {
    for (element in this) if (predicate(element)) return element
    return null
}
```

```kotlin
/**
 * Returns a random character from this char sequence.
 */
public fun * @throws [NoSuchElementException] if the char sequence is empty.
 */
public fun Random.nextChar(): Char {
    return this[Random.nextInt(length)]
}
```
substring(0, index + 1)
    return ""
}

/**
 * Returns a subsequence of this char sequence containing all characters except first characters that satisfy the given [predicate].
 * 
 * @sample samples.text.Strings.drop
 */
public inline fun CharSequence.dropWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in this.indices)
        if (!predicate(this[index]))
            return subSequence(index, length)
    return ""
}

/**
 * Returns a string containing all characters except first characters that satisfy the given [predicate].
 * 
 * @sample samples.text.Strings.drop
 */
public inline fun String.dropWhile(predicate: (Char) -> Boolean): String {
    for (index in this.indices)
        if (!predicate(this[index]))
            return substring(index)
    return ""
}

/**
 * Returns a char sequence containing only those characters from the original char sequence that match the given [predicate].
 * 
 * @param [predicate] function that takes the index of a character and the character itself
 * and returns the result of predicate evaluation on the character.
 * 
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun CharSequence.filter(predicate: (Char) -> Boolean): CharSequence {
    return filterTo(StringBuilder(), predicate)
}

/**
 * Returns a string containing only those characters from the original string that match the given [predicate].
 * 
 * @param [predicate] function that takes the index of a character and the character itself
 * and returns the result of predicate evaluation on the character.
 * 
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun String.filter(predicate: (Char) -> Boolean): String {
    return filterTo(StringBuilder(), predicate).toString()
}

/**
 * Returns a char sequence containing only those characters from the original char sequence that do not match the given [predicate].
 * 
 * @param [predicate] function that takes the index of a character and the character itself
 * and returns the result of predicate evaluation on the character.
 * 
 * @sample samples.collections.Collections.Filtering.filterNot
 */
public inline fun CharSequence.filterNot(predicate: (Char) -> Boolean): CharSequence {
    return filterNotTo(StringBuilder(), predicate)
}

/**
 * Returns a string containing only those characters from the original string that do not match the given [predicate].
 * 
 * @param [predicate] function that takes the index of a character and the character itself
 * and returns the result of predicate evaluation on the character.
 * 
 * @sample samples.collections.Collections.Filtering.filterNot
 */
public inline fun String.filterNot(predicate: (Char) -> Boolean): String {
    return filterNotTo(StringBuilder(), predicate).toString()
}

/**
 * Appends all characters matching the given [predicate] to the given [destination].
 * 
 * @sample samples.collections.Collections.Filtering.filterIndexed
 */
public inline fun <C : Appendable> CharSequence.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {
    for (index in 0 until length) {
        val element = get(index)
        if (predicate(index, element))
            destination.append(element)
    }
    return destination
}

/**
 * Appends all characters not matching the given [predicate] to the given [destination].
 * 
 * @sample samples.collections.Collections.Filtering.filterTo
 */
public inline fun <C : Appendable> CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (index in 0 until length) {
        val element = get(index)
        if (!predicate(element))
            destination.append(element)
    }
    return destination
}

/**
 * Returns a char sequence containing only those characters from the original char sequence that do not match the specified range of [indices].
 * 
 * @param [indices] range of indices to slice
 * @sample samples.text.Strings.slice
 */
public fun CharSequence.slice(indices: IntRange): CharSequence {
    if (indices.isEmpty()) return ""
    return subSequence(indices)
}

/**
 * Returns a string containing only those characters from the original string that do not match the specified range of [indices].
 * 
 * @param [indices] range of indices to slice
 * @sample samples.text.Strings.slice
 */
public fun String.slice(indices: IntRange): String {
    if (indices.isEmpty()) return ""
    return substring(indices)
}

/**
 * Returns a char sequence containing characters of the original char sequence at the specified range of [indices].
 * 
 * @param [indices] range of indices to slice
 * @sample samples.text.Strings.slice
 */
public fun(CharSequence: CharSequence, indices: Iterable<Int>): CharSequence {
    val size = indices.collectionSizeOrDefault(10)
    var destination = StringBuilder()
    if (indices.isEmpty()) return destination
    return destination
}

/**
 * Returns a char sequence containing characters of the original char sequence at the specified range of [indices].
 * 
 * @param [indices] range of indices to slice
 * @sample samples.text.Strings.slice
 */
public fun(String: String, indices: Iterable<Int>): String {
    val size = indices.collectionSizeOrDefault(10)
    var destination = StringBuilder()
    if (indices.isEmpty()) return destination
    return destination
}
val result = StringBuilder(size)
  for (i in indices) {
    result.append(get(i))
  }
  return result
}

/**
 * Returns a string containing characters from the original string at specified [indices].
 */
@kotlin.internal.InlineOnly
public inline fun String.slice(indices: Iterable<Int>): String {
  return subSequence((this as CharSequence).slice(indices).toSet().size)
}

/**
 * Returns a string containing the first [n] characters from this string, or the entire string if this string is shorter.
 */
@kotlin.internal.InlineOnly
public inline fun String.take(n: Int): String {
  if (n >= 0) {
    return substring(0, n.coerceAtMost(length))
  }
  return this
}

/**
 * Returns a string containing the last [n] characters from this string, or the entire string if this string is shorter.
 */
@kotlin.internal.InlineOnly
public inline fun String.takeLast(n: Int): String {
  if (n >= 0) {
    return substring(length - n.coerceAtMost(length))
  }
  return this
}

/**
 * Returns a subsequence of this char sequence containing last characters that satisfy the given [predicate].
 */
public inline fun CharSequence.takeLastWhile(predicate: (Char) -> Boolean): CharSequence {
  for (index in lastIndex downTo 0) {
    if (!predicate(this[index])) {
      return subSequence(index + 1, length)
    }
  }
  return subSequence(0, length)
}

/**
 * Returns a string containing last characters that satisfy the given [predicate].
 */
public inline fun String.takeLastWhile(predicate: (Char) -> Boolean): String {
  for (index in lastIndex downTo 0) {
    if (!predicate(this[index])) {
      return substring(index + 1)
    }
  }
  return this
}

/**
 * Returns a subsequence of this char sequence containing the first characters that satisfy the given [predicate].
 */
public inline fun CharSequence.takeWhile(predicate: (Char) -> Boolean): CharSequence {
  for (index in 0 until length) {
    if (!predicate(get(index))) {
      return subSequence(0, index)
    }
  }
  return subSequence(0, length)
}

/**
 * Returns a string containing the first characters that satisfy the given [predicate].
 */
public inline fun String.takeWhile(predicate: (Char) -> Boolean): String {
  for (index in 0 until length) {
    if (!predicate(get(index))) {
      return substring(0, index)
    }
  }
  return this
}

/**
 * Returns a char sequence in reversed order.
 */
public fun CharSequence.reversed(): CharSequence {
  return this.reversed
}

/**
 * Returns a string in reversed order.
 */
@kotlin.internal.InlineOnly
public inline fun String.reversed(): String {
  return (this as CharSequence).reversed().toString()
}

/**
 * Returns a [Map] containing key-value pairs provided by [transform] function applied to characters of the given char sequence.
 */
public inline fun <K, V> CharSequence.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {
  val capacity = mapCapacity(length).coerceAtMost(16)
  return associateTo(LinkedHashMap<K, V>(capacity),
    transform)
}

/**
 * Returns a string with characters in reversed order.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.reverse(): String {
  return this.reversed()

mapCapacity(length).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Char>(capacity),
    keySelector) * Returns a [Map] containing the values provided by [valueTransform] and indexed by
    [keySelector] functions applied to characters of the given char sequence. * If any two characters would have
    the same key returned by [keySelector] the last one gets added to the map. * The returned map preserves the
    entry iteration order of the original char sequence. * * @sample
    samples.text.Strings.associateByWithValueTransform

    CharSequence.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> { * val capacity = mapCapacity(length).coerceAtLeast(16) * return associateByTo(LinkedHashMap<K, V>(capacity),
    keySelector, valueTransform) * Populates and returns the [destination] mutable map with key-value
    pairs. * * If any two characters would have the same key returned by [keySelector] the last one gets added to the
    map. * * @sample
    samples.text.Strings.associateByTo

    @SinceKotlin("1.3")
    CharSequence.associateWith(valueSelector: (Char) -> V): Map<Char, V> { * val result = LinkedHashMap<Char, V>(mapCapacity(length.coerceAtMost(128)).coerceAtLeast(16)) * return associateWithTo(result,
    valueSelector) * Populates and returns the [destination] mutable map with key-value pairs. * * Where key is provided by the
    [valueSelector] function and value is provided by the [valueTransform] function applied to characters of the
    given char sequence. * * If any two characters would have the same key returned by [keySelector] the last one gets added to the
    map. * * @sample
    samples.text.Strings.associateWithTo

    @SinceKotlin("1.3")
    CharSequence.toCollection(destination: C): C { * for (item in this) * return destination.add(item) * } * Returns a new [HashSet] of all characters. * * @sample
    samples.text.Strings.toCollection

    @SinceKotlin("1.3")
    CharSequence.toMutableList(): MutableList<Char> { * return toCollection(ArrayList<Char>(length)) * } * Returns a [MutableList] filled with all characters of this char sequence. * * @sample
    samples.text.Strings.toMutableList

    @SinceKotlin("1.3")
    CharSequence.toSet(): Set<Char> { * return toCollection(HashSet<Char>(result(128))) * } * Returns a [Set] of all characters. * * @sample
    samples.text.Strings.toSet
sequence.
 */
public fun CharSequence.toSet(): Set<Char> { return when (length) {
    0 -> emptySet()
    1 -> setOf(this[0])
    else -> toCollection(LinkedHashSet<Char>(mapCapacity(length.coerceAtMost(128))))
}
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each character of original char sequence.
 * @param characters
 * @param index
 * @param destination
 * @sample samples.collections.Transformations.flatMap
 */
@kotlin.jvm.JvmName("flatMapIndexed")
public inline fun <R> CharSequence.flatMapIndexed(transform: (Char, Int) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each character
 * and its index in the original char sequence, to the given [destination].
 * @param characters
 * @param index
 * @param destination
 * @sample samples.collections.Transformations.flatMapIndexedTo
 */
@kotlin.jvm.JvmName("flatMapIndexed Iterable")
public inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapIndexedTo(destination: C, transform: (Char, Int) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(element, index++)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups characters of the original char sequence by the key returned by the given [keySelector] function
 * applied to each character and puts to the [destination] map each group key associated with a list of corresponding characters.
 * @param characters
 * @param index
 * @param destination
 * @sample samples.collections.Transformations.groupBy
 */
@kotlin.jvm.JvmName("groupByTo")
public inline fun <K, M : MutableMap<in K, List<Char>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Char>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each character of the original char sequence
 * by the key returned by the given [keySelector] function
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 * @param characters
 * @param index
 * @param destination
 * @sample samples.collections.Transformations.groupByKeysAndValues
 */
@kotlin.jvm.JvmName("groupByKeysAndValuesTo")
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByKeysAndValuesTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Char>() }
        list.add(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each character
 * @param characters
 * @param index
 * @param destination
 * @sample samples.collections.Transformations.flatMap
 */
@kotlin.jvm.JvmName("flatMapTo")
public inline fun <R> CharSequence.flatMapTo(destination: R, transform: (Char) -> Iterable<R>): R {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups characters of the original char sequence by the key returned by the given [keySelector] function
 * applied to each character and puts to the [destination] map each group key associated with a list of corresponding characters.
 * @param characters
 * @param index
 * @param destination
 * @sample samples.collections.Transformations.groupBy
 */
@kotlin.jvm.JvmName("groupBy")
public inline fun <K, M : MutableMap<in K, List<Char>>> CharSequence.groupBy(destination: M, keySelector: (Char) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Char>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each character of the original char sequence
 * by the key returned by the given [keySelector] function
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 * @param characters
 * @param index
 * @param destination
 * @sample samples.collections.Transformations.groupByKeysAndValues
 */
@kotlin.jvm.JvmName("groupByKeysAndValues")
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByKeysAndValues(destination: M, keySelector: (Char) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Char>() }
        list.add(element)
        destination.addAll(list)
    }
    return destination
}
valueTransform: (Char) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Creates a [Grouping] source from a char sequence to be used later with one of group-and-
fold operations * using the specified [keySelector] function to extract a key from each character.
 * @param @sample samples.collections.Grouping.groupingByEachCount */
CharSequence.groupingBy(crossinline keySelector: (Char) -> K): Grouping<Char, K> {
    return object {
        override fun sourceIterator(): Iterator<Char> = this@groupingBy.iterator()
        override fun keyOf(element: Char): K = keySelector(element)
    }
}

/**
 * Returns a list containing the results of applying the given [transform] function * to each character in the original char sequence.
 * @param @sample samples.text.Strings.map*/
CharSequence.map(transform: (Char) -> R): List<R> {
    return mapToArray(List<R>(length), transform) *
}

/**
 * Returns a list containing only the non-null results of applying the given [transform] function * to each character and its index in the original char sequence.
 * @param @param [transform] function that takes the index of a character and the character itself
 * and returns the result of the transform applied to the character.
 * @param @sample samples.collections.Collections.Transformations.mapNotNull*/
CharSequence.mapNotNull(transform: (Char) -> R?): List<R> {
    return mapNotNullToList(ArrayList<R>(), transform) *
}

/**
 * Applies the given [transform] function to each character and the character itself
 * and returns the result of the transform applied to the character.
 * @param @sample samples.collections.Collections.Transformations.mapIndexed*/
CharSequence.mapIndexed(transform: (index: Int, Char) -> R): List<R> {
    return mapIndexedToArray(List<R>(length), transform) *
}

/**
 * Applies the given [transform] function to each character and the character itself
 * and returns the result of the transform applied to the character.
 * @param @sample samples.collections.Collections.Transformations.mapIndexedNotNull*/
CharSequence.mapIndexedNotNull(transform: (index: Int, Char) -> R?): List<R> {
    return mapIndexedNotNullToArray(ArrayList<R>(), transform) *
}

/**
 * Applies the given [transform] function to each character of the original char sequence
 * and appends the results to the given [destination].
 * @param @sample samples.collections.Collections.Transformations.mapIndexedNotNullTo*/
CharSequence.mapIndexedNotNullTo(destination: Collection<R>, transform: (index: Int, Char) -> R?): Collection<R> {
    for (item in this) {
        transform(index, item)?.let { destination.add(it) }
    }
    return destination
}

/**
 * Applies the given [transform] function to each character of the original char sequence
 * and appends the results to the given [destination].
 * @param @sample samples.collections.Collections.Transformations.mapIndexedTo*/
CharSequence.mapIndexedTo(destination: Collection<R>, transform: (index: Int, Char) -> R): Collection<R> {
    for (item in this) {
        destination.add(transform(index, item))
    }
    return destination
}

/**
 * Applies the given [transform] function to each character in the original char sequence
 * and appends only the non-null results to the given [destination].
 * @param @sample samples.collections.Collections.Transformations.mapNotNullTo*/
CharSequence.mapNotNullTo(destination: Collection<R>, transform: (Char) -> R?): Collection<R> {
    for (element in this) {
        transform(element)?.let { destination.add(it) }
    }
    return destination
}

/**
 * Applies the given [transform] function to each character of the original char sequence
 * and appends the results to the given [destination].
 * @param @sample samples.collections.Collections.Transformations.mapTo*/
CharSequence.mapTo(destination: Collection<R>, transform: (Char) -> R): Collection<R> {
    for (item in this) {
        destination.add(transform(item))
    }
    return destination
}
samples.collections.Collections.Aggregates.any

```kotlin
/*
 * @param predicate - function that takes an element of this and returns `true` if it matches the given condition.
 */
public fun CharSequence.any(predicate: (Char) -> Boolean): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if at least one character matches the given [predicate].
 */
public fun CharSequence.any(predicate: Char.() -> Boolean): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right. *
 * Returns the specified [initial] value if the char sequence is empty. *
 */
public inline fun <R> CharSequence.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left. *
 * Returns the specified [initial] value if the char sequence is empty. *
 */
public inline fun <R> CharSequence.foldRight(initial: R, operation: (Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index), accumulator)
        --index
    }
    return accumulator
}
```

```kotlin
/**
 * Performs the given [action] on each character. *
 */
public inline fun CharSequence.forEach(action: (Char) -> Unit): Unit {
    for (element in this) action(element)
}
```

```kotlin
/**
 * Performs the given [action] on each character, providing sequential index with the character. *
 * @param action - function that takes the index of a character and the character itself and performs the action on the character. *
 */
public inline fun CharSequence.forEachIndexed(action: (index: Int, Char) -> Unit): Unit {
    var index = 0
    for (item in this) action(index++, item)
}
```

```kotlin
@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}
```

```kotlin
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public inline fun <R : Comparable<R>> CharSequence.maxBy(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var maxElem = first()
    var lastIndex = 0
    for (i in 1..lastIndex) {
        if (selector(get(i)) > selector(get(lastIndex))) {
            maxElem = get(i)
            lastIndex = i
        }
    }
    return maxElem
}
```

```kotlin
Open Source Used In DNAC Platform Antelope  9384
```

```kotlin
samples.collections.Collections.Aggregates.any
```
applied to each character in the char sequence.

* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
* @throws NoSuchElementException if the char sequence is empty.

*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public @JvmName("maxOf") @kotlin.experimental.ExperimentalTypeInference inline fun CharSequence.

maxOf(selector: (Char) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each character in the char sequence.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the char sequence is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public @JvmName("maxOf") @kotlin.experimental.ExperimentalTypeInference inline fun CharSequence.

maxOf(selector: (Char) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each character in the char sequence.
 * @throws NoSuchElementException if the char sequence is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public @JvmName("maxOf") @kotlin.experimental.ExperimentalTypeInference inline fun CharSequence.

maxOf(selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function. Returns the largest value among all values produced by [selector] function applied to each character in the char sequence.
 * @throws NoSuchElementException if the char sequence is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public @JvmName("maxOf") @kotlin.experimental.ExperimentalTypeInference inline fun CharSequence.

maxOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}
function applied to each character in the char sequence or `null` if there are no characters.

```kotlin
CharSequence.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {
    return null
}
```

Returns the largest character or `null` if there are no characters.

```kotlin
CharSequence.maxOrNull(): Char? {
    return null
}
```

Returns the largest character.

```kotlin
CharSequence.maxWith(comparator: Comparator<in Char>): Char? {
    return null
}
```

Returns the first character having the largest value according to the provided comparator or `null` if there are no characters.

```kotlin
CharSequence.min(): Char? {
    return null
}
```

Returns the smallest character.

```kotlin
CharSequence.minWithOrNull(comparator: Comparator<in Char>): Char? {
    return null
}
```

Returns the first character yielding the smallest value of the given function or `null` if there are no characters.

```kotlin
CharSequence.minOf(selector: (Char) -> Double): Double {
    return null
}
```

Returns the smallest value among all values produced by function when applied to each character in the char sequence.

```kotlin
CharSequence.minOf(selector: (Char) -> Float): Float {
    return null
}
```

Returns the smallest value among all values produced by function when applied to each character in the char sequence.

```kotlin
CharSequence.minOf(selector: (Char) -> R): R {
    return null
}
```

Returns the smallest value among all values produced by function when applied to each character in the char sequence.
characters.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

Returns the smallest value among all values produced by [selector] function applied to each character in the char sequence or `null` if there are no characters.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each character in the char sequence.

* * *

Returns the first character having the smallest value according to the provided [comparator] or `null` if there are no characters.

* * *

Returns `true` if no characters match the given [predicate].

* * *

Performs the given [action] on each character and returns the char sequence itself.
afterwards.\n*\n@SinceKotlin("1.1")\npublic inline fun <S : CharSequence> S.onEach(action: (Char) -> Unit): S
{\n    return apply { for (element in this) action(element) }\n}***\n* Performs the given [action] on each character, providing sequential index with the character,\n* and returns the char sequence afterwards.\n*\n@param [action] function that takes the index of a character and the character itself\n* and performs the action on the character.\n*\n@SinceKotlin("1.4")\npublic inline fun <S : CharSequence> S.onEachIndexed(action: (index: Int, Char) -> Unit): S
{\n    return apply { forEachIndexed(action) }\n}***\n* Accumulates value starting with the first character and applying [operation] from left to right\n* to current accumulator value and each character.\n*\n* Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n* please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n* \n* @param [operation] function that takes the index of a character, current accumulator value and the character itself,\n* and calculates the next accumulator value.\n* \n* @sample samples.collections.Collections.Aggregates.reduce\nCharSequence.reduce(operation: (acc: Char, Char) -> Char): Char
{\n    if (isEmpty()) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}***\n* Accumulates value starting with the first character and applying [operation] from left to right\n* to current accumulator value and each character with its index in the original char sequence.\n*\n* Returns `null` if the char sequence is empty.\n* \n* @param [operation] function that takes the index of a character, current accumulator value and the character itself,\n* and calculates the next accumulator value.\n* \n* @sample samples.collections.Collections.Aggregates.reduceIndexed\nCharSequence.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char
{\n    if (isEmpty()) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}***\n* Accumulates value starting with the first character and applying [operation] from left to right\n* to current accumulator value and each character with its index in the original char sequence.\n*\n* Returns `null` if the char sequence is empty.\n* \n* @param [operation] function that takes the index of a character, current accumulator value and the character itself,\n* and calculates the next accumulator value.\n* \n* @sample samples.collections.Collections.Aggregates.reduceIndexedOrNull\nCharSequence.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char
{\n    if (isEmpty()) return null\n    var accumulator = this[0]\n    for (index in 1..lastIndex) {\n        accumulator = operation(index, accumulator, this[index])\n    }\n    return accumulator\n}***\n* Accumulates value starting with the first character and applying [operation] from left to right\n* to current accumulator value and each character.\n*\n* Returns `null` if the char sequence is empty.\n* \n* @param [operation] function that takes the index of a character, current accumulator value and the character itself,\n* and calculates the next accumulator value.\n* \n* @sample samples.collections.Collections.Aggregates.reduceRight\nCharSequence.reduceRight(operation: (Char, acc: Char) -> Char): Char
{\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.")\n    var accumulator = get(index - 1)\n    while (index != 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return accumulator\n}***\n* Accumulates value starting with the last character and applying [operation] from right to
left\n * to each character with its index in the original char sequence and current accumulator value.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use \n * \n * function that takes the index of a character, the character itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @param operation \n * \n * @sample samples.collections.Collections.Aggregates.reduceRightIndexedOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@publication inline fun CharSequence.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Returns a list containing successive accumulation values generated by applying \n * \n * @param operation \n * \n * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")
public inline fun CharSequence.runningReduce(initial: Char, operation: (Char, Char) -> Char): List<Char> {
    val result = ArrayList<Char>(length + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}
CharSequence.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> { 
    if (isEmpty()) return emptyList() 
    var accumulator = this[0] 
    val result = ArrayList<Char>(this.length).apply { add(accumulator) } 
    for (index in 1 until length) { 
        accumulator = operation(accumulator, this[index]) 
        result.add(accumulator) 
    } 
    return result 
} 

/** 
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each character, its index in the original char sequence and current accumulator value that starts with the first character of this char sequence. 
 * Note that 'acc' value passed to [operation] function should not be mutated: it otherwise would affect the previous value in resulting list. 
 * @param [operation] function that takes the index of a character, current accumulator value and a character. 
 * @sample samples.collections.Collections.Aggregates.runningReduceIndexed */
@SinceKotlin("1.4")
@ExperimentalStdlibApi
public inline fun <R> 
CharSequence.scan(initial: R, operation: (acc: R, Char) -> R): List<R> { 
    return runningFold(initial, operation) 
} 

/** 
 * Returns the sum of all values produced by [selector] function applied to each character in the char sequence. 
 * @sample samples.collections.Collections.Aggregates.sumOf */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution 
ByLambdaReturnType 
@kotlin.jvm.JvmName("sumOfDouble") 
@kotlin.internal.InlineOnly 
public inline fun 
CharSequence.sumOf(selector: (Char) -> Double): Double { 
    var sum: Double = 0.toDouble() 
    for (element in this) { 
        sum += selector(element) 
    } 
    return sum 
}
function applied to each character in the char sequence.

// Since Kotlin("1.4") @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOf")
@kotlin.internal.InlineOnly
public inline fun
CharSequence.sumOf(selector: (Char) -> Int): Int { var sum: Int = 0.toInt() for (element in this) { sum += selector(element) } return sum } * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.

// Since Kotlin("1.4") @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfLong")
@kotlin.internal.InlineOnly
public inline fun
CharSequence.sumOf(selector: (Char) -> Long): Long { var sum: Long = 0.toLong() for (element in this) { sum += selector(element) } return sum } * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.

// Since Kotlin("1.4") @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfUInt")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
CharSequence.sumOf(selector: (Char) -> UInt): UInt { var sum: UInt = 0.toUInt() for (element in this) { sum += selector(element) } return sum } * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.

// Since Kotlin("1.4") @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
CharSequence.sumOf(selector: (Char) -> ULong): ULong { var sum: ULong = 0.toULong() for (element in this) { sum += selector(element) } return sum } * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.

// Since Kotlin("1.2")
public fun CharSequence.chunked(size: Int): List<String> { return windowed(size, size, partialWindows = true) } * Splits this char sequence into a list of strings each not exceeding the given [size]. The last string in the resulting list must have less characters than the given [size]. @param size the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence. * * @sample samples.text.Strings.chunked

// Since Kotlin("1.2")
public fun <R> CharSequence.chunked(size: Int, transform: (CharSequence) -> R): List<R> { return windowed(size, size, partialWindows = true, transform = transform) } * Splits this char sequence into several char sequences each not exceeding the given [size] and applies the given [transform] function to each. * * @return list of results of the [transform] applied to each char sequence. * * @sample samples.text.Strings.chunkedTransform

// Since Kotlin("1.2")
public fun CharSequence.chunkedSequence(size: Int): Sequence<String> { return windowed(size, size, partialWindows = true) } * Splits this char sequence into several char sequences each not exceeding the given [size] and applies the given [transform] function to each. * * @return list of results of the [transform] applied to each char sequence. * * @sample samples.collections.Collections.Transformations.chunked

// Since Kotlin("1.2")
public fun CharSequence.chunkedToSequence(size: Int, transform: (CharSequence) -> R): Sequence<String> { return windowed(size, size, partialWindows = true, transform = transform) } * Splits this char sequence into several char sequences each not exceeding the given [size] and applies the given [transform] function to each. * * @return list of results of the [transform] applied to each char sequence. * * @sample samples.text.Strings.chunkedTransformToSequence

// Since Kotlin("1.2")
public fun CharSequence.splitToSequence(size: Int): Sequence<String> { return chunkedSequence(size) { it.toString() } } * Splits this char sequence into several char sequences each not exceeding the given [size] and applies the given [transform] function to each. * * @return list of results of the [transform] applied to each char sequence. * * @sample samples.collections.Collections.Transformations.chunked

Open Source Used In DNAC Platform Antelope 9391
sequence into pair of char sequences,
where `first` char sequence contains characters for which `predicate` yielded `true`, `second` char sequence contains characters for which `predicate` yielded `false`.

@sample samples.text.Strings.partition

*/
public inline fun CharSequence.partition(predicate: (Char) -> Boolean): Pair<CharSequence, CharSequence> {
    val first = StringBuilder()
    val second = StringBuilder()
    for (element in this) {
        if (predicate(element)) {
            first.append(element)
        } else {
            second.append(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original string into pair of strings,
 * where `first` string contains characters for which `predicate` yielded `true`, `second` string contains characters for which `predicate` yielded `false`.
 * @sample samples.text.Strings.partition
 */
public inline fun String.partition(predicate: (Char) -> Boolean): Pair<String, String> {
    val first = StringBuilder()
    val second = StringBuilder()
    for (element in this) {
        if (predicate(element)) {
            first.append(element)
        } else {
            second.append(element)
        }
    }
    return Pair(first.toString(), second.toString())
}

/**
 * Returns a list of snapshots of the window of the given size sliding along this char sequence with the given step, where each snapshot is a string.
 * Several last strings may have less characters than the given size. Both size and step must be positive and can be greater than the number of elements in this char sequence.
 * You should not store it or allow it to escape in some way, unless you made a snapshot of it.
 * Several last char sequences may have less characters than the given size.
 * Both size and step must be positive and can be greater than the number of elements in this char sequence.
 * @param size the number of elements to take in each window
 * @param step the number of elements to move the window forward by on each step, by default 1
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 */
@SinceKotlin("1.2")
public fun CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<String> {
    return windowed(size, step, partialWindows) { it.toString() }
}

/**
 * Returns a list of results of applying the given transform function to an each char sequence representing a view over the window of the given size sliding along this char sequence with the given step.
 * Several last strings may have less characters than the given size.
 * Both size and step must be positive and can be greater than the number of elements in this char sequence.
 * @param size the number of elements to take in each window
 * @param step the number of elements to move the window forward by on each step, by default 1
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 */
@SinceKotlin("1.2")
public fun <R> CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): List<R> {
    checkWindowSizeStep(size, step)
    val thisSize = this.length
    val resultCapacity = thisSize / step
    if (thisSize % step == 0) resultCapacity = 0
    val result = ArrayList<R>(resultCapacity)
    var index = 0
    while (index in 0 until thisSize) {
        val end = index + size
        val coercedEnd = if (partialWindows) thisSize else end
        result.add(transform(subSequence(index, coercedEnd)))
        index += step
    }
    return result
}

/**
 * Returns a sequence of snapshots of the window of the given size sliding along this char sequence with the given step.
 * Several last strings may have less characters than the given size.
 * Both size and step must be positive and can be greater than the number of elements in this char sequence.
 * @param size the number of elements to take in each window
 * @param step the number of elements to move the window forward by on each step, by default 1
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 * @param partialWindows controls whether or not to keep partial windows in the end if any
 * @param partialWindows won't be preserved if size is a multiple of the step
 */
@SinceKotlin("1.2")
public fun CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<String> {
    return windowedSequence(size, step, partialWindows) { it.toString() }
}
public inline fun CharSequence.trimStart(predicate: (Char) -> Boolean): CharSequence {
    for (index in this.indices) if (!predicate(this[index]))
        return subSequence(index, length)
    return ""
}

/**
 * Returns a string having leading characters matching the [predicate] removed.
 */
public inline fun String.trimStart(predicate: (Char) -> Boolean): String =
    (this as CharSequence).trimStart(predicate).toString()

/**
 * Returns a sub sequence of this char sequence having leading and trailing characters from the [chars] array removed.
 */
public fun CharSequence.trim(vararg chars: Char): CharSequence = trimStart { it in chars }

/**
 * Returns a string having leading characters from the [chars] array removed.
 */
public fun String.trimStart(vararg chars: Char): String = trimStart { it in chars }

/**
 * Returns a sub sequence of this char sequence having trailing characters from the [chars] array removed.
 */
public fun CharSequence.trimEnd(vararg chars: Char): CharSequence = trimEnd { it in chars }

/**
 * Returns a string having trailing characters from the [chars] array removed.
 */
public fun String.trimEnd(vararg chars: Char): String = trimEnd { it in chars }

/**
 * Returns a sub sequence of this char sequence having leading and trailing whitespace removed.
 */
public fun CharSequence.trim(): CharSequence = trim(Char::isWhitespace)

/**
 * Returns a string having leading and trailing whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun String.trim(): String = (this as CharSequence).trim().toString()

/**
 * Returns a sub sequence of this char sequence having trailing whitespace removed.
 */
public fun CharSequence.trimEnd(): CharSequence = trimEnd(Char::isWhitespace)

/**
 * Returns a string having trailing whitespace removed.
 */
@kotlin.internal.InlineOnly
public inline fun String.trimEnd(): String = (this as CharSequence).trimEnd().toString()
consisting of ‘this’ char sequence appended with [padChar] as many times
* @sample samples.text.Strings.padEnd
* @public fun CharSequence.padEnd(length: Int, padChar: Char = ' '): String
 * @param length the desired string length.
 * @param padChar the character to pad string with, if it has length less
 * than the [length] specified. Space is used by default.
 * @return Returns a string of length at least [length]
 * consisting of ‘this’ string appended with [padChar] as many times
 * as are necessary to reach that length.
 * @sample samples.text.Strings.padEnd
 */
public fun CharSequence.padEnd(length: Int, padChar: Char = ' '): CharSequence {
    if (length < 0)
        throw IllegalArgumentException("Desired length $length is less
than zero.")
    if (length <= this.length)
        return this.subSequence(0, this.length)

    val sb = StringBuilder(length)
    sb.append(this)
    for (i in 1..(length - this.length))
        sb.append(padChar)
    return
sb.toString() * Pads the string to the specified [length] at the end with the specified character or space.
* @sample samples.text.Strings.padEnd
*/
public fun String.padEnd(length: Int, padChar: Char = ' '): String =
    (this as CharSequence).padEnd(length, padChar).toString()

/**
 * Returns `true` if this nullable char sequence is
 * either `null` or empty.
 * @sample samples.text.Strings.stringIsNullOrEmpty
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence?.isNullOrEmpty(): Boolean {
    contract {
        returns(false) implies (this@isNullOrEmpty != null)
    }

    return this == null || this.length == 0
}

/**
 * Returns `true` if this char sequence is empty (contains no characters).
 * @sample samples.text.Strings.stringIsEmpty
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.isEmpty(): Boolean = length == 0

/**
 * Returns `true` if this char sequence is not empty.
 * @sample samples.text.Strings.stringIsNotEmpty
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.isNotEmpty(): Boolean = length > 0

/**
 * Returns `true` if this char sequence is not empty and contains
 * some characters except of whitespace characters.
 * @sample samples.text.Strings.stringIsNotBlank
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.isNotBlank(): Boolean = !isBlank()

/**
 * Returns `true` if this nullable char sequence is either
 * `null` or empty or consists solely of whitespace characters.
 * @sample samples.text.Strings.stringIsNullOrBlank
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence?.isNullOrBlank(): Boolean {
    contract {
        returns(false) implies (this@isNullOrBlank != null)
    }

    return this == null || this.isBlank()
}

/**
 * Iterator for characters of the given char sequence.
 */
public operator fun CharSequence.iterator(): CharIterator = object : CharIterator() {
    private var index = 0

    public override fun nextChar(): Char = get(index++)

    public override fun hasNext(): Boolean = index < length
}

/** Returns the string if it is not `null`, or the empty string otherwise.
 */
@kotlin.internal.InlineOnly
public inline fun String?.orEmpty(): String = this ?: ""

/**
 * Returns this char sequence if it's not empty
 * or the result of calling [defaultValue] function if the char sequence is empty.
 * @sample samples.text.Strings.stringIfEmpty
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : CharSequence, C : R =
    if (isEmpty()) defaultValue() else this

/**
 * Returns this char sequence if it is not empty and doesn't consist solely of whitespace characters,
 * or the result of calling [defaultValue] otherwise.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <C, R> C.ifBlank(defaultValue: () -> R): R where C : CharSequence, C : R =
    if (isBlank()) defaultValue() else this

/**
 * Returns the range of valid character indices for this char sequence.
 */
public val CharSequence.indices: IntRange
    get() = 0..length - 1

/**
 * Returns the index of the last character in the char sequence or -1 if it is empty.
 */
public val CharSequence.lastIndex: Int
    get() = this.length - 1

/**
 * Returns the index of the last character in the char sequence or -1 if it is empty.
 */
public val CharSequence.lastIndex: Int
    get() = this.length - 1

/**
 * Returns `true` if this char sequence has Unicode surrogate pair at the specified [index].
 */
@public fun CharSequence.hasSurrogatePairAt(index: Int): Boolean {
    return (index < length) && this[index].isHighSurrogate()
}

1)\n\n/**
 * Returns a subsequence of this char sequence.
 *
 * This extension is chosen only for invocation with old-named parameters.\n * Replace parameter names with the same as those of [CharSequence.subSequence].\n *
 * @param startIndex the start index (inclusive).\n * @param endIndex the end index (exclusive). \n * If not specified, the length of the char sequence is used.\n *
 * @kotlin.internal.InlineOnly\n @Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning\n @Deprecated("Use parameters named start and end Index.") , ReplaceWith("subSequence(startIndex = start, endIndex = end)")\n */\n@OptIn(ExperimentalStdlibApi::class)\npublic inline fun String.replaceRange(startIndex: Int, endIndex: Int, replacement: String): String {
    if (endIndex < startIndex) \n        throw IndexOutOfBoundsException("End index ($endIndex) is less than start index ($startIndex).")\n    val sb = StringBuilder()\n    sb.appendRange(this, 0, startIndex) \n    sb.append(replacement) \n    sb.appendRange(this, startIndex + 1, endIndex - startIndex) \n    return sb.toString()\n}

/**
 * Returns a substring of this string from the specified range.
 */\npublic fun IntRange.substring(range: IntRange): String = \n    substring(range.start, range.endInclusive + 1)\n
/**
 * Returns a char sequence with content of this char sequence where its part at the given range is replaced with the [replacement] char sequence.
 */\n@OptIn(ExperimentalStdlibApi::class)\npublic inline fun CharSequence.replace\n @param start the start index (inclusive).\n @param endIndex the end index (exclusive).\n If not specified, the length of the char sequence is used.\n *
 @kotlin.internal.InlineOnly\n @Suppress("EXTENSION_SHADOWED_BY_MEMBER") // false warning\n @Deprecated("Use parameters named start and end Index.") , ReplaceWith("subSequence(startIndex = start, endIndex = end)")\n */\n@OptIn(ExperimentalStdlibApi::class)\npublic inline fun String.replaceRange(startIndex: Int, endIndex: Int, replacement: String): String {
    if (endIndex < startIndex) \n        throw IndexOutOfBoundsException("End index ($endIndex) is less than start index ($startIndex).")\n    val sb = StringBuilder()\n    sb.appendRange(this, 0, startIndex) \n    sb.append(replacement) \n    sb.appendRange(this, startIndex + 1, endIndex - startIndex) \n    return sb.toString()\n}
the [suffix].

* Otherwise returns this string unchanged.

* An 

* Otherwise returns this string unchanged.
* An 

* Otherwise returns this string unchanged.

* An 

* Otherwise returns this string unchanged.

* An 

* Otherwise returns this string unchanged.

* An 

* Otherwise returns this string unchanged.

* An 

* Otherwise returns this string unchanged.

* An 

* Otherwise returns this string unchanged.

* An 

* Otherwise returns this string unchanged.

* An 

* Otherwise returns this string unchanged.
The replacement string *can* consist of any combination of literal text and $-substitutions. To treat the replacement string *literally* escape it with the [kotlin.text.Regex.Companion.escapeReplacement] method.

```kotlin
@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, replacement: String): String = regex.replace(this, replacement)

/**
 * Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression with the given replacement expression, with the result of the given function [transform] that takes [MatchResult] and returns a string to be used as a replacement for that match.
 * @param replacement The replacement string, which may include literal as well as $-substitutions.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, noinline transform: (MatchResult) -> String): String =
    regex.replace(this, transform)

/**
 * Replaces the first occurrence of the given regular expression in this char sequence with the specified replacement expression.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.replaceFirst(regex: Regex, replacement: String): String =
    regex.replaceFirst(this, replacement)

/**
 * Returns `true` if this char sequence matches the given regular expression.
 */
@kotlin.internal.InlineOnly
public inline infix fun CharSequence.matches(regex: Regex): Boolean = regex.matches(this)

/**
 * Implementation of [regionMatches] for CharSequences.
 */
internal fun CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean): Boolean {
    if ((otherOffset < 0) || (thisOffset < 0) || (thisOffset > this.length - length) || (otherOffset > other.length - length)) {
        return false
    }

    for (index in 0 until length) {
        if (!this[thisOffset + index].equals(other[otherOffset + index], ignoreCase))
            return false
    }
    return true
}

/**
 * Returns `true` if this char sequence starts with the specified char.
 */
public fun CharSequence.startsWith(char: Char, ignoreCase: Boolean = false): Boolean =
    this.length > 0 &&
    this[0].equals(char, ignoreCase)

/**
 * Returns `true` if this char sequence ends with the specified char.
 */
public fun CharSequence.endsWith(char: Char, ignoreCase: Boolean = false): Boolean =
    this.length > 0 &&
    this[lastIndex].equals(char, ignoreCase)

/**
 * Returns `true` if this char sequence starts with the specified prefix.
 * @param ignoreCase `true` to ignore case when matching a character. By default `false`.
 */
public fun CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && prefix is String)
        return this.startsWith(prefix)
    else
        return regionMatchesImpl(0, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if this char sequence ends with the specified suffix.
 * @param ignoreCase `true` to ignore case when matching a character. By default `false`.
 */
public fun CharSequence.endsWith(suffix: CharSequence, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && suffix is String)
        return this.endsWith(suffix)
    else
        return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
}

/**
 * Returns the longest string `prefix` such that this char sequence and another char sequence both start with this prefix,
 * taking care not to split surrogate pairs.
 * @param ignoreCase `true` to ignore character case when matching a character. By default false.
 */
public fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {
    val shortestLength = minOf(this.length, other.length)

    var i = 0
    while (i < shortestLength &&
        this[i].equals(other[i], ignoreCase)) {
        i++
    }

    return if (this.hasSurrogatePairAt(i - 1) &&
        other.hasSurrogatePairAt(i - 1)) {
        return subSequence(0, i).toString()
    }

    return if (this[i].equals(other[i], ignoreCase)) {
        return if (this[0].equals(other[0], ignoreCase)) {
            return subSequence(0, i).toString()
        } else {
            return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
        }
    }

    val shortestLength = minOf(this.length, other.length)
    var i = 0
    while (i < shortestLength &&
        this[i].equals(other[i], ignoreCase)) {
        i++
    }

    return if (this[i].equals(other[i], ignoreCase)) {
        return if (this[0].equals(other[0], ignoreCase)) {
            return subSequence(0, i).toString()
        } else {
            return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
        }
    }

    return if (this[i].equals(other[i], ignoreCase)) {
        return if (this[0].equals(other[0], ignoreCase)) {
            return subSequence(0, i).toString()
        } else {
            return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
        }
    }

    return
}

/**
 * Returns the longest string `suffix` such that this char sequence and another char sequence both end with this suffix,
 * taking care not to split surrogate pairs.
 * @param ignoreCase `true` to ignore character case when matching a character. By default false.
 */
public fun CharSequence.commonSuffixWith(other: CharSequence, ignoreCase: Boolean = false): String {
    val thisLength = this.length
    val otherLength = other.length
    val shortestLength = minOf(thisLength, otherLength)
    var i = 0
    while (i < shortestLength &&
        this[i].equals(other[i], ignoreCase)) {
        i++
    }

    return if (this[i].equals(other[i], ignoreCase)) {
        return if (this[0].equals(other[0], ignoreCase)) {
            return subSequence(0, i).toString()
        } else {
            return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
        }
    }

    return if (this[i].equals(other[i], ignoreCase)) {
        return if (this[0].equals(other[0], ignoreCase)) {
            return subSequence(0, i).toString()
        } else {
            return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
        }
    }

    return if (this[i].equals(other[i], ignoreCase)) {
        return if (this[0].equals(other[0], ignoreCase)) {
            return subSequence(0, i).toString()
        } else {
            return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
        }
    }

    return
}
### Open Source Used In DNAC Platform Antelope 9400

Avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from

* the end

occurrence of matched string from [strings] and the string matched or `null` if none of [strings] are found.

To avoid character case when matching a string. By default `false`.

* @return A pair of an index of the first occurrence of any of the specified [chars] in this char sequence, `null` if none of [chars] are found.

* @param ignoreCase `true` to ignore character case when matching a character. By default `false`.

* @return A pair of an index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.

* @param ignoreCase `true` to ignore character case when matching a character. By default `false`.

* @return An index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.

* @param ignoreCase `true` to ignore character case when matching a character. By default `false`.

* @return A pair of an index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.

* @param ignoreCase `true` to ignore character case when matching a character. By default `false`.

* @return A pair of an index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.

* @param ignoreCase `true` to ignore character case when matching a character. By default `false`.

* @return A pair of an index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.

* @param ignoreCase `true` to ignore character case when matching a character. By default `false`.

* @return A pair of an index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.

* @param ignoreCase `true` to ignore character case when matching a character. By default `false`.
toward the beginning of this string, and finds at each position the first element in [strings]
that matches this string at that position.

*public fun CharSequence.lastIndexOfAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =
findAnyOf(strings, startIndex, ignoreCase, last = true)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the first occurrence of matched string from [strings] or -1 if none of [strings] are found.
* To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from
* the beginning to the end of this string, and finds at each position the first element in [strings]
* that matches this string at that position.

*public fun
CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Int =
findAnyOf(strings, startIndex, ignoreCase, last = true)?.first ?: -1

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of matched string from [strings] or -1 if none of [strings] are found.
* To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from
* the end toward the beginning of this string, and finds at each position the first element in [strings]
* that matches this string at that position.

*public fun
CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =
findAnyOf(strings, startIndex, ignoreCase, last = true)?.first ?: -1

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =
findAnyOf(strings, startIndex, ignoreCase, last = true)?.first ?: -1

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)

*"@param ignoreCase 'true' to ignore character case when matching a string. By default 'false'.
* @return An index of the last occurrence of the specified character from the specified
* string, starting from the specified [startIndex].

*public fun
CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =
if (other is String)
One or more strings to be used as delimiters.

- **public fun CharSequence.splitToSequence(vararg delimiters: String): Sequence<String>*

  * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.
  * @param limit The maximum number of substrings to return. Zero by default means no limit is set.
  * To avoid ambiguous results when strings in [delimiters] have characters in common, this method proceeds from \n  * the beginning to the end of this string, and finds at each position the first element in
  * [delimiters] that matches this string at that position.

  *public fun CharSequence.splitToSequence(vararg delimiters: Char): Sequence<String>*

  * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.
  * @param limit The maximum number of substrings to return. Zero by default means no limit is set.

  *public fun CharSequence.lineSequence(): Sequence<String>*

  * The lines returned do not include terminating line separators.

  *public fun CharSequence.lineSequence(): Sequence<String>*

  * The lines returned do not include terminating line separators.

- **public fun CharSequence.split(regex: Regex): List<String>*

  * The lines returned do not include terminating line separators.
CharSequence.lines(): List<String> = lineSequence().toList() /* Copyright 2010-2018 JetBrains s.r.o. and
Kotlin Programming Language contributors. * Use of this source code is governed by the Apache 2.0 license that
can be found in the license/LICENSE.txt file. */

package kotlin.text

import kotlin.js.RegExp

/**
 * Converts the characters in the specified array to a string.
 */
@SinceKotlin("1.2")
@Deprecated("Use CharArray.concatToString() instead", ReplaceWith("chars.concatToString()"))
public actual fun String(chars: CharArray): String {
    var result = ""
    for (char in chars) {
        result += char
    }
    return result
}

/**
 * Converts the characters from a portion of the specified array to a string.
 */
@SinceKotlin("1.2")
@Deprecated("Use CharArray.concatToString(startIndex, endIndex) instead", ReplaceWith("chars.concatToString(startIndex, endIndex)"))
public actual fun String(chars: CharArray, offset: Int, length: Int): String {
    if (offset < 0 || length < 0 || chars.size - offset < length)
        throw IndexOutOfBoundsException("size: $chars.size; offset: $offset; length: $length")
    var result = ""
    for (index in offset until offset + length) {
        result += chars[index]
    }
    return result
}

/**
 * Concatenates characters in this [CharArray] into a String.
 */
@SinceKotlin("1.4")
public actual fun CharArray.concatToString(): String {
    var result = ""
    for (char in this) {
        result += char
    }
    return result
}

/**
 * Concatenates characters in this [CharArray] or its subrange into a String.
 *
 * @param startIndex the beginning (inclusive) of the subrange of characters, 0 by default.
 * @param endIndex the end (exclusive) of the subrange of characters, size of this array by default.
 * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun CharArray.concatToString(startIndex: Int = 0, endIndex: Int = this.size): String {
    AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)
    return CharArray(endIndex - startIndex) { get(startIndex + it) }
}

/**
 * Returns a [CharArray] containing characters of this string.
 */
@SinceKotlin("1.4")
public actual fun String.toCharArray(): CharArray {
    return CharArray(length) { get(it) }
}

/**
 * Returns a [CharArray] containing characters of this string or its substring.
 *
 * @param startIndex the beginning (inclusive) of the substring, 0 by default.
 * @param endIndex the end (exclusive) of the substring, size of this string by default.
 * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this string.
 * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.toCharArray(startIndex: Int = 0, endIndex: Int = this.length): CharArray {
    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)
    return CharArray(endIndex - startIndex) { get(startIndex + it) }
}

/**
 * Decodes a string from the bytes in UTF-8 encoding in this array.
 */
@SinceKotlin("1.4")
public actual fun ByteArray.decodeToString(): String {
    return decodeUtf8(this, 0, size, false)
}

/**
 * Decodes a string from the bytes in UTF-8 encoding in this array or its subrange.
 *
 * @param startIndex the beginning (inclusive) of the subrange to decode, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to decode, size of this array by default.
 * @param throwOnInvalidSequence specifies whether to throw an exception on malformed byte sequence or replace it by the replacement character \uFFFD.
 * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].
 * @throws CharacterCodingException if the byte array contains malformed UTF-8 byte sequence and [throwOnInvalidSequence] is true.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun ByteArray.decodeToString(startIndex: Int = 0, endIndex: Int = this.length, throwOnInvalidSequence: Boolean = false): String {
    val size = endIndex - startIndex
    val buffer = CharBuffer(size)
    for (i in startIndex until endIndex) {
        val codePoint = this[i].toInt() and 0xFF
        if (codePoint < 0x80) {
            buffer.put((codePoint and 0x7F) or 0x00)
        } else if (codePoint < 0x800) {
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x00)
        } else if (codePoint < 0x10000) {
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x00)
        } else if (codePoint < 0x20000) {
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x00)
        } else {
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x80)
            buffer.put((codePoint and 0x7F) or 0x00)
        }
    }
    return buffer.toString()
Language contributors.* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.* Auto-generated file. DO NOT EDIT!* open source used in DNAC platform Antelope 9406
array[index++].toUInt() else throw NoSuchElementException(index.toString())
    }

    override fun contains(element: UInt): Boolean {
        // TODO: Eliminate this check after KT-30016 gets fixed.
        // Currently JS BE does not generate special bridge method for this method.
        return false
    }

    override fun containsAll(elements: Collection<UInt>): Boolean {
        // TODO: Eliminate this check after KT-30016 gets fixed.
        // Currently JS BE does not generate special bridge method for this method.
        return false
    }

    override fun isEmpty(): Boolean = this.storage.size == 0
}

/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
    return UIntArray(IntArray(size) { index ->
        init(index).toInt()
    })
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun uintArrayOf(vararg elements: UInt): UIntArray = elements

/**
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// Auto-generated file. DO NOT EDIT!

package kotlin

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class ULongArray
@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@PublishedApi
internal constructor(@PublishedApi internal val storage: LongArray) : Collection<ULong> {
    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    public constructor(size: Int) :
        this(LongArray(size))

    /**
     * Returns the array element at the given [index]. This method can be called using the index operator.
     * If [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS.
     * where the behavior is unspecified.
     */
    public operator fun get(index: Int): ULong = storage[index].toULong()

    /**
     * Sets the element at the given [index] to the given [value]. This method can be called using the index operator.
     * If [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS
     * where the behavior is unspecified.
     */
    public operator fun set(index: Int, value: ULong) {
        storage[index] = value.toLong()
    }

    /** Returns the number of elements in the array. */
    public override val size: Int get() = storage.size

    /** Creates an iterator over the elements of the array. */
    public override operator fun iterator(): ULongIterator =

    private class Iterator(private val array: LongArray) : ULongIterator() {
        private var index = 0
        override fun hasNext() = index < array.size
        override fun nextULong() = if (index < array.size)
            array[index++].toULong() else throw NoSuchElementException(index.toString())
    }

    override fun contains(element: ULong): Boolean {
        // TODO: Eliminate this check after KT-30016 gets fixed.
        // Currently JS BE does not generate special bridge method for this method.
        return false
    }

    override fun containsAll(elements: Collection<ULong>): Boolean {
        return false
    }

    override fun isEmpty(): Boolean = this.storage.size == 0
}

/**
 * Creates a new array of the specified [size], where each element is calculated by calling the specified [init] function.
 * The function [init] is called for each array element sequentially starting from the first one.
 * It should return the value for an array element given its index.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray(size: Int, init: (Int) -> ULong): ULongArray {
    return ULongArray(LongArray(size) { index ->
        init(index).toInt()
    })
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ulongArrayOf(vararg elements: ULong): ULongArray = elements

/**
 * Copyright 2010-2020 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// Auto-generated file. DO NOT EDIT!

package kotlin

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class
UShortArray

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@PublishedApi
internal constructor(@PublishedApi internal val storage: ShortArray) : Collection<UShort>()

/**
 * Creates a new array of the specified [size], with all elements initialized to zero.
 * @param size The size of the array.
 */
internal constructor(size: Int) : this(ShortArray(size))

/**
 * Returns the number of elements in the array.
 * @param element The element to check for containment.
 */
override val size: Int get() = storage.size

/** Creates an iterator over the elements of the array. */
override operator fun iterator(): UShortIterator = Iterator(storage)

private class Iterator(private val array: ShortArray) : UShortIterator() {
    private var index = 0
    // ... (rest of the class)
}

/**
 * Returns 1st *element* from the array.
 * @param index The index of the element to return.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.component1(): UInt {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * @param index The index of the element to return.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.component1(): ULong {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * @param index The index of the element to return.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component1(): UByte {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * @param index The index of the element to return.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UCharArray.component1(): UChar {
    return get(0)
}

// More code here...

// Copyright 2010-2020 JetBrains s.r.o.
// Use of this source code is governed by the Apache 2.0 license
// that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("UArraysKt")
@file:kotlin.jvm.JvmPackageName("kotlin.collections.unsigned")
package kotlin.collections
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component1(): UShort {\n    return get(0)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component2(): UInt {\n    return get(1)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component2(): ULong {\n    return get(1)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component2(): UByte {\n    return get(1)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component2(): UShort {\n    return get(1)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component3(): UInt {\n    return get(2)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component3(): ULong {\n    return get(2)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component3(): UByte {\n    return get(2)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component3(): UShort {\n    return get(2)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component4(): UInt {\n    return get(3)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component4(): ULong {\n    return get(3)\n}\n
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component4(): UByte {\n    return get(3)\n}
UShortArray.component4(): UShort {return (get(3))}/* Returns 5th *element* from the array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun UShortArray.component4(): UShort {return get(3)}* Returns 5th *element* from the array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun UIntArray.component5(): UInt {return get(4)}* Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun ULongArray.component5(): ULong {return get(4)}* Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun UByteArray.component5(): UByte {return get(4)}* Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun UShortArray.component5(): UShort {return get(4)}* Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun UIntArray.elementAt(index: Int): UInt {return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)}* Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun ULongArray.elementAt(index: Int): ULong {return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)}* Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun UByteArray.elementAt(index: Int): UByte {return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)}* Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline operator fun UShortArray.elementAt(index: Int): UShort {return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)}* Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.*/

If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS where the behavior is unspecified.
index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample samples_collections_Collections_Elements.elementAtOrNull
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.elementAtOrNull(index: Int): UInt? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample samples_collections_Collections_Elements.elementAtOrNull
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.elementAtOrNull(index: Int): ULong? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample samples_collections_Collections_Elements.elementAtOrNull
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.elementAtOrNull(index: Int): UByte? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample samples_collections_Collections_Elements.elementAtOrNull
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.elementAtOrNull(index: Int): UShort? {
    return this.getOrNull(index)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples_collections_Collections_Elements.find
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.find(predicate: (UInt) -> Boolean): UInt? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples_collections_Collections_Elements.find
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.find(predicate: (ULong) -> Boolean): ULong? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples_collections_Collections_Elements.find
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.find(predicate: (UByte) -> Boolean): UByte? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples_collections_Collections_Elements.find
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.find(predicate: (UShort) -> Boolean): UShort? {
    return firstOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples_collections_Collections_Elements.findLast
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.findLast(predicate: (UInt) -> Boolean): UInt? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples_collections_Collections_Elements.findLast
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.findLast(predicate: (ULong) -> Boolean): ULong? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples_collections_Collections_Elements.findLast
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.findLast(predicate: (UByte) -> Boolean): UByte? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples_collections_Collections_Elements.findLast
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {
    return lastOrNull(predicate)
}

Open Source Used In DNAC Platform Antelope 9411
UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {
    return lastOrNull(predicate)
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun
UByteArray.first(): UByte {
    return storage.first().toUByte()
}

/**
 * Returns first element.
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun
UShortArray.first(): UShort {
    return storage.first().toUShort()
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.firstOrNull(predicate: (UInt) -> Boolean): UInt? {
    for (element in this) if (predicate(element)) return
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.firstOrNull(): UInt? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.firstOrNull(): ULong? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.firstOrNull(): UByte? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.firstOrNull(): UShort? {
    return if (isEmpty()) null else this[0]
}
[predicate], or ‘null’ if element was not found.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.firstOrNull(predicate: (UByte) -> Boolean): UByte? {
    for (element in this) if (predicate(element))
        return element
    return null
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {
    for (element in this) if (predicate(element))
        return element
    return null
}
```

/**
 * Returns the first element matching the given [predicate], or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.getOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

/**
 * Returns the first element matching the given [predicate], or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.getOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.getOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

/**
 * Returns first index of [element], or -1 if the array does not contain element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.indexOf(element: UByte): Int {
    return storage.indexOf(element.toByte())
}
```

/**
 * Returns first index of [element], or -1 if the array does not contain element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.indexOf(element: UShort): Int {
    return storage.indexOf(element)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.getOrNull(index: Int): UInt? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```

/**
 * Returns first index of [element], or null if the [index] is out of bounds of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.getOrNull(index: Int): UShort? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
```
first index of [element], or -1 if the array does not contain element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.indexOf(element: UShort): Int {
    return storage.indexOfOf(element.toShort())
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUInt()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toULong()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUByte()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.indexOfFirst(predicate: (UShort) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUShort()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUInt()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toULong()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUByte()) }
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {
    return storage.indexOfLast { predicate(it.toUShort()) }
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.last(): UInt {
    return storage.last().toUInt()
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.last(): ULong {
    return storage.last().toULong()
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.last(): UByte {
    return storage.last().toUByte()
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.last(): UShort {
    return storage.last().toUShort()
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.last(predicate: (UInt) -> Boolean): UInt {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains
```
no element matching the predicate.

* @throws [NoSuchElementException] if no such element is found.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
ULongArray.last(predicate: (ULong) -> Boolean): ULong { 
    for (index in this.indices.reversed()) { 
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.

* @throws [NoSuchElementException] if no such element is found.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UByteArray.last(predicate: (UByte) -> Boolean): UByte { 
    for (index in this.indices.reversed()) { 
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.

* @throws [NoSuchElementException] if no such element is found.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun
UShortArray.last(predicate: (UShort) -> Boolean): UShort { 
    for (index in this.indices.reversed()) { 
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.

* @throws [NoSuchElementException] if no such element is found.

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun
UIntArray.lastIndexOf(element: UInt): Int {
    return storage.lastIndexOf(element.toInt())
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun
ULongArray.lastIndexOf(element: ULong): Int {
    return storage.lastIndexOf(element.toLong())
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun
UByteArray.lastIndexOf(element: UByte): Int {
    return storage.lastIndexOf(element.toByte())
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun
UShortArray.lastIndexOf(element: UShort): Int {
    return storage.lastIndexOf(element.toShort())
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun
UIntArray.lastOrNull(): UInt? {
    return if (isEmpty()) null else this[size - 1]
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun
ULongArray.lastOrNull(): ULong? {
    return if (isEmpty()) null else this[size - 1]
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun
UByteArray.lastOrNull(): UByte? {
    return if (isEmpty()) null else this[size - 1]
}

@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun
UShortArray.lastOrNull(): UShort? {
    return if (isEmpty()) null else this[size - 1]
}
val element = this[index] when (predicate(element)) return element
    return null

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.lastIndexOf(predicate: (UShort) -> Boolean): UShort? {
    for (index in this.indices.reversed()) {
        val element = this[index] when (predicate(element)) return element
    }
    return null
}

/**
 * Returns a random element from this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.random(): UInt {
    return random(Random())
}

/**
 * Returns a random element from this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.random(): ULong {
    return random(Random())
}

/**
 * Returns a random element from this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.random(): UByte {
    return random(Random())
}

/**
 * Returns a random element from this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.random(): UShort {
    return random(Random())
}

/**
 * Returns a random element from this array using the specified source of randomness.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public fun UIntArray.random(random: Random): UInt {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public fun ULongArray.random(random: Random): ULong {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public fun UByteArray.random(random: Random): UByte {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public fun UShortArray.random(random: Random): UShort {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return get(random.nextInt(size))
}
get(random.nextInt(size)); }\n
public fun ULongArray.randomOrNull(random: Random): ULong? {
    if (isEmpty())
        return null
    return get(random.nextInt(size)); }\n
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.single(): UInt
    {\n        var single: UInt? = null
        var found = false
        for (element in this) {
            if (predicate(element)) {
                if (found) throw IllegalArgumentException("Array contains more than one matching element.")
                single = element
                found = true
            }
        }
        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    }\n
/**
 * Returns a single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.single(predicate: (ULong) -> Boolean): ULong
    {\n        var single: ULong? = null
        var found = false
        for (element in this) {
            if (predicate(element)) {
                if (found) throw IllegalArgumentException("Array contains more than one matching element.")
                single = element
                found = true
            }
        }
        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    }\n
/**
 * Returns a single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.single(predicate: (UByte) -> Boolean): UByte
    {\n        var single: UByte? = null
        var found = false
        for (element in this) {
            if (predicate(element)) {
                if (found) throw IllegalArgumentException("Array contains more than one matching element.")
                single = element
                found = true
            }
        }
        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    }\n
/**
 * Returns a single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.single(predicate: (UShort) -> Boolean): UShort

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.single(predicate: (UShort) -> Boolean): UShort
    {\n        var single: UShort? = null
        var found = false
        for (element in this) {\n            if (predicate(element)) {\n                if (found) throw IllegalArgumentException("Array contains more than one matching element.")\n                single = element\n                found = true\n            } \n        } \n        if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")\n    }\n```

@Suppress("UNCHECKED_CAST")

```kotlin
UShortArray.single(predicate: (UShort) -> Boolean): UShort

```
require(n >= 0) { "Requested element count $n is less than zero." } return takeLast((size - n).coerceAtLeast(0))

/**
 * Returns a list containing all elements except first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.drop(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.dropLast(n: Int): List<UInt> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))
}

/**
 * Returns a list containing all elements except last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.dropLast(n: Int): List<ULong> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}
`fun UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> {
    var yielding = false
    val list = ArrayList<UInt>()
    for (item in this) {
        if (yielding) {
            list.add(item)
        } else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.dropWhile(predicate: (ULong) -> Boolean): List<ULong> {
    var yielding = false
    val list = ArrayList<ULong>()
    for (item in this) {
        if (yielding) {
            list.add(item)
        } else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {
    var yielding = false
    val list = ArrayList<UByte>()
    for (item in this) {
        if (yielding) {
            list.add(item)
        } else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {
    var yielding = false
    val list = ArrayList<UShort>()
    for (item in this) {
        if (yielding) {
            list.add(item)
        } else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Filtering.filter
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.filter(predicate: (UInt) -> Boolean): List<UInt> {
    return filterTo(ArrayList<UInt>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Filtering.filter
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {
    return filterTo(ArrayList<ULong>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Filtering.filter
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {
    return filterTo(ArrayList<UByte>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Filtering.filter
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {
    return filterTo(ArrayList<UShort>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * @param [predicate] function that takes the index of an element and the element itself and returns the result of predicate evaluation on the element.
 * 
 * @sample samples.collections.Collections.Filtering.filterIndexed
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.filterIndexed(predicate: (index: Int, UInt) -> Boolean): List<UInt> {
    return filterIndexedTo(ArrayList<UInt>(), predicate)
}

/**
 * Returns a list containing only elements matching the given [predicate].
 * @param [predicate] function that takes the index of an element and the element itself and returns the result of predicate evaluation on the element.
 * 
 * @sample samples.collections.Collections.Filtering.filterIndexed
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.filterIndexed(predicate: (index: Int, ULong) -> Boolean): List<ULong> {
    return filterIndexedTo(ArrayList<ULong>(), predicate)
}
filterIndexedTo(ArrayList<ULong>(), predicate)\n\n* Returns a list containing only elements matching the given [predicate].\n* \n* @param [predicate] function that takes the index of an element and the element itself\n* and returns the result of predicate evaluation on the element.\n* \n* @sample

samples.collections.Collections.Filtering.filterIndexedToU\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UByteArray.filterIndexedTo(predicate: (index: Int, UByte) -> Boolean): List<UByte> {\n    return \n    filterIndexedTo(ArrayList<UByte>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UShortArray.filterIndexedTo(predicate: (index: Int, UShort) -> Boolean): List<UShort> {\n    return \n    filterIndexedTo(ArrayList<UShort>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UIntegerArray.filterIndexedTo(predicate: (index: Int, UInt) -> Boolean): List<UInteger> {\n    return \n    filterIndexedTo(ArrayList<UInteger>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

ULongArray.filterIndexedTo(predicate: (index: Int, ULong) -> Boolean): List<ULong> {\n    return \n    filterIndexedTo(ArrayList<ULong>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UByteArray.filterIndexedTo(predicate: (index: Int, UByte) -> Boolean): List<UByte> {\n    return \n    filterIndexedTo(ArrayList<UByte>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UShortByteArray.filterIndexedTo(predicate: (index: Int, UShort) -> Boolean): List<UShort> {\n    return \n    filterIndexedTo(ArrayList<UShortByte>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UIntegerByteArray.filterIndexedTo(predicate: (index: Int, UInt) -> Boolean): List<UIntegerByte> {\n    return \n    filterIndexedTo(ArrayList<UIntegerByte>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

ULongByteArray.filterIndexedTo(predicate: (index: Int, ULong) -> Boolean): List<ULong> {\n    return \n    filterIndexedTo(ArrayList<ULong>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UShortByteArray.filterIndexedTo(predicate: (index: Int, UShort) -> Boolean): List<UShort> {\n    return \n    filterIndexedTo(ArrayList<UShortByte>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UIntegerByteArray.filterIndexedTo(predicate: (index: Int, UInt) -> Boolean): List<UIntegerByte> {\n    return \n    filterIndexedTo(ArrayList<UIntegerByte>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

ULongByteArray.filterIndexedTo(predicate: (index: Int, ULong) -> Boolean): List<ULong> {\n    return \n    filterIndexedTo(ArrayList<ULong>(), predicate)\n}

/**
 * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedToUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UIntegerArray.filterNot(predicate: (UByte) -> Boolean): List<UByte> {\n    return \n    filterNotToArray(ArrayList<UByte>(), predicate)\n}

/**
 * Returns a list containing all elements not matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterNotToArrayU\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UShortArray.filterNot(predicate: (UShort) -> Boolean): List<UShort> {\n    return \n    filterNotToArray(ArrayList<UShort>(), predicate)\n}

/**
 * Returns a list containing all elements not matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterNotToArrayUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

UIntegerArray.filterNot(predicate: (UInt) -> Boolean): List<UInteger> {\n    return \n    filterNotToArrayArrayList<UInteger>()\n}

/**
 * Returns a list containing all elements not matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterNotToArrayUno\n
*\n* @SinceKotlin("1.3")\n* @ExperimentalUnsignedTypes\n* @kotlin.internal.InlineOnly\n* @public inline fun

ULongArray.filterNot(predicate: (ULong) -> Boolean): List<ULong> {\n    return \n    filterNotToArrayArrayList<ULong>()\n}
samples.collections.Collections.Filtering.filter

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}UByteArray.filterNot(predicate: (UByte) -> Boolean): List<UByte> \texttt{\{_return\texttt{\textbackslash n}filterNotTo(ArrayList\texttt{<UByte>>(\texttt{)},\texttt{predicate))\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Returns a list containing all elements not matching the given [predicate].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filter

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}UShortArray.filterNot(predicate: (UShort) -> Boolean): List<UShort> \texttt{\{_return\texttt{\textbackslash n}filterNotTo(ArrayList\texttt{<UShort>>(\texttt{)},\texttt{predicate))\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements not matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}UIntArray.filterNotTo(destination: C, predicate: (UInt) -> Boolean): C \texttt{\{_for\texttt{\textbackslash n}(element in this) if (!predicate(element)) destination.add(element)\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements not matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}ULongArray.filterNotTo(destination: C, predicate: (ULong) -> Boolean): C \texttt{\{_for\texttt{\textbackslash n}(element in this) if (!predicate(element)) destination.add(element)\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements not matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}UByteArray.filterNotTo(destination: C, predicate: (UByte) -> Boolean): C \texttt{\{_for\texttt{\textbackslash n}(element in this) if (!predicate(element)) destination.add(element)\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements not matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}UShortArray.filterNotTo(destination: C, predicate: (UShort) -> Boolean): C \texttt{\{_for\texttt{\textbackslash n}(element in this) if (!predicate(element)) destination.add(element)\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements not matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C \texttt{\{_for\texttt{\textbackslash n}(element in this) if (predicate(element)) destination.add(element)\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C \texttt{\{_for\texttt{\textbackslash n}(element in this) if (predicate(element)) destination.add(element)\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C \texttt{\{_for\texttt{\textbackslash n}(element in this) if (predicate(element)) destination.add(element)\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

*\texttt{@SinceKotlin("1.3"\texttt{\textbackslash n}@\texttt{@ExperimentalUnsignedTypes\textbackslash n}@kotlin.internal.InlineOnly\texttt{\textbackslash n}public inline fun\texttt{\textbackslash n}UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C \texttt{\{_for\texttt{\textbackslash n}(element in this) if (predicate(element)) destination.add(element)\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\textbackslash n}}\texttt{\textbackslash n}**\texttt{\n}* Appends all elements matching the given [predicate] to the given [destination].\texttt{\textbackslash n} * \texttt{\textbackslash n} * \texttt{\textbackslash n} samples.collections.Collections.Filtering.filterTo

Open Source Used In DNAC Platform Antelope 9422
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: IntRange):
List<UInt> {\n    if (indices.isEmpty()) return ListOf<UInt>() \n    return copyOfRange(indices.start, indices.endInclusive + 1).asList<UInt>() \n
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: IntRange):
List<ULong> {\n    if (indices.isEmpty()) return ListOf<ULong>() \n    return copyOfRange(indices.start, indices.endInclusive + 1).asList<ULong>() \n
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: IntRange):
List<UByte> {\n    if (indices.isEmpty()) return ListOf<UByte>() \n    return copyOfRange(indices.start, indices.endInclusive + 1).asList<UByte>() \n
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: IntRange):
List<UShort> {\n    if (indices.isEmpty()) return ListOf<UShort>() \n    return copyOfRange(indices.start, indices.endInclusive + 1).asList<UShort>() \n
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: Iterable<Int>):
List<UInt> {\n    val size = indices.collectionSizeOrDefault(10) \n    if (size == 0) return emptyList<UInt>() \n    val list = ArrayList<UInt>(size) \n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list \n
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: Iterable<Int>):
List<ULong> {\n    val size = indices.collectionSizeOrDefault(10) \n    if (size == 0) return emptyList<ULong>() \n    val list = ArrayList<ULong>(size) \n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list \n
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: Iterable<Int>):
List<UByte> {\n    val size = indices.collectionSizeOrDefault(10) \n    if (size == 0) return emptyList<UByte>() \n    val list = ArrayList<UByte>(size) \n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list \n
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: Iterable<Int>):
List<UShort> {\n    val size = indices.collectionSizeOrDefault(10) \n    if (size == 0) return emptyList<UShort>() \n    val list = ArrayList<UShort>(size) \n    for (index in indices) {\n        list.add(get(index))\n    }\n    return list \n
*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sliceArray(indices: Collection<Int>):
UIntArray {\n    return UIntArray(storage.sliceArray(indices))\n}

*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sliceArray(indices: Collection<Int>):
ULongArray {\n    return ULongArray(storage.sliceArray(indices))\n}

*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sliceArray(indices: Collection<Int>):
UByteArray {\n    return UByteArray(storage.sliceArray(indices))\n}

*\@SinceKotlin("1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sliceArray(indices: Collection<Int>):
UShortArray {\n    return UShortArray(storage.sliceArray(indices))\n}
specified [indices] range.

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: IntRange): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}

/**
 * Returns a list containing first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: IntRange): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}

/**
 * Returns a list containing first [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: IntRange): UIntArray {
    return UIntArray(storage.sliceArray(indices))
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: IntRange): ULongArray {
    return ULongArray(storage.sliceArray(indices))
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: IntRange): UByteArray {
    return UByteArray(storage.sliceArray(indices))
}

/**
 * Returns a list containing last [n] elements.
 * @throws IllegalArgumentException if [n] is negative.
 * @sample samples.collections.Collections.Transformations.take
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: IntRange): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}
```kotlin
fun <T> takeLastWhile(predicate: (T) -> Boolean): List<T> {
    for (index in size downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    }
    return toList()
}

/**
 * Reverses elements in the array in-place.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reverse(): Unit {
    storage.reverse()
}

/**
 * Reverses elements in the array in-place.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reverse(): Unit {
    storage.reverse()
}
```

---

Open Source Used In DNAC Platform Antelope 9425
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reverse(): Unit {
    storage.reverse()
}

/**
 * Reverses elements in the array in-place.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reverse(): Unit {
    storage.reverse()
}

/**
 * Reverses elements of the array in the specified range in-place.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

/**
 * Reverses elements of the array in the specified range in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * 
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.reversed(): List<UInt> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.reversed(): List<ULong> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.reversed(): List<UByte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.reversed(): List<UShort> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reversedArray(): UIntArray {
    return UIntArray(storage.reversedArray())
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reversedArray(): ULongArray {
    return ULongArray(storage.reversedArray())
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reversedArray(): UByteArray {
    return UByteArray(storage.reversedArray())
}

/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reversedArray(): UShortArray {
    return UShortArray(storage.reversedArray())
}
UShortArray.reversedArray(): UShortArray {
    return UShortArray(storage.reversedArray())
}

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.shuffle(): Unit {
    shuffle(Random)
}

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.shuffle(): Unit {
    shuffle(Random)
}

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.shuffle(): Unit {
    shuffle(Random)
}

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.shuffle(): Unit {
    shuffle(Random)
}

Randomly shuffles elements in this array in-place.

/**
 * Randomly shuffles elements in this array in-place.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.shuffle(): Unit {
    shuffle(Random)
}

/**
 * Randomly shuffles elements in this array in-place.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.shuffle(): Unit {
    shuffle(Random)
}

/**
 * Randomly shuffles elements in this array in-place.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.shuffle(): Unit {
    shuffle(Random)
}

/**
 * Randomly shuffles elements in this array in-place using the specified [random]
 * instance as the source of randomness.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1)
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
}

/**
 * Randomly shuffles elements in this array in-place using the specified [random]
 * instance as the source of randomness.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1)
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
}

/**
 * Randomly shuffles elements in this array in-place using the specified [random]
 * instance as the source of randomness.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1)
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
}

/**
 * Randomly shuffles elements in this array in-place using the specified [random]
 * instance as the source of randomness.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1)
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
}

/**
 * Randomly shuffles elements in this array in-place using the specified [random]
 * instance as the source of randomness.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1)
        val j = random.nextInt(i + 1)
        val copy = this[i]
        this[i] = this[j]
        this[j] = copy
}

Sorts elements in the array in-place according to their natural sort order.

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sortDescending(): Unit {
    if (size > 1)
        sort()
        reverse()
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sortDescending(): Unit {
    if (size > 1)
        sort()
        reverse()
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sortDescending(): Unit {
    if (size > 1)
        sort()
        reverse()
}

/**
 * Sorts elements in the array in-place according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sortDescending(): Unit {
    if (size > 1)
        sort()
        reverse()
}

Returns a list of all elements sorted according to their natural sort order.

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sorted(): List<UInt> {
    return copyOf().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sorted(): List<ULong> {
    return copyOf().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sorted(): List<UByte> {
    return copyOf().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sorted():
List<UInt> { return copyOf().apply { sort() }.asList() } // Returns an array with all elements of this array sorted according to their natural sort order.

@ExperimentalUnsignedTypes fun UIntArray.sortedArray(): UIntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() } // Returns an array with all elements of this array sorted according to their natural sort order.
}

@ExperimentalUnsignedTypes fun ULongArray.sortedArray(): ULongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() } // Returns an array with all elements of this array sorted according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UByteArray.sortedArray(): UByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() } // Returns an array with all elements of this array sorted according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UShortArray.sortedArray(): UShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() } // Returns an array with all elements of this array sorted according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UIntArray.sortedArrayDescending(): UIntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() } // Returns an array with all elements of this array sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun ULongArray.sortedArrayDescending(): ULongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() } // Returns an array with all elements of this array sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UByteArray.sortedArrayDescending(): UByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() } // Returns an array with all elements of this array sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UShortArray.sortedArrayDescending(): UShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() } // Returns an array with all elements of this array sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UIntArray.sortedDescending(): List<UInt> {
    return copyOf().apply { sort() }.reversed() // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun ULongArray.sortedDescending(): List<ULong> {
    return copyOf().apply { sort() }.reversed() // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UByteArray.sortedDescending(): List<UByte> {
    return copyOf().apply { sort() }.reversed() // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UShortArray.sortedDescending(): List<UShort> {
    return copyOf().apply { sort() }.reversed() // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UIntArray.asByteArray(): ByteArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UIntArray.asList(): List<UInt> {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UByteArray.asByteArray(): ByteArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UShortArray.asList(): List<UShort> {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes expects fun UIntArray.asList(): List<UInt> {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes inline fun UByteArray.toByteArray(): ByteArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes inline fun UIntArray.toByteArray(): ByteArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes inline fun UShortArray.toByteArray(): ByteArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UIntArray.asIntArray(): IntArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun ULongArray.asIntArray(): IntArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UByteArray.asIntArray(): IntArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}

@ExperimentalUnsignedTypes fun UShortArray.asIntArray(): IntArray {
    return storage // Returns a list of all elements sorted descending according to their natural sort order.
}
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun ULongArray.asList():
    List<ULong>  

/**
 * Returns a [List] that wraps the original array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UByteArray.asList():
    List<UByte>

/**
 * Returns a [List] that wraps the original array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UShortArray.asList():
    List<UShort>

/**
 * Returns an array of type [LongArray], which is a view of this array where each element is a
 * signed reinterpretation\n * of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.asLongArray(): LongArray {
    return storage
}

/**
 * Returns an array of type [ShortArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.asShortArray(): ShortArray {
    return storage
}

/**
 * Returns an array of type [UByteArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
fun ByteArray.asUByteArray(): UByteArray {
    return UByteArray(this)
}

/**
 * Returns an array of type [UIntArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
fun IntArray.asUIntArray(): UIntArray {
    return UIntArray(this)
}

/**
 * Returns an array of type [ULongArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
fun LongArray.asULongArray(): ULongArray {
    return ULongArray(this)
}

/**
 * Returns an array of type [UShortArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
fun ShortArray.asUShortArray(): UShortArray {
    return UShortArray(this)
}

/**
 * Returns `true` if the two
 * specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the
 * same order.\n */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public infix fun UIntArray.contentEquals(other: UIntArray):
    Boolean {
        return this.contentEquals(other)
    }

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public infix fun ULongArray.contentEquals(other: ULongArray):
    Boolean {
        return this.contentEquals(other)
    }

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public infix fun UByteArray.contentEquals(other: UByteArray):
    Boolean {
        return this.contentEquals(other)
    }

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation
 warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public infix fun UShortArray.contentEquals(other: UShortArray):
    Boolean {
        return this.contentEquals(other)
    }
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public infix fun UIntArray?.contentEquals(other: UIntArray?): Boolean {
    return this?.storage.contentEquals(other?.storage)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public infix fun ULongArray?.contentEquals(other: ULongArray?): Boolean {
    return this?.storage.contentEquals(other?.storage)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public infix fun UByteArray?.contentEquals(other: UByteArray?): Boolean {
    return this?.storage.contentEquals(other?.storage)
}

/**
 * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public infix fun UShortArray?.contentEquals(other: UShortArray?): Boolean {
    return this?.storage.contentEquals(other?.storage)
}

/**
 * Returns a hash code based on the contents of this array as if it is [List].
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.contentHashCode(): Int {
    return this.contentHashCode()
}

/**
 * Returns a hash code based on the contents of this array as if it is [List].
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.contentHashCode(): Int {
    return this.contentHashCode()
}

/**
 * Returns a hash code based on the contents of this array as if it is [List].
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.contentHashCode(): Int {
    return this.contentHashCode()
}

/**
 * Returns a hash code based on the contents of this array as if it is [List].
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.contentHashCode(): Int {
    return this.contentHashCode()
}

/**
 * Returns a string representation of the contents of the specified array as if it is [List].
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.contentToString(): String {
    return this.contentToString()
}

/**
 * Returns a string representation of the contents of the specified array as if it is [List].
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.contentToString(): String {
    return this.contentToString()
}

/**
 * Returns a string representation of the contents of the specified array as if it is [List].
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.contentToString(): String {
    return this.contentToString()
}

/**
 * Returns a string representation of the contents of the specified array as if it is [List].
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.contentToString(): String {
    return this.contentToString()
}
@sample samples.collections.Arrays.ContentOperations.contentToString
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.contentToString(): String {
    return this.contentToString()
}

/**
 * Returns a string representation of the contents of the specified array as if it is [List].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.contentToString(): String {
    return this.contentToString()
}

/**
 * Returns a string representation of the contents of the specified array as if it is [List].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray?.contentToString(): String {
    return this?.joinToString(", ", "[", "]") ?: "null"
}

/**
 * Returns a string representation of the contents of the specified array as if it is [List].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray?.contentToString(): String {
    return this?.joinToString(", ", "[", "]") ?: "null"
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array to the [destination] and even specify the subrange so that it overlaps with the [destination] array.
 * 
 * @param destination the array to copy to, 0 by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * 
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.
 * 
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UIntArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array to the [destination] and even specify the subrange so that it overlaps with the [destination] array.
 * 
 * @param destination the array to copy to, 0 by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * 
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.
 * 
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.copyInto(destination: ULongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ULongArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
}
ULongArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
} * @param destination the array to copy to.

It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.

 public inline fun UByteArray.copyInto(destination: UByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UByteArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
}

public inline fun UShortArray.copyInto(destination: UShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UShortArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
}

public inline fun UIntArray.copyOf(): UIntArray {
    return UIntArray(storage.copyOf())
}

public inline fun ULongArray.copyOf(): ULongArray {
    return ULongArray(storage.copyOf())
}

public inline fun UByteArray.copyOf(): UByteArray {
    return UByteArray(storage.copyOf())
}

public inline fun UShortArray.copyOf(): UShortArray {
    return UShortArray(storage.copyOf())
}

public inline fun UIntArray.copyOf(newSize: Int): UIntArray {
    return UIntArray(storage.copyOf(newSize))
}

public inline fun UByteArrayOf(): UByteArrayOf {
    return UByteArrayOf(storage.copyOf())
}

public inline fun UShortArrayOf(): UShortArrayOf {
    return UShortArrayOf(storage.copyOf())
}

public inline fun UIntArrayOf(): UIntArrayOf {
    return UIntArrayOf(storage.copyOf())
}

public inline fun UByteArrayOf(newSize: Int): UByteArrayOf {
    return UByteArrayOf(storage.copyOf(newSize))
}

public inline fun UShortArrayOf(newSize: Int): UShortArrayOf {
    return UShortArrayOf(storage.copyOf(newSize))
}

*/
/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.copyOf(newSize: Int): ULongArray {
    return ULongArray(storage.copyOf(newSize))
}*/

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values. If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.copyOf(newSize: Int): UByteArray {
    return UByteArray(storage.copyOf(newSize))
}*/

Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary. If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyOf(newSize: Int): UShortArray {
    return UShortArray(storage.copyOf(newSize))
}*/

Returns new array which is a copy of the specified range of the original array, inclusive to copy.

/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.copyOfRange(fromIndex: Int, toIndex: Int): UIntArray {
    return UIntArray(storage.copyOfRange(fromIndex, toIndex))
}*/

Returns a new array which is a copy of the specified range of the original array.

/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.copyOfRange(fromIndex: Int, toIndex: Int): ULongArray {
    return ULongArray(storage.copyOfRange(fromIndex, toIndex))
}*/

Returns a new array which is a copy of the specified range of the original array.

/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.copyOfRange(fromIndex: Int, toIndex: Int): UByteArray {
    return UByteArray(storage.copyOfRange(fromIndex, toIndex))
}*/

Returns a new array which is a copy of the specified range of the original array.

/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {
    return UShortArray(storage.copyOfRange(fromIndex, toIndex))
}*/

Returns a new array which is a copy of the specified range of the original array.

/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.fill(element: UInt, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toInt(), fromIndex, toIndex)
}*/

Fills this array or its subrange with the specified [element] value.

/*\n\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.fill(element: UByte, fromIndex: Int = 0, toIndex: Int = size): Void {
    storage.fill(element, fromIndex, toIndex)
}*/

Fills this array or its subrange with the specified [element] value.
or its subrange with the specified [element] value.

@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

*/*}
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.fill(element: ULong,
fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toLong(), fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.fill(element: UByte,
fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toByte(), fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.fill(element: UShort,
fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toShort(), fromIndex, toIndex)
}

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UIntArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val ULongArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UByteArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UShortArray.indices: IntRange
    get() = storage.indices

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.plus(element: UInt): UIntArray {
    return UIntArray(storage + element.toInt())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.plus(element: ULong): ULongArray {
    return ULongArray(storage + element.toLong())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.plus(element: UByte): UByteArray {
    return UByteArray(storage + element.toByte())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return UShortArray(storage + element.toShort())
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntArrayקודס_9434

Open Source Used In DNAC Platform Antelope 9434
fun UIntArray.plus(elements: Collection<UInt>): UIntArray {
    var index = size
    val result = storage.copyOf(size + elements.size) * Returns an array containing all elements of the original array and then all elements of the given [elements].
    for (element in elements) result[index++] = element.toInt() * Returns an array containing all elements of the original array and then all elements of the given [elements].
    return UIntArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongArray.plus(elements: Collection<ULong>): ULongArray {
    var index = size
    val result = storage.copyOf(size + elements.size) * Returns an array containing all elements of the original array and then all elements of the given [elements].
    for (element in elements) result[index++] = element.toLong() * Returns an array containing all elements of the original array and then all elements of the given [elements].
    return ULongArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UByteArray.plus(elements: Collection<UByte>): UByteArray {
    var index = size
    val result = storage.copyOf(size + elements.size) * Returns an array containing all elements of the original array and then all elements of the given [elements].
    for (element in elements) result[index++] = element.toByte() * Returns an array containing all elements of the original array and then all elements of the given [elements].
    return UByteArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UShortArray.plus(elements: Collection<UShort>): UShortArray {
    var index = size
    val result = storage.copyOf(size + elements.size) * Returns an array containing all elements of the original array and then all elements of the given [elements].
    for (element in elements) result[index++] = element.toShort() * Returns an array containing all elements of the original array and then all elements of the given [elements].
    return UShortArray(result)
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sort(): Unit {
    if (size > 1) sortArray(this, 0, size)
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sort(): Unit {
    if (size > 1) sortArray(this, 0, size)
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sort(): Unit {
    if (size > 1) sortArray(this, 0, size)
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sort(): Unit {
    if (size > 1) sortArray(this, 0, size)
}
@sample samples.collections.Arrays.Sorting.sortRangeOfArray

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    sortArray(this, fromIndex, toIndex)
}

/**
 * Sorts a range in the array in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than
 * the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 * 
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray
 * 
 * @SinceKotlin("1.4")
 * @ExperimentalUnsignedTypes
 * @public fun UByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
 *     AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
 *     sortArray(this, fromIndex, toIndex)
 * }
[IntArray], which is a copy of this array where each element is a signed reinterpretation of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.toIntArray(): IntArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [LongArray], which is a copy of this array where each element is a signed reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.toLongArray(): LongArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [ShortArray], which is a copy of this array where each element is a signed reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.toShortArray(): ShortArray {
    return storage.copyOf()
}

/**
 * Returns a typed object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.toTypedArray(): Array<UInt> {
    return Array(size) { index -> this[index] }
}

/**
 * Returns a typed object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.toTypedArray(): Array<ULong> {
    return Array(size) { index -> this[index] }
}

/**
 * Returns an array of type [UByteArray], which is a copy of this array where each element is an unsigned reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ByteArray.toUByteArray(): UByteArray {
    return UByteArray(this.copyOf())
}

/**
 * Returns an array of type [UIntArray], which is a copy of this array where each element is an unsigned reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.toUIntArray(): UIntArray {
    return UIntArray(this.copyOf())
}

/**
 * Returns an array of type [ULongArray], which is a copy of this array where each element is an unsigned reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun LongArray.toULongArray(): ULongArray {
    return ULongArray(this.copyOf())
}

/**
 * Returns an array of type [UShortArray], which is a copy of this array where each element is an unsigned reinterpretation of the corresponding element of this array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ShortArray.toUShortArray(): UShortArray {
    return UShortArray(this.copyOf())
}

/**
 * Returns a [Map]
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ShortArray.toUShortArray(): UShortArray {
    return UShortArray(this.copyOf())
}
```
where keys are elements from the given array and values are \n * produced by the \[valueSelector\] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith

```kotlin
*\@Since Kotlin("1.4")\@ExperimentalUnsignedTypes\n\@kotlin.internal.InlineOnly\n\@public inline fun <V>
UIntArray.associateWith(valueSelector: (UInt) -> V): Map<UInt, V> {\n    \n    val result = LinkedHashMap<UInt, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n*\n
\n
*\@Since Kotlin("1.4")\@ExperimentalUnsignedTypes\n\@kotlin.internal.InlineOnly\n\@public inline fun <V>
ULongArray.associateWith(valueSelector: (ULong) -> V): Map<ULong, V> {\n    \n    val result = LinkedHashMap<ULong, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n*\n
\n
*\@Since Kotlin("1.4")\@ExperimentalUnsignedTypes\n\@kotlin.internal.InlineOnly\n\@public inline fun <V>
UByteArray.associateWith(valueSelector: (UByte) -> V): Map<UByte, V> {\n    \n    val result = LinkedHashMap<UByte, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n*\n
\n
*\@Since Kotlin("1.4")\@ExperimentalUnsignedTypes\n\@kotlin.internal.InlineOnly\n\@public inline fun <V>
UShortArray.associateWith(valueSelector: (UShort) -> V): Map<UShort, V> {\n    \n    val result = LinkedHashMap<UShort, V>(mapCapacity(size).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}\n*\n
```

```kotlin
*\@Since Kotlin("1.4")\@ExperimentalUnsignedTypes\n\@kotlin.internal.InlineOnly\n\@public inline fun <V, M : MutableMap<in UInt, in V>> UIntArray.associateWithTo(destination: M, valueSelector: (UInt) -> V): M {\n    \n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n*\n
```

```kotlin
*\@Since Kotlin("1.4")\@ExperimentalUnsignedTypes\n\@kotlin.internal.InlineOnly\n\@public inline fun <V, M : MutableMap<in ULong, in V>> ULongArray.associateWithTo(destination: M, valueSelector: (ULong) -> V): M {\n    \n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n*\n
```

```kotlin
*\@Since Kotlin("1.4")\@ExperimentalUnsignedTypes\n\@kotlin.internal.InlineOnly\n\@public inline fun <V, M : MutableMap<in UByte, in V>> UByteArray.associateWithTo(destination: M, valueSelector: (UByte) -> V): M {\n    \n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n*\n
```
for (element in this) {
    destination.put(element, valueSelector(element))
} return destination

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of
 * the given array, where key is the element itself and value is provided by the [valueSelector] function applied
 * to that key. If any two elements are equal, the last one overwrites the former value in the map.
 */
@sample samples.collections.Collections.Transformations.associateWithTo

/*@SinceKotlin("1.4")*/
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <V, M : MutableMap<in UShort, in V>> UShortArray.associateWithTo(destination: M, valueSelector: (UShort) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being
 * invoked on each element of original array.
 */
@sample samples.collections.Collections.Transformations.flatMap

/*@SinceKotlin("1.3")*/
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <R>
    UIntArray.flatMap(transform: (UInt) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being
 * invoked on each element of original array.
 */
@sample samples.collections.Collections.Transformations.flatMap

/*@SinceKotlin("1.3")*/
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <R>
    ULongArray.flatMap(transform: (ULong) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being
 * invoked on each element of original array.
 */
@sample samples.collections.Collections.Transformations.flatMap

/*@SinceKotlin("1.3")*/
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <R>
    UByteArray.flatMap(transform: (UByte) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being
 * invoked on each element and its index in the original array.
 */
@sample samples.collections.Collections.Transformations.flatMapIndexed

/*@SinceKotlin("1.4")*/
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <R>
    UIntArray.flatMapIndexed(transform: (index: Int, UInt) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being
 * invoked on each element and its index in the original array.
 */
@sample samples.collections.Collections.Transformations.flatMapIndexed

/*@SinceKotlin("1.4")*/
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <R>
    ULongArray.flatMapIndexed(transform: (index: Int, ULong) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being
 * invoked on each element and its index in the original array.
 */
@sample samples.collections.Collections.Transformations.flatMapIndexed

/*@SinceKotlin("1.4")*/
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <R>
    UByteArray.flatMapIndexed(transform: (index: Int, UByte) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}
element of original array, to the given [destination].

```kotlin
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <R, C : MutableCollection<in R>> UShortArray.flatMapTo(destination: C, transform: (UShort) -> Iterable<R>): C {
    var list = transform(element)
    destination.addAll(list)
    return destination
}

*@
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <K>
UIntArray.groupBy(keySelector: (UInt) -> K): Map<K, List<UInt>> {
    return groupByTo(LinkedHashMap<K, MutableList<UInt>>(), keySelector)
}

*@
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <K>
ULongArray.groupBy(keySelector: (ULong) -> K): Map<K, List<ULong>> {
    return groupByTo(LinkedHashMap<K, MutableList<ULong>>(), keySelector)
}

*@
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <K>
UByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {
    return groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)
}

*@
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <K>
UShortArray.groupBy(keySelector: (UShort) -> K): Map<K, List<UShort>> {
    return groupByTo(LinkedHashMap<K, MutableList<UShort>>(), keySelector)
}
```

The returned map preserves the entry iteration order of the keys produced from the original array and the list of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the original array and the list of corresponding elements.

*@
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <K>
UIntArray.groupByKeysAndValues(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

*@
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <K>
ULongArray.groupByKeysAndValues(keySelector: (ULong) -> K, valueTransform: (ULong) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

*@
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <K>
UByteArray.groupByKeysAndValues(keySelector: (UByte) -> K, valueTransform: (UByte) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}

*@
@SinceKotlin("1.3."
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <K>
UShortArray.groupByKeysAndValues(keySelector: (UShort) -> K, valueTransform: (UShort) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
```
with a list of corresponding values.
The returned map preserves the entry iteration order of the keys produced from the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, V> UByteArray.groupBy(keySelector: (UByte) -> K, valueTransform: (UByte) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, V> UShortArray.groupBy(keySelector: (UShort) -> K, valueTransform: (UShort) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, M : MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<UInt>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, M : MutableMap<in K, MutableList<ULong>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<ULong>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, M : MutableMap<in K, MutableList<UByte>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<UByte>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, M : MutableMap<in K, MutableList<UShort>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<UShort>() }
        list.add(element)
    }
    return destination
}
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <K, V> samples.collections.Collections.Transformations.groupByKeysAndValues
```
valueTransform: (UInt) -> V): M { 'n for (element in this) { 'n val key = keySelector(element) 'n val list = destination.getOrPut(key) { ArrayList<V>() } 'n list.add(valueTransform(element)) 'n } 'n return destination 'n } 'n /**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 * @param 
 * @return The [destination] map.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K, valueTransform: (ULong) -> V): M { 'n for (element in this) { 'n val key = keySelector(element) 'n val list = destination.getOrPut(key) { ArrayList<V>() } 'n list.add(valueTransform(element)) 'n } 'n return destination 'n } 'n /**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 * @param 
 * @return The [destination] map.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K, valueTransform: (UByte) -> V): M { 'n for (element in this) { 'n val key = keySelector(element) 'n val list = destination.getOrPut(key) { ArrayList<V>() } 'n list.add(valueTransform(element)) 'n } 'n return destination 'n } 'n /**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 * @param 
 * @return The [destination] map.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K, valueTransform: (UShort) -> V): M { 'n for (element in this) { 'n val key = keySelector(element) 'n val list = destination.getOrPut(key) { ArrayList<V>() } 'n list.add(valueTransform(element)) 'n } 'n return destination 'n } 'n /**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @param 
 * @return The list.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.map(transform: (UInt) -> R): List<R> { 'n return mapTo(ArrayList<R>(size), transform) 'n } 'n /**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @param 
 * @return The list.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.map(transform: (ULong) -> R): List<R> { 'n return mapTo(ArrayList<R>(size), transform) 'n } 'n /**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @param 
 * @return The list.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.map(transform: (UByte) -> R): List<R> { 'n return mapTo(ArrayList<R>(size), transform) 'n } 'n /**
 * Returns a list containing the results of applying the given [transform] function
 * to each element in the original array.
 * @param 
 * @return The list.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.map(transform: (UShort) -> R): List<R> { 'n return mapTo(ArrayList<R>(size), transform) 'n }
given [transform] function to each element and its index in the original array. * @param [transform] function that takes the index of an element and the element itself* and returns the result of the transform applied to the element.

*  

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {
    var index = 0
    for (item in this)
        destination.add(transform(index++, item))
    return destination
}

<code>@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableList<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}
function to each element of the original array
* and appends the results to the given [destination].

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun <R, C : MutableCollection<R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {
    for (item in this)
        return destination.add(transform(item))
    return destination
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun <R, C : MutableCollection<R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UIntArray.withIndex(): Iterable<IndexedValue<UInt>> {
    return IndexingIterable { iterator() }
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun ULongArray.withIndex(): Iterable<IndexedValue<ULong>> {
    return IndexingIterable { iterator() }
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UByteArray.withIndex(): Iterable<IndexedValue<UByte>> {
    return IndexingIterable { iterator() }
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UShortArray.withIndex(): Iterable<IndexedValue<UShort>> {
    return IndexingIterable { iterator() }
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UIntArray.all(predicate: (UInt) -> Boolean): Boolean {
    for (element in this)
        if (!predicate(element)) return false
    return true
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun ULongArray.all(predicate: (ULong) -> Boolean): Boolean {
    for (element in this)
        if (!predicate(element)) return false
    return true
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UByteArray.all(predicate: (UByte) -> Boolean): Boolean {
    for (element in this)
        if (!predicate(element)) return false
    return true
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline fun UShortArray.all(predicate: (UShort) -> Boolean): Boolean {
    for (element in this)
        if (!predicate(element)) return false
    return true
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UIntArray.any(): Boolean {
    return storage.any()
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun ULongArray.any(): Boolean {
    return storage.any()
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UByteArray.any(): Boolean {
    return storage.any()
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UShortArray.any(): Boolean {
    return storage.any()
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UIntArray.any(): Boolean {
    return storage.any()
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun ULongArray.any(): Boolean {
    return storage.any()
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UByteArray.any(): Boolean {
    return storage.any()
}

*\n@Since Kotlin("1.3.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic fun UShortArray.any(): Boolean {
    return storage.any()
}
the given [predicate].

*\@Since Kotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@public inline fun UIntArray.any(predicate: (UInt) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.any(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.any(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.any(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.count(predicate: (UInt) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.count(predicate: (ULong) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.count(predicate: (UByte) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.count(predicate: (UShort) -> Boolean): Int {
    var count = 0
    for (element in this) if (predicate(element)) ++count
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Returns the number of elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Returns the number of elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Returns the number of elements matching the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.fold(initial: R, operation: (acc: R, UShort) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Returns the specified [initial] value if the array is empty.
 * 
 * @param [operation] function that takes the current accumulator value
 * and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public fun UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Returns the specified [initial] value if the array is empty.
 * 
 * @param [operation] function that takes the current accumulator value
 * and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public fun ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Returns the specified [initial] value if the array is empty.
 * 
 * @param [operation] function that takes the current accumulator value
 * and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public fun UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

/**
 * Returns the specified [initial] value if the array is empty.
 * 
 * @param [operation] function that takes the current accumulator value
 * and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public fun UShortArray.fold(initial: R, operation: (acc: R, UShort) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
takes current accumulator value and an element, and calculates the next accumulator value.

```java
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R>
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}
```

```java
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R>
UIntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

```java
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R>
ULongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

```java
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R>
UByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

```java
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R>
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}
```

```java
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R>
UIntArray.foldRight(initial: R, operation: (UInt, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

```java
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R>
ULongArray.foldRight(initial: R, operation: (ULong, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```

```java
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun <R>
UShortArray.foldRight(initial: R, operation: (UShort, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
```
@SinceKotlin("1.3.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.foldRightIndexed(initial: R, operation: (Int, UByte) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.forEach(action: (ULong) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.forEach(action: (UByte) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.forEach(action: (UShort) -> Unit): Unit {
    for (element in this) action(element)
}
Performs the given \[\text{action}\] on each element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.forEach(action: (UByte) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given \[\text{action}\] on each element, providing sequential index with the element.\n * \param \[\text{action}\] function that takes the index of an element and the element itself\n * performs the action on the element.\n */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.forEachIndexed(action: (Int, UShort) -> Unit): Unit {
    var index = 0
    for (item in this)
        action(index++, item)
}

@Deprecated("Use maxOrNull instead.",
ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun
UIntArray.max(): UInt? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.",
ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun
ULongArray.max(): ULong? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.",
ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun
UByteArray.max(): UByte? {
    return maxOrNull()
}

@Deprecated("Use maxByOrNull instead.",
ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
<R : Comparable<R>> UIntArray.maxBy(selector: (UInt) -> R): UInt? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.",
ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
<R : Comparable<R>> ULongArray.maxBy(selector: (ULong) -> R): ULong? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.",
ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
<R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.",
ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
<R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort? {
    return maxByOrNull(selector)
}
```

Open Source Used In DNAC Platform Antelope 9449
<R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte? { return maxByOrNull(selector)[0] * Returns the first element yielding the largest value of the given function or `null` if there are no elements.

@SinceKotlin("1.4")@@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline fun <R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort? { return maxByOrNull(selector)

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")@@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline fun UIntArray.maxByOrNull(selector: (UInt) -> R): UInt? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v)
            { maxElem = e
            maxValue = v
        }
    }
    return maxElem

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")@@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline fun ULongArray.maxByOrNull(selector: (ULong) -> R): ULong? { return maxByOrNull(selector)

/**
 * Returns the largest value among all values produced by 
 * [selector] function applied to each element in the array.
 * 
 * @throws NoSuchElementException if the array is empty.
 * 
 * @sample @OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
 * ByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline fun UIntArray.maxOf(selector: (UInt) -> Double): Double? {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue

/**
 * Returns the largest value among all values produced by 
 * [selector] function applied to each element in the array.
 * 
 * @throws NoSuchElementException if the array is empty.
 * 
 * @sample @OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
 * ByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly@public inline fun ULongArray.maxOf(selector: (ULong) -> Double): Double? {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue

by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.maxOf(selector: (UByte) -> Double): Double {\n    val maxMap = selector(this[0])\n    for (i in 1 .. lastIndex) {\n        val v = selector(this[i])\n        maxMap = maxOf(maxMap, v)\n    }\n    return maxMap\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.maxOf(selector: (UShort) -> Double): Double {\n    val maxMap = selector(this[0])\n    for (i in 1 .. lastIndex) {\n        val v = selector(this[i])\n        maxMap = maxOf(maxMap, v)\n    }\n    return maxMap\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.maxOf(selector: (UInt) -> Float): Float {\n    val maxMap = selector(this[0])\n    for (i in 1 .. lastIndex) {\n        val v = selector(this[i])\n        maxMap = maxOf(maxMap, v)\n    }\n    return maxMap\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.maxOf(selector: (ULong) -> Float): Float {\n    val maxMap = selector(this[0])\n    for (i in 1 .. lastIndex) {\n        val v = selector(this[i])\n        maxMap = maxOf(maxMap, v)\n    }\n    return maxMap\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArrayList.maxOf(selector: (UByte) -> Float): Float {\n    val maxMap = selector(this[0])\n    for (i in 1 .. lastIndex) {\n        val v = selector(this[i])\n        maxMap = maxOf(maxMap, v)\n    }\n    return maxMap\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArrayList.maxOf(selector: (UShort) -> Float): Float {\n    val maxMap = selector(this[0])\n    for (i in 1 .. lastIndex) {\n        val v = selector(this[i])\n        maxMap = maxOf(maxMap, v)\n    }\n    return maxMap\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArrayList.maxOf(selector: (UInt) -> Float): Float {\n    val maxMap = selector(this[0])\n    for (i in 1 .. lastIndex) {\n        val v = selector(this[i])\n        maxMap = maxOf(maxMap, v)\n    }\n    return maxMap\n}

/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArrayList.maxOf(selector: (ULong) -> Float): Float {\n    val maxMap = selector(this[0])\n    for (i in 1 .. lastIndex) {\n        val v = selector(this[i])\n        maxMap = maxOf(maxMap, v)\n    }\n    return maxMap\n}
NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
val ULongArray.maxOf(selector: (ULong) -> R): R { if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
val UByteArray.maxOf(selector: (UByte) -> R): R { if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
val UShortArray.maxOf(selector: (UShort) -> R): R { if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
val UIntArray.maxOfOrNull(selector: (UInt) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
val ULongArray.maxOfOrNull(selector: (ULong) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
val UByteArray.maxOfOrNull(selector: (UByte) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
public inline fun UShortArray.maxOfOrNull(selector: (UShort) -> Double): Double? {
    if (isEmpty()) return null

    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.maxOfOrNull(selector: (UInt) -> Float): Float? {
    if (isEmpty()) return null

    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.maxOfOrNull(selector: (ULong) -> Float): Float? {
    if (isEmpty()) return null

    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.maxOfOrNull(selector: (UByte) -> Float): Float? {
    if (isEmpty()) return null

    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UIntArray.maxOfOrNull(selector: (UInt) -> R): R? {
    if (isEmpty()) return null

    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.maxOfOrNull(selector: (ULong) -> R): R? {
    if (isEmpty()) return null

    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.maxOfOrNull(selector: (UByte) -> R): R? {
    if (isEmpty()) return null

    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}
maxValue = selector(this[0]) \n for (i in 1..lastIndex) { \n   val v = selector(this[i]) \n   if (maxValue < v) { \n     maxValue = v \n   } \n } \n return maxValue \n
/**
 * Returns the largest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.maxOrNull(selector: (UShort) -> R): R? { \n   if (isEmpty()) return null \n   var maxValue = selector(this[0]) \n   for (i in 1..lastIndex) { \n     val v = selector(this[i]) \n     if (maxValue < v) { \n       maxValue = v \n     } \n   } \n   return maxValue \n}

/**
 * Returns the largest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.maxOrNull(selector: (ULong) -> R): R? { \n   if (isEmpty()) return null \n   var maxValue = selector(this[0]) \n   for (i in 1..lastIndex) { \n     val v = selector(this[i]) \n     if (maxValue < v) { \n       maxValue = v \n     } \n   } \n   return maxValue \n}

/**
 * Returns the largest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.maxOrNull(selector: (UByte) -> R): R? { \n   if (isEmpty()) return null \n   var maxValue = selector(this[0]) \n   for (i in 1..lastIndex) { \n     val v = selector(this[i]) \n     if (maxValue < v) { \n       maxValue = v \n     } \n   } \n   return maxValue \n}

/**
 * Returns the largest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.maxOrNull(selector: (UInt) -> R): R? { \n   if (isEmpty()) return null \n   var maxValue = selector(this[0]) \n   for (i in 1..lastIndex) { \n     val v = selector(this[i]) \n     if (maxValue < v) { \n       maxValue = v \n     } \n   } \n   return maxValue \n}

/**
 * Returns the largest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.maxOrNull(selector: (UShort) -> R): R? { \n   if (isEmpty()) return null \n   var maxValue = selector(this[0]) \n   for (i in 1..lastIndex) { \n     val v = selector(this[i]) \n     if (maxValue < v) { \n       maxValue = v \n     } \n   } \n   return maxValue \n}
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
ULongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
UByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
UShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.maxOrNull(): UInt? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        var e = this[i]
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.maxOrNull(): ULong? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        var e = this[i]
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.maxOrNull(): UByte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        var e = this[i]
        if (max < e) max = e
    }
    return max
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.maxOrNull(): UShort? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        var e = this[i]
        if (max < e) max = e
    }
    return max
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
public fun UIntArray.maxWith(comparator: Comparator<in UInt>): UInt? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
public fun ULongArray.maxWith(comparator: Comparator<in ULong>): ULong? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
public fun UByteArray.maxWith(comparator: Comparator<in UByte>): UByte? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
    return maxWithOrNull(comparator)
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.maxWithOrNull(comparator: Comparator<in UInt>): UInt? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
public fun ULongArray.maxWithOrNull(comparator: Comparator<in ULong>): ULong? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
public fun UByteArray.maxWithOrNull(comparator: Comparator<in UByte>): UByte? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
public fun UShortArray.maxWithOrNull(comparator: Comparator<in UShort>): UShort? {
    return maxWithOrNull(comparator)
}
Comparator<in UInt>): UInt? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.maxWithOrNull(comparator: Comparator<in ULong>): ULong? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.maxWithOrNull(comparator: Comparator<in UByte>): UByte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.maxWithOrNull(comparator: Comparator<in UShort>): UShort? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.min(): UInt? {
    return minOrNull()
}

/**
 * Returns the first element having the smallest value according to the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.min(): UByte? {
    return minOrNull()
}

/**
 * Returns the first element having the smallest value according to the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.min(): UShort? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.min(): ULong? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.min(): UInt? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.min(): UByte? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.min(): UShort? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun IntArray.min(): Int? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun LongArray.min(): Long? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun CharArray.min(): Char? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun StringArray.min(): String? {
    return minOrNull()
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun IntArray.minBy(selector: (Int) -> Int): Int? {
    return minByOrNull(selector)
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun LongArray.minBy(selector: (Long) -> Long): Long? {
    return minByOrNull(selector)
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun CharArray.minBy(selector: (Char) -> Char): Char? {
    return minByOrNull(selector)
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun StringArray.minBy(selector: (String) -> String): String? {
    return minByOrNull(selector)
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.minByOrNull(selector: (UInt) -> UInt): UInt? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (minValue > v)
```kotlin
/**
 * Returns the smallest value among all values produced by 
 * [selector] function\n * applied to each element in the array.\n * \n * @sample samples.collections.Collections.Aggregates.minOf
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.minOf(selector: (UInt) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced 
 * by [selector] function\n * applied to each element in the array.\n * \n * @sample samples.collections.Collections.Aggregates.minOf
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.minOf(selector: (ULong) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced 
 * by [selector] function\n * applied to each element in the array.\n * \n * @sample samples.collections.Collections.Aggregates.minOf
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.minOf(selector: (UByte) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}
```
UShortArray.minOf(selector: (UShort) -> Double): Double

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    minValue = minOf(minValue, v)
}

return minValue

// Returns the smallest value among all values produced by [selector] function
// applied to each element in the array.

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UintArray.minOf(selector: (UInt) -> Float): Float

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    minValue = minOf(minValue, v)
}

return minValue

// Returns the smallest value among all values produced by [selector]
// function * applied to each element in the array. 

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.minOf(selector: (ULong) -> Float): Float

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    minValue = minOf(minValue, v)
}

return minValue

// Returns the smallest value among all values produced by [selector]
// function * applied to each element in the array.

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.minOf(selector: (UByte) -> Float): Float

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    minValue = minOf(minValue, v)
}

return minValue

// Returns the smallest value among all values produced by [selector]
// function * applied to each element in the array.

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.minOf(selector: (UShort) -> Float): Float

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    minValue = minOf(minValue, v)
}

return minValue

// Returns the smallest value among all values produced by [selector]
// function * applied to each element in the array.

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UIntArray.minOf(selector: (UInt) -> R): R

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    if (minValue > v) {
        minValue = v
    }
}

return minValue

// Returns the smallest value among all values produced by [selector]
// function * applied to each element in the array.

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.minOf(selector: (ULong) -> R): R

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    if (minValue > v) {
        minValue = v
    }
}

return minValue

// Returns the smallest value among all values produced by [selector]
// function * applied to each element in the array.

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.minOf(selector: (UByte) -> R): R

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    if (minValue > v) {
        minValue = v
    }
}

return minValue

// Returns the smallest value among all values produced by [selector]
// function * applied to each element in the array.

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.minOf(selector: (UShort) -> R): R

var minValue = selector(this[0])

for (i in 1..lastIndex) {
    val v = selector(this[i])
    if (minValue > v) {
        minValue = v
    }
}

return minValue

// Returns the smallest value among all values produced by [selector]
// function * applied to each element in the array.

// If any of values produced by [selector] function is 'NaN',
// the returned result is 'NaN'.
// @throws NoSuchElementException if the array is empty.

/*
 * @SinceKotlin("1.4")
 * @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
 * @OverloadResolution
 */

ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> open inline fun
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.minOf(selector: (UByte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function.* applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.minOf(selector: (UShort) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function.* applied to each element in the array or `null` if there are no elements.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.minOfOrNull(selector: (UInt) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function.* applied to each element in the array or `null` if there are no elements.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.minOfOrNull(selector: (ULong) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function.* applied to each element in the array or `null` if there are no elements.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.minOfOrNull(selector: (UByte) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function.* applied to each element in the array or `null` if there are no elements.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.minOfOrNull(selector: (UShort) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function.* applied to each element in the array or `null` if there are no elements.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.minOfOrNull(selector: (UInt) -> Float): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}
function is 'NaN', the returned result is 'NaN'.

*\@Since Kotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@Unsignature\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
ULongArray.minOfOrNull(selector: (ULong) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue-field **temp** * Returns the smallest value among all values produced by [selector] function\n applied to each element in the array or 'null' if there are no elements.\n * * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.

*\@Since Kotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@Unsignature\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UByteArray.minOfOrNull(selector: (UByte) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue-field **temp** * Returns the smallest value among all values produced by [selector] function\n applied to each element in the array or 'null' if there are no elements.\n * * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.

*\@Since Kotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@Unsignature\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UShortArray.minOfOrNull(selector: (UShort) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue-field **temp** * Returns the smallest value among all values produced by [selector] function\n applied to each element in the array or 'null' if there are no elements.\n * * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.

*\@Since Kotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@Unsignature\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UIntArray.minOfOrNull(selector: (UInt) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) \n minValue = v\n }\n return minValue-field **temp** * Returns the smallest value among all values produced by [selector] function\n applied to each element in the array or 'null' if there are no elements.\n * * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.

*\@Since Kotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@Unsignature\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
ULongArray.minOfOrNull(selector: (ULong) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) \n minValue = v\n }\n return minValue-field **temp** * Returns the smallest value among all values produced by [selector] function\n applied to each element in the array or 'null' if there are no elements.\n * * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.

*\@Since Kotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@Unsignature\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UByteArray.minOfOrNull(selector: (UByte) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) \n minValue = v\n }\n return minValue-field **temp** * Returns the smallest value among all values produced by [selector] function\n applied to each element in the array or 'null' if there are no elements.\n * * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.

*\@Since Kotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@Unsignature\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun
UShortArray.minOfOrNull(selector: (UShort) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) \n minValue = v\n }\n return minValue-field **temp** * Returns the smallest value according to the provided [comparator] function\n among all values produced by [selector] function applied to each element in the array.\n * * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.

*\@Since Kotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@Unsignature\@ExperimentalUnsignedTypes\@kotlin.internal.InlineOnly\public inline fun <R>
UIntArray.minOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 **/ * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")*\n*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*\n*\@OverloadResolution

ByLambdaReturnType\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly*\n*\@public inline fun <R> ULongArray.minOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
 **/ * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")*\n*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*\n*\@OverloadResolution

ByLambdaReturnType\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly*\n*\@public inline fun <R> UByteArray.minOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 **/ * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")*\n*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*\n*\@OverloadResolution

ByLambdaReturnType\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly*\n*\@public inline fun <R> UShortArray.minOfWith(nullOrEmpty(comparator: Comparator<in R>, selector: (UShort) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 **/ * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")*\n*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*\n*\@OverloadResolution

ByLambdaReturnType\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly*\n*\@public inline fun <R> UIntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 **/ * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")*\n*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*\n*\@OverloadResolution

ByLambdaReturnType\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly*\n*\@public inline fun <R> ULongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 **/ * @throws NoSuchElementException if the array is empty.

*\@SinceKotlin("1.4")*\n*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*\n*\@OverloadResolution

ByLambdaReturnType\n*\@ExperimentalUnsignedTypes\n*\@kotlin.internal.InlineOnly*\n*\@public inline fun <R> UByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 **/ * @throws NoSuchElementException if the array is empty.
Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@ExperimentalUnsignedTypes\@Unpublished inline fun <R> UShortArray.minOfOrNull(comparator: Comparator\<in R>, selector: (UShort) -> R): R? {
    if (isNullOrEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}
```

Returns the smallest element or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UIntArray.minOrNull(): UInt? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

Returns the smallest element or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun ULongArray.minOrNull(): ULong? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

Returns the smallest element or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UByteArray.minOrNull(): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

Returns the smallest element or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UShortArray.minOrNull(): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

Returns the smallest element or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UByteArray.minOrNull(): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

Returns the smallest element or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UShortArray.minOrNull(): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}
```

Returns the first element having the smallest value according to the provided [comparator] or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UIntArray.minWithOrNull(comparator: Comparator\<in UInt>): UInt? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
```

Returns the first element having the smallest value according to the provided [comparator] or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun ULongArray.minWithOrNull(comparator: Comparator\<in ULong>): ULong? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
```

Returns the first element having the smallest value according to the provided [comparator] or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UByteArray.minWithOrNull(comparator: Comparator\<in UByte>): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
```

Returns the first element having the smallest value according to the provided [comparator] or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UShortArray.minWithOrNull(comparator: Comparator\<in UShort>): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
```

Returns the first element having the smallest value according to the provided [comparator] or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UByteArray.minWithOrNull(comparator: Comparator\<in UByte>): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
```

Returns the first element having the smallest value according to the provided [comparator] or null if there are no elements.

```kotlin
*\@SinceKotlin("1.4")\@Unpublished fun UShortArray.minWithOrNull(comparator: Comparator\<in UShort>): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
```
Comparator<in UShort>): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        var e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns `true` if the array has no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.none
 *
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.none
 *
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.none
 *
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if no elements match the given 
 * [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 *
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.none(predicate: (UInt) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given 
 * [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 *
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.none(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given 
 * [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 *
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given 
 * [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 *
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Performs the given [action] on each element and returns the array itself
 * afterwards.
 * 
 * @param [action] function that takes the index of an element
 * and performs the action on the element.
 * 
 * @sample DNAC.UnsignedArray,onEachIndexed
 *
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.onEachIndexed(action: (index: Int, UInt) -> Unit): UIntArray {
    return apply {
        forEachIndexed { action(it, it) }
    }
}

/**
 * Performs the given [action] on each element and returns the array itself
 * afterwards.
 * 
 * @param [action] function that takes the index of an element
 * and performs the action on the element.
 * 
 * @sample DNAC. UnsignedArray, onEachIndexed
 *
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.onEachIndexed(action: (index: Int, ULong) -> Unit): ULongArray {
    return apply {
        forEachIndexed { action(it, it) }
    }
}

/**
 * Performs the given [action] on each element and returns the array itself
 * afterwards.
 * 
 * @param [action] function that takes the index of an element
 * and performs the action on the element.
 * 
 * @sample DNAC. UnsignedArray, onEachIndexed
 *
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.onEachIndexed(action: (index: Int, UByte) -> Unit): UByteArray {
    return apply {
        forEachIndexed { action(it, it) }
    }
}

/**
 * Performs the given [action] on each element and returns the array itself
 * afterwards.
 * 
 * @param [action] function that takes the index of an element
 * and performs the action on the element.
 * 
 * @sample DNAC. UnsignedArray, onEachIndexed
 *
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.onEachIndexed(action: (index: Int, UShort) -> Unit): UShortArray {
    return apply {
        forEachIndexed { action(it, it) }
    }
}
element and the element itself * and performs the action on the element.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.onEachIndexed(action: (index: Int, ULong) -> Unit): ULongArray {
    return apply {
        forEachIndexed(action) *
    }
}

/**
 * Performs the given [action] on each element, providing sequential index
 * with the element, * and returns the array itself afterwards.
 * @param [action] function that takes the index of an element and the element itself * and performs the action on the element.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.onEachIndexed(action: (index: Int, UByte) -> Unit): UByteArray {
    return apply {
        forEachIndexed(action) *
    }
}

/**
 * Performs the given [action] on each element, providing sequential index
 * with the element, * and returns the array itself afterwards.
 * @param [action] function that takes the index of an element and the element itself * and performs the action on the element.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.onEachIndexed(action: (index: Int, UShort) -> Unit): UShortArray {
    return apply {
        forEachIndexed(action) *
    }
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. *
 * Throws an exception if this array is empty. If the array can be empty in an expected way, *
 * please use [reduceOrNull] instead. It returns `null` when its receiver is empty. *
 * @param [operation] function that takes current accumulator value and an element, *
 * and calculates the next accumulator value. *
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.reduce(operation: (acc: UInt, UInt) -> UInt): UInt {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. *
 * Throws an exception if this array is empty. If the array can be empty in an expected way, *
 * please use [reduceOrNull] instead. It returns `null` when its receiver is empty. *
 * @param [operation] function that takes current accumulator value and an element, *
 * and calculates the next accumulator value. *
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. *
 * Throws an exception if this array is empty. If the array can be empty in an expected way, *
 * please use [reduceOrNull] instead. It returns `null` when its receiver is empty. *
 * @param [operation] function that takes current accumulator value and an element, *
 * and calculates the next accumulator value. *
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element. *
 * Throws an exception if this array is empty. If the array can be empty in an expected way, *
 * please use [reduceOrNull] instead. It returns `null` when its receiver is empty. *
 * @param [operation] function that takes current accumulator value and an element, *
 * and calculates the next accumulator value. *
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.reduce(operation: (acc: UShort, UShort) -> UShort): UShort {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```
1..lastIndex) {
    accumulator = operation(accumulator, this[index])
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and calculates the next accumulator value.
 * @since K
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.reduceIndexedOrNull(operation: (index: Int, acc: UByte, UByte) -> UByte): Nullable UByte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and calculates the next accumulator value.
 * @since K
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and calculates the next accumulator value.
 * @since K
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and calculates the next accumulator value.
 * @since K
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, current accumulator value
 * and calculates the next accumulator value.
 * @since K
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline func
UIntArray.reduceIndexedOrNull(operation: (index: Int, acc: UInt, UInt) -> UInt): Nullable UInt {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
return null
var accumulator = this[0]
for (index in 1..lastIndex) {
  accumulator = operation(index, accumulator, this[index])
}
return accumulator

Returns ‘null’ if the array is empty.

@sample samples.collections.Aggregates.reduceOrNull

/*@SinceKotlin("1.4.4")*/
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceIndexedOrNonull(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong? {
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(index, accumulator, this[index])
  }
  return accumulator
}

Returns ‘null’ if the array is empty.

@sample samples.collections.Aggregates.reduceOrNull

/*@SinceKotlin("1.4.4")*/
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceIndexedOrNonull(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte? {
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(index, accumulator, this[index])
  }
  return accumulator
}

Returns ‘null’ if the array is empty.

@sample samples.collections.Aggregates.reduceOrNull

/*@SinceKotlin("1.4.4")*/
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(index, accumulator, this[index])
  }
  return accumulator
}

Returns ‘null’ if the array is empty.

@sample samples.collections.Aggregates.reduceOrNull

/*@SinceKotlin("1.4.4")*/
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceOrNull(operation: (acc: UInt, UInt) -> UInt): UInt? {
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(accumulator, this[index])
  }
  return accumulator
}

Returns ‘null’ if the array is empty.

@sample samples.collections.Aggregates.reduceOrNull

/*@SinceKotlin("1.4.4")*/
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceOrNull(operation: (acc: ULong, ULong) -> ULong): ULong? {
  var accumulator = this[0]
  for (index in 1..lastIndex) {
    accumulator = operation(accumulator, this[index])
  }
  return accumulator
}

Returns ‘null’ if the array is empty.

@sample samples.collections.Aggregates.reduceOrNull

/*@SinceKotlin("1.4.4")*/
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceOrNull(operation: (acc: UByte, UByte) ->

UByte): UByte? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex)
        accumulator = operation(accumulator, this[index])
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex)
        accumulator = operation(accumulator, this[index])
    return accumulator
}

/**
 * Accumulates
 * value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @throws an exception if this array is empty. If the array can be empty in an expected way, use [reduceRightOrNull] instead.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceRight(operation: (UInt, acc: UInt) -> UInt): UInt {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates
 * value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @throws an exception if this array is empty. If the array can be empty in an expected way, use [reduceRightOrNull] instead.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceRight(operation: (ULong, acc: ULong) -> ULong): ULong {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates
 * value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @throws an exception if this array is empty. If the array can be empty in an expected way, use [reduceRightOrNull] instead.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates
 * value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * @throws an exception if this array is empty. If the array can be empty in an expected way, use [reduceRightOrNull] instead.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort): UShort {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator =
get(index--) while (index >= 0) {
    accumulator = operation(get(index--), accumulator)
} return accumulator

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way, 
 * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@FunctionalInterface
@kotlin.internal.InlineOnly
public fun <T> List<T>.reduceRightIndexed(operation: (index: Int, T, acc: T) -> T): T {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@FunctionalInterface
@kotlin.internal.InlineOnly
public fun <T> List<T>.reduceRightIndexed(operation: (index: Int, T, acc: T) -> T): T {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reduceRightIndexedOrNull(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.reduceRightIndexedOrNull(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UByteArray.reduceRightIndexedOrNull(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UIntArray.reduceRightOrNull(operation: (UInt, acc: UInt) -> UInt): UInt? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
ULongArray.reduceRightOrNull(operation: (ULong, acc: ULong) -> ULong): ULong? {
    var index = lastIndex
    if (index < 0) return null
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull
 */
left\n* to each element and current accumulator value.\n* \n* Returns `null` if the array is empty.\n* \n* @param 
[operation] function that takes an element and current accumulator value \n* and calculates the next accumulator \n* \n* @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
\*@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly

public inline fun UByteArray.reduceRightOrNull(operation: (UByte, acc: UByte) -> UByte): UByte? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n
    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n
    return accumulator
}\n
/**
 * Accumulates value starting with the last element and applying [operation] from right to \n* to each element and current accumulator value.\n* \n* Returns `null` if the array is empty.\n* \n* @param 
* \n* function that takes an element and current accumulator value,\n* and calculates the next accumulator \n* \n* @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
\*@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly

    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)

    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }

    return accumulator
}

/**
 * Returns a list containing successive accumulation values generated by applying \n[operation] from left to right\n* to each element and current accumulator value that starts with \n* \n* Note that `acc` value passed to \n[operation] function should not be mutated;\n* otherwise it would affect the \nprevious value in resulting list.\n* \n* @param \n* \n* function that takes current accumulator value and an \n* \n* @sample samples.collections.Collections.Aggregates.runningFold\n
\*@SinceKotlin("1.4")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly

public inline fun <R> UIntArray.runningFold(initial: R, operation: (acc: R, UInt) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }

    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying \n[operation] from left to right\n* to each element and current accumulator value that starts with \n* \n* Note that `acc` value passed to \n[operation] function should not be mutated;\n* otherwise it would affect the \nprevious value in resulting list.\n* \n* @param \n* \n* function that takes current accumulator value and an \n* \n* @sample samples.collections.Collections.Aggregates.runningFold\n
\*@SinceKotlin("1.4")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly

public inline fun <R> ULongArray.runningFold(initial: R, operation: (acc: R, ULong) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }

    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying \n[operation] from left to right\n* to each element and current accumulator value that starts with \n* \n* Note that `acc` value passed to \n[operation] function should not be mutated;\n* otherwise it would affect the \nprevious value in resulting list.\n* \n* @param \n* \n* function that takes current accumulator value and an \n* \n* @sample samples.collections.Collections.Aggregates.runningFold\n
\*@SinceKotlin("1.4")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly

public inline fun <R> UByteArray.runningFold(initial: R, operation: (acc: R, UByte) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }

    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying \n[operation] from left to right\n* to each element and current accumulator value that starts with \n* \n* Note that `acc` value passed to \n[operation] function should not be mutated;\n* otherwise it would affect the \nprevious value in resulting list.\n* \n* @param \n* \n* function that takes current accumulator value and an \n* \n* @sample samples.collections.Collections.Aggregates.runningFold\n
\*@SinceKotlin("1.4")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly

    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }

    return result
}
previous value in resulting list.

*param* operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index]) result.add(accumulator)
    }
    return result
}

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index]) result.add(accumulator)
    }
    return result
}

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index]) result.add(accumulator)
    }
    return result
}

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index]) result.add(accumulator)
    }
    return result
}
```

Note that `acc` value passed to `[operation]` function should not be mutated; otherwise it would affect the previous value in resulting list.

*param* operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}
```

The previous value in resulting list.

*param* operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}
```

The previous value in resulting list.

*param* operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}
```

The previous value in resulting list.

*param* operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}
```

The previous value in resulting list.

*param* operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}
```

The previous value in resulting list.

*param* operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}
```

The previous value in resulting list.

*param* operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}
```

The previous value in resulting list.

*param* operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) } var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element) result.add(accumulator)
    }
    return result
}
```
values generated by applying [operation] from left to right * to each element and current accumulator value that starts with the first element of this array. * Note that `acc` value passed to [operation] function should not be mutated; * otherwise it would affect the previous value in resulting list. * * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

@SinceKotlin("1.4")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.runningReduce(operation: (acc: UInt, UInt) -> UInt): List<UInt>
{
    if (isEmpty()) return emptyList()
    val result = ArrayList<UInt>(size).apply { add(acc) } for (index in 1 until size) { accumulator = operation(accumulator, this[index]); result.add(accumulator) }
    return result
}

@SinceKotlin("1.4")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.runningReduce(operation: (acc: ULong, ULong) -> ULong): List<ULong>
{
    if (isEmpty()) return emptyList()
    val result = ArrayList<ULong>(size).apply { add(acc) } for (index in 1 until size) { accumulator = operation(accumulator, this[index]); result.add(accumulator) }
    return result
}

@SinceKotlin("1.4")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.runningReduce(operation: (acc: UByte, UByte) -> UByte): List<UByte>
{
    if (isEmpty()) return emptyList()
    val result = ArrayList<UByte>(size).apply { add(acc) } for (index in 1 until size) { accumulator = operation(accumulator, this[index]); result.add(accumulator) }
    return result
}

@SinceKotlin("1.4")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
{
    if (isEmpty()) return emptyList()
    val result = ArrayList<UShort>(size).apply { add(acc) } for (index in 1 until size) { accumulator = operation(accumulator, this[index]); result.add(accumulator) }
    return result
}

@SinceKotlin("1.4")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.runningReduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): List<UInt>
{
    if (isEmpty()) return emptyList()
    val result = ArrayList<UInt>(size).apply { add(acc) } for (index in 1 until size) { accumulator = operation(accumulator, this[index], index); result.add(accumulator) }
    return result
}
for (index in 1 until size) {
    accumulator = operation(index, accumulator, this[index])
    result.add(accumulator)
} else return result

result.add(accumulator)\n
@sample

samples.Collections.Collections.Aggregates.runningReduce

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n\npublic inline fun ULongArray.runningReduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): List<ULong> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<ULong>(size).apply {
        add(accumulator)\n        for (index in 1 until size) {
            accumulator = operation(index, accumulator, this[index])
            result.add(accumulator)
        }
    }\n    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right \n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * Note that \`acc\` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n\npublic inline fun UByteArray.runningReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UByte>(size).apply {
        add(accumulator)\n        for (index in 1 until size) {
            accumulator = operation(index, accumulator, this[index])
            result.add(accumulator)
        }
    }\n    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right \n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * Note that \`acc\` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n\npublic inline fun UShortArray.runningReduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): List<UShort> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UShort>(size).apply {
        add(accumulator)\n        for (index in 1 until size) {
            accumulator = operation(index, accumulator, this[index])
            result.add(accumulator)
        }
    }\n    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right \n * to each element and current accumulator value that starts with \[initial\] value.\n * Note that \`acc\` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n\npublic inline fun UIntArray.scan(initial: UInt, operation: (acc: UInt, UInt) -> UInt): List<UInt> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right \n * to each element and current accumulator value that starts with \[initial\] value.\n * Note that \`acc\` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n\npublic inline fun <R> UIntArray.scan(initial: R, operation: (acc: R, UInt) -> R): List<R> {
    return runningFold(initial, operation)
}
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.scan(initial: R, operation: (acc: R, ULong) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * 
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * 
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.scan(initial: R, operation: (acc: R, UByte) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * 
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * 
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * 
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}
/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 * @param selector function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample
 * samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return runningFoldIndexed(initial, operation)
}

@Deprecated("Use runningReduce instead.", ReplaceWith("runningReduce(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.scanReduce(operation: (acc: UInt, UInt) -> UInt): List<UInt> {
    return runningReduce(operation)
}

@Deprecated("Use runningReduce instead.", ReplaceWith("runningReduce(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.scanReduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): List<ULong> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.", ReplaceWith("runningReduceIndexed(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.scanReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.", ReplaceWith("runningReduceIndexed(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return runningReduceIndexed(operation)
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.sumBy(selector: (UInt) -> UInt): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.sumBy(selector: (ULong) -> UInt): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.sumBy(selector: (UByte) -> UByte): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumBy(selector: (UShort) -> UShort): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
UByteArray.sumBy(selector: UByte -> UInt): UInt {
    var sum: UInt = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by 
 * [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.sumBy(selector: UByte -> UInt): UInt {
    var sum: UInt = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by 
 * [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.sumBy(selector: UShort -> UInt): UInt {
    var sum: UInt = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by 
 * [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.sumByDouble(selector: UInt -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by 
 * [selector] function applied to each element in the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.sumByDouble(selector: ULong -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by 
 * [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.sumOf(selector: UInt -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by 
 * [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.sumOf(selector: ULong -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by 
 * [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.sumOf(selector: UByte -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by 
 * [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.sumOf(selector: UShort -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
@kotlin.jvm.JvmName("sumOfInt")

@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.sumOf(selector: (UInt) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfInt")

@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.sumOf(selector: (ULong) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfInt")

@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.sumOf(selector: (UByte) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfInt")

@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumOf(selector: (UShort) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfLong")

@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.sumOf(selector: (UInt) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfLong")

@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.sumOf(selector: (ULong) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfLong")

@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.sumOf(selector: (UByte) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfLong")

@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.sumOf(selector: (UShort) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumOf(selector: (UInt) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in
 * the array.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector: (ULong) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in
 * the array.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector: (UByte) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in
 * the array.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfUInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns a list of pairs built from the elements of `this` array and
 * the [other] array with the same index.\n * The returned list has length of the shortest collection.\n */
@sample
Open Source Used In DNAC Platform Antelope 9478
/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@public infix fun <R> UIntArray.zip(other: Array<out R>): List<Pair<UInt, R>> \n{ \n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n*/

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@public infix fun <R> ULongArray.zip(other: Array<out R>): List<Pair<ULong, R>> \n{ \n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n*/

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@public infix fun <R> UByteArray.zip(other: Array<out R>): List<Pair<UByte, R>> \n{ \n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n*/

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@public infix fun <R> UShortArray.zip(other: Array<out R>): List<Pair<UShort, R>> \n{ \n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n*/

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@public infix fun <R> UIntArray.zip(other: Iterable<R>): List<Pair<UInt, R>> \n{ \n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n*/

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@public infix fun <R> ULongArray.zip(other: Iterable<R>): List<Pair<ULong, R>> \n{ \n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n*/

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@public infix fun <R> UByteArray.zip(other: Iterable<R>): List<Pair<UByte, R>> \n{ \n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n*/

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@public infix fun <R> UShortArray.zip(other: Iterable<R>): List<Pair<UShort, R>> \n{ \n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n*/

fun <R> UIntArray.zip(other: Array<out R>): List<Pair<UInt, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

fun <R> ULongArray.zip(other: Array<out R>): List<Pair<ULong, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

fun <R> UByteArray.zip(other: Array<out R>): List<Pair<UByte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

fun <R> UShortArray.zip(other: Array<out R>): List<Pair<UShort, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

fun <R> UIntArray.zip(other: Iterable<R>): List<Pair<UInt, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

fun <R> ULongArray.zip(other: Iterable<R>): List<Pair<ULong, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

fun <R> UByteArray.zip(other: Iterable<R>): List<Pair<UByte, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

fun <R> UShortArray.zip(other: Iterable<R>): List<Pair<UShort, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}
/*
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> ULongArray.zip(other: Iterable<R>): List<Pair<ULong, R>> { 
    return zip(other) { t1, t2 -> t1 to t2 } } //fun
 */

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UByteArray.zip(other: Iterable<R>): List<Pair<UByte, R>> { 
    return zip(other) { t1, t2 -> t1 to t2 } } //fun

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UShortArray.zip(other: Iterable<R>): List<Pair<UShort, R>> { 
    return zip(other) { t1, t2 -> t1 to t2 } } //fun

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UIntArray.zip(other: UIntArray): List<Pair<UInt, UInt>> { 
    return zip(other) { t1, t2 -> t1 to t2 } } //fun

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> ULongArray.zip(other: ULongArray): List<Pair<ULong, ULong>> { 
    return zip(other) { t1, t2 -> t1 to t2 } } //fun

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UByteArray.zip(other: UByteArray): List<Pair<UByte, UByte>> { 
    return zip(other) { t1, t2 -> t1 to t2 } } //fun

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
    return zip(other) { t1, t2 -> t1 to t2 } } //fun

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index.
 * Using the provided [transform] function applied to each pair of elements. The returned list has length of the shortest collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UIntArray.zip(other: UIntArray, transform: (a: UInt, b: R) -> V): List<V> { 
    val arraySize = size documentaries
val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
} //fun

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> ULongArray.zip(other: ULongArray, transform: (a: ULong, b: R) -> V): List<V> { 
    val arraySize = size documentaries
val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
} //fun

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UByteArray.zip(other: UByteArray, transform: (a: UByte, b: R) -> V): List<V> { 
    val arraySize = size documentaries
val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
} //fun

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UShortArray.zip(other: UShortArray, transform: (a: UShort, b: R) -> V): List<V> { 
    val arraySize = size documentaries
val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
} //fun

Open Source Used In DNAC Platform Antelope 9480
shortest collection.

*\@SinceKotlin("1.3\")*\@ExperimentalUnsignedTypes*\@public infix fun UByteArray.zip\(other: UByteArray\): List\&Pair\&UByte, UByte\)\{\n  return zip\(other\) { t1, t2 \rightarrow t1 to t2 }\n\}\n\*/

*\@SinceKotlin("1.3\")*\@ExperimentalUnsignedTypes*\@public infix fun UShortArray.zip\(other: UShortArray\): List\&Pair\&UShort, UShort\)\{\n  return zip\(other\) { t1, t2 \rightarrow t1 to t2 }\n\}\n\*/

*\@SinceKotlin("1.3\")*\@ExperimentalUnsignedTypes*\@public infix fun UIntArray.zip\(other: UIntArray, transform: (a: UInt, b: UInt) \rightarrow V\): List\&V\)\{\n  val size = minOf(size, other.size)\n  val list = ArrayList\&V\(size\)\n  for (i in 0 until size) {\n    list.add\(transform\(this\[i\], other\[i\]\))\n  }\n  return list\}\n\*/

*\@SinceKotlin("1.3\")*\@ExperimentalUnsignedTypes*\@public infix fun UByteArray.zip\(other: UByteArray, transform: (a: UByte, b: UByte) \rightarrow V\): List\&V\)\{\n  val size = minOf(size, other.size)\n  val list = ArrayList\&V\(size\)\n  for (i in 0 until size) {\n    list.add\(transform\(this\[i\], other\[i\]\))\n  }\n  return list\}\n\*/

*\@SinceKotlin("1.3\")*\@ExperimentalUnsignedTypes*\@public infix fun ULongArray.zip\(other: ULongArray, transform: (a: ULong, b: ULong) \rightarrow V\): List\&V\)\{\n  val size = minOf(size, other.size)\n  val list = ArrayList\&V\(size\)\n  for (i in 0 until size) {\n    list.add\(transform\(this\[i\], other\[i\]\))\n  }\n  return list\}\n\*/

*\@SinceKotlin("1.3\")*\@ExperimentalUnsignedTypes*\@public fun Array<out UInt>.sum\(): UInt\{\n    var sum: UInt = 0u\n    for (element in this) {\n      sum += element\n    }\n    \n    return sum\n\}
**UIntArray.sum()**: UInt { `n return storage.sum().toUInt();` }\n\n**Returns the sum of all elements in the array.**

`n *\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun\n
```kotlin
UIntArray.sum(): ULong {\n    return storage.sum().toULong();\n}

/**\n * Returns the sum of all elements in the array.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

```kotlin
UByteArray.sum(): UInt {\n    return sumBy { it.toInt(); }\n}

/**\n * Returns the sum of all elements in the array.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

```kotlin
UByteArray.sum(): UInt {\n    return sumBy { it.toUInt(); }\n}

/**\n * Returns the sum of all elements in the array.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

```kotlin
UShortArray.sum(): UInt {\n    return sumBy { it.toUInt(); }\n}

/**\n * Returns the sum of all elements in the array.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

```kotlin
ULongArray.sum(): ULong {\n    return storage.sum().toULong();\n}

/**\n * Returns the sum of all elements in the array.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

```kotlin
"/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n
@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("UCollectionsKt")\n
package kotlin.collections\n
//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n
import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed\n
/**\n * Returns an array of UByte containing all of the elements of this collection.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun\n
```kotlin
Collection<UByte>.toUByteArray(): UByteArray {\n    val result = UByteArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}

/**\n * Returns an array of UInt containing all of the elements of this collection.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun\n
```kotlin
Collection<UInt>.toUIntArray(): UIntArray {\n    val result = UIntArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}

/**\n * Returns an array of ULong containing all of the elements of this collection.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun\n
```kotlin
Collection<ULong>.toULongArray(): ULongArray {\n    val result = ULongArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}

/**\n * Returns an array of UShort containing all of the elements of this collection.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun\n
```kotlin
Collection<UShort>.toUShortArray(): UShortArray {\n    val result = UShortArray(size)\n    var index = 0\n    for (element in this)\n        result[index++] = element\n    return result\n}

/**\n * Returns the sum of all elements in the collection.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun

```kotlin
Iterable<UInt>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {
        sum += element
    }
    return sum
}

/**\n * Returns the sum of all elements in the collection.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun

```kotlin
Iterable<ULong>.sum(): ULong {\n    var sum: ULong = 0uL\n    for (element in this) {
        sum += element
    }
    return sum
}

/**\n * Returns the sum of all elements in the collection.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun

```kotlin
Iterable<UByte>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {
        sum += element
    }
    return sum
}

/**\n * Returns the sum of all elements in the collection.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun

```kotlin
Iterable<UShort>.sum(): UInt {\n    var sum: UInt = 0u\n    for (element in this) {
        sum += element
    }
    return sum
}

*/\n
/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n
@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName("UComparisonsKt")\n
package kotlin.comparisons\n
//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n
import kotlin.random.*

/**\n * Returns the greater of two values.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun\n
```kotlin
maxOf(a: UInt, b: UInt): UInt {\n    return if (a >= b) a else b\n}

/**\n * Returns the greater of two values.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun

```kotlin
maxOf(a: ULong, b: ULong): ULong {\n    return if (a >= b) a else b\n}

/**\n * Returns the greater of two values.\n */\n\n@SinceKotlin("1.3.3")\n@ExperimentalUnsignedTypes\n@public fun

```kotlin
maxOf(a: UByte, b: UByte): UByte {\n    return if (a >= b) a else b\n}

*/\n
Open Source Used In DNAC Platform Antelope 9482
public fun maxOf(a: UShort, b: UShort): UShort {
    return if (a >= b) a else b
}

/**
 * Returns the greater of three values.
 */
public fun maxOf(a: UInt, b: UInt, c: UInt): UInt {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of three
 * values.
 */
public fun maxOf(a: ULong, b: ULong, c: ULong): ULong {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of three
 * values.
 */
public fun maxOf(a: UByte, b: UByte, c: UByte): UByte {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of three
 * values.
 */
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of the given values.
 */
public fun maxOf(a: UInt, vararg other: UInt): UInt {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 */
public fun maxOf(a: ULong, vararg other: ULong): ULong {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 */
public fun maxOf(a: UByte, vararg other: UByte): UByte {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the greater of the given values.
 */
public fun maxOf(a: UShort, vararg other: UShort): UShort {
    var max = a
    for (e in other) max = maxOf(max, e)
    return max
}

/**
 * Returns the smaller of two values.
 */
public fun minOf(a: UInt, b: UInt): UInt {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 */
public fun minOf(a: ULong, b: ULong): ULong {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 */
public fun minOf(a: UByte, b: UByte): UByte {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of two values.
 */
public fun minOf(a: UShort, b: UShort): UShort {
    return if (a <= b) a else b
}

/**
 * Returns the smaller of three values.
 */
public fun minOf(a: UInt, b: UInt, c: UInt): UInt {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the smaller of three values.
 */
public fun minOf(a: UByte, b: UByte, c: UByte): UByte {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the smaller of three values.
 */
    return minOf(a, minOf(b, c))
}

/**
 * Returns the smaller of the given values.
 */
public fun minOf(a: UInt, vararg other: UInt): UInt {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of the given values.
 */
public fun minOf(a: UByte, vararg other: UByte): UByte {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of the given values.
 */
public fun minOf(a: UShort, vararg other: UShort): UShort {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}
governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

An@SinceKotlin("1.3")@ExperimentalUnsignedTypes

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("URangesKt")

package

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*

/**
 * Returns a random element from this range.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun UIntRange.random(): UInt {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public inline fun ULongRange.random(): ULong {
    return random(Random)
}

/**
 * Returns a random element from this range using the specified source of randomness.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun UIntRange.random(random: Random): UInt {
    try {
        return random.nextUInt(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the specified source of randomness.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
public fun ULongRange.random(random: Random): ULong {
    try {
        return random.nextULong(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range, or `null` if this range is empty.
 */
@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public inline fun UIntRange.randomOrNull(): UInt? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this range, or `null` if this range is empty.
 */
@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public inline fun ULongRange.randomOrNull(): ULong? {
    return randomOrNull(Random)
}

/**
 * Returns `true` if this range contains the specified element.
 * Always returns `false` if the element is `null`.
 */
@SinceKotlin("1.3")
public operator fun UIntRange.contains(element: UByte): Boolean {
    return contains(element.toUInt())
}

/**
 * Checks if the specified value belongs to this range.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntRange.contains(value: UByte): Boolean {
    return contains(value.toUInt())
}

/**
 * Checks if the specified value belongs to this range.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: UByte): Boolean {
    return contains(value.toULong())
}

/**
 * Checks if the specified value belongs to this range.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: UInt): Boolean {
    return contains(value.toULong())
}

/**
 * Checks if the specified value belongs to this range.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: UByte): Boolean {
    return contains(value.toULong())
}

/**
 * Checks if the specified value belongs to this range.
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: ULong): Boolean {
    return contains(value.toULong())
}
Open Source Used In DNAC Platform Antelope 9485

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntRange.contains(value: UShort): Boolean {
    return contains(value.toUInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongRange.contains(value: UShort): Boolean {
    return contains(value.toULong())
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * 
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun UByte.downTo(to: UByte): UIntProgression {
    return UIntProgression.fromClosedRange(this.toUInt(), to.toUInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * 
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun UInt.downTo(to: UInt): UIntProgression {
    return UIntProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * 
 * If the [to] value is greater than ‘this’ value the returned progression is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun ULong.downTo(to: ULong): ULongProgression {
    return ULongProgression.fromClosedRange(this, to, -1)
}

/**
 * Returns a progression that goes over the same range in the opposite direction with the same step.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntProgression.reversed(): UIntProgression {
    return UIntProgression.fromClosedRange(last, first, -step)
}

/**
 * Returns a progression that goes over the same range in the opposite direction with the same step.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongProgression.reversed(): ULongProgression {
    return ULongProgression.fromClosedRange(last, first, -step)
}

/**
 * Returns a progression that goes over the same range with the given step.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun UIntProgression.step(step: Int): UIntProgression {
    checkStepIsPositive(step > 0, step)
    return UIntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)
}

/**
 * Returns a progression that goes over the same range with the given step.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun ULongProgression.step(step: Long): ULongProgression {
    checkStepIsPositive(step > 0, step)
    return ULongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * 
 * If the [to] value is less than or equal to ‘this’ value, then the returned range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun UByte.until(to: UByte): UIntRange {
    if (to <= UByte.MIN_VALUE) return UIntRange.EMPTY
    return this.toUInt() .. (to - 1u).toUInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * 
 * If the [to] value is less than or equal to ‘this’ value, then the returned range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun UInt.until(to: UInt): UIntRange {
    if (to <= UInt.MIN_VALUE) return UIntRange.EMPTY
    return this .. (to - 1u).toUInt()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * 
 * If the [to] value is less than or equal to ‘this’ value, then the returned range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun ULong.until(to: ULong): ULongRange {
    if (to <= ULong.MIN_VALUE) return ULongRange.EMPTY
    return this .. (to - 1u).toULong()
}

/**
 * Returns a range from this value up to but excluding the specified [to] value.
 * 
 * If the [to] value is less than or equal to ‘this’ value, then the returned range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun UShort.until(to: UShort): UIntRange {
    if (to <= UShort.MIN_VALUE) return UIntRange.EMPTY
    return this .. (to - 1u).toUInt()
}
UShort): UIntRange {\n  if (to <= UShort.MIN_VALUE) return UIntRange.EMPTY
  return this.toUInt()...(to - 1U).toUInt()\n}\n\n@ExperimentalUnsignedTypes
public fun UInt.coerceAtLeast(minimumValue: UInt): UInt {\n  return if (this < minimumValue) minimumValue else this\n}\n\n/**
 * Ensures that this value is not less than the specified [minimumValue].\n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun ULong.coerceAtLeast(minimumValue: ULong): ULong {\n  return if (this < minimumValue) minimumValue else this\n}\n\n/**
 * Ensures that this value lies in the specified range [minimumValue].[maximumValue].\n * @return this value if it's less than or equal to the specified range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * @sample samples.comparisons.ComparableOps.coerceInRangeUnsigned
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UInt.coerceInRange(minimumValue: UInt, maximumValue: UInt): UInt {\n  if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")\n  return if (this < minimumValue) minimumValue
    else if (this > maximumValue) maximumValue
    else this
\n}"
Ensures that this value lies in the specified range $[minimumValue..maximumValue]$.

@return this value if it's in the range, or $minimumValue$ if this value is less than $minimumValue$, or $maximumValue$ if this value is greater than $maximumValue$. samples.comparisons.ComparableOps.coerceInUnsigned

public fun ULong.coerceIn(minimumValue: ULong, maximumValue: ULong): ULong {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.

    return this
}

/**
 * Ensures that this value lies in the specified range $[minimumValue..maximumValue]$. 

 * @return this value if it's in the range, or $minimumValue$ if this value is less than $minimumValue$, or $maximumValue$ if this value is greater than $maximumValue$. samples.comparisons.ComparableOps.coerceInUnsigned

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.

    return this
}

/**
 * Ensures that this value lies in the specified range $[minimumValue..maximumValue]$. 

 * @return this value if it's in the range, or $minimumValue$ if this value is less than $minimumValue$, or $maximumValue$ if this value is greater than $maximumValue$. samples.comparisons.ComparableOps.coerceInUnsigned

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.

    return this
}

/**
 * Ensures that this value lies in the specified $range$.

 * @return this value if it's in the range, or $range.start$ if this value is less than $range.start$, or $range.endInclusive$ if this value is greater than $range.endInclusive$. samples.comparisons.ComparableOps.coerceInUnsigned

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UInt.coerceIn(range: ClosedRange<UInt>): UInt {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<UInt>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.

    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else ->
            this
    }
}

/**
 * Ensures that this value lies in the specified $range$. 

 * @return this value if it's in the range, or $range.start$ if this value is less than $range.start$, or $range.endInclusive$ if this value is greater than $range.endInclusive$. samples.comparisons.ComparableOps.coerceInUnsigned

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULong.coerceIn(range: ClosedRange<ULong>): ULong {
    if (range is ClosedFloatingPointRange) {
        return this.coerceIn<ULong>(range)
    }
    if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.

    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else ->
            this
    }
}

/**
 * Returns the sum of all elements in the sequence.

 * The operation is _terminal_.

 */
@kotlin.jvm.JvmName("sumOfUInt")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<UInt>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }

    return sum
}
return sum
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfULong")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<ULong>.sum(): ULong {
    var sum: ULong = 0uL
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfUByte")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<UByte>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 *
 * The operation is _terminal_.
 */
@kotlin.jvm.JvmName("sumOfUShort")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Sequence<UShort>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin

public expect open class Error : Throwable {
    constructor()
    constructor(message: String?)
    constructor(message: String?, cause: Throwable?)
    constructor(cause: Throwable?)
}

public expect open class Exception : Throwable {
    constructor()
    constructor(message: String?)
    constructor(message: String?, cause: Throwable?)
    constructor(cause: Throwable?)
}

public expect open class RuntimeException : Exception {
    constructor()
    constructor(message: String?)
    constructor(message: String?, cause: Throwable?)
    constructor(cause: Throwable?)
}

public expect open class IllegalArgumentException : RuntimeException {
    constructor()
    constructor(message: String?)
    constructor(message: String?, cause: Throwable?)
    constructor(cause: Throwable?)
}

public expect open class IllegalStateException : RuntimeException {
    constructor()
    constructor(message: String?)
    constructor(message: String?, cause: Throwable?)
    constructor(cause: Throwable?)
}

public expect open class IndexOutOfBoundsException : RuntimeException {
    constructor()
}

public expect open class ConcurrentModificationException : RuntimeException {
    constructor()
    constructor(message: String?)
    @Deprecated("The constructor is not supported on all platforms and will be removed from kotlin-stdlib-common soon.", level = DeprecationLevel.ERROR)
    constructor(message: String?, cause: Throwable?)
    @Deprecated("The constructor is not supported on all platforms and will be removed from kotlin-stdlib-common soon.", level = DeprecationLevel.ERROR)
    constructor(cause: Throwable?)
}

public expect open class UnsupportedOperationException : RuntimeException {
    constructor()
    constructor(message: String?)
    constructor(message: String?, cause: Throwable?)
    constructor(cause: Throwable?)
}

public expect open class NumberFormatException : IllegalArgumentException {
    constructor()
    constructor(message: String?)
}

public expect open class NullPointerException : RuntimeException {
    constructor()
    constructor(message: String?)
}

public expect open class ClassCastException : RuntimeException {
    constructor()
    constructor(message: String?)
}

public expect open class AssertionError : Error {
    constructor()
    constructor(message: Any?)
    constructor(message: String?, cause: Throwable?)
}

/**
 * Thrown after invocation of a function or property that was expected to return `Nothing`,
 * but returned something instead.
 */
@SinceKotlin("1.4")
@PublishedApi
internal class KotlinNothingValueException : RuntimeException {
    constructor() : super()
    constructor(message: String?) : super(message + " is not expected")
    constructor(message: String?, cause: Throwable?) : super(message, cause)
}
super(message)

constructor(message: String?, cause: Throwable?) : super(message, cause)

constructor(cause: Throwable?) : super(cause)

/**
 * Returns the detailed description of this throwable with its stack trace.
 * The detailed description includes:
 * - the short description (see [Throwable.toString])
 * of this throwable;
 * - the complete stack trace;
 * - detailed descriptions of the exceptions that were
 * [suppressed][suppressedExceptions] in order to deliver this exception;
 * - the detailed description of each
 * throwable in the [Throwable.causes] chain.
 */

SinceKotlin("1.4")

public expect fun Throwable.stackTraceToString(): String

public expect fun Throwable.printStackTrace(): Unit

public expect fun Throwable.addSuppressed(exception: Throwable)

public expect val Throwable.suppressedExceptions: List<Throwable>

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 */

package kotlin.js

import kotlin.annotation.AnnotationTarget.*

/**
 * Gives a declaration (a function, a
 * property or a class) specific name in JavaScript.
 */
@Target(CLASS, FUNCTION, PROPERTY, CONSTRUCTOR, PROPERTY_GETTER, PROPERTY_SETTER)
@OptionalExpectation
public expect annotation class JsName(val name: String)

/**
 * Marks experimental JS export annotations.
 *
 * Note that
 * behavior of these annotations will likely be changed in the future.
 * Usages of such annotations will be reported as warnings unless an explicit opt-in with
 * the [OptIn] annotation, e.g.
 * `@OptIn(ExperimentalJsExport::class)`
 * or with the `-Xopt-in=kotlin.js.ExperimentalJsExport` compiler option is given.
 */
@Suppress("DEPRECATION")
@Experimental(level = Experimental.Level.WARNING)
@RequiresOptIn(level = RequiresOptIn.Level.WARNING)
@MustBeDocumented
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.4")
@OptionalExpectation
public expect annotation class ExperimentalJsExport

/**
 * Exports top-level declaration on JS
 * platform.
 *
 * Compiled module exposes declarations that are marked with this annotation without name
 * mangling.
 * This annotation can be applied to either files or top-level declarations.
 * It is currently
 * prohibited to export the following kinds of declarations:
 * **inline functions with reified type parameters**
 * **suspend functions**
 * **secondary constructors without `@JsName`**
 * **extension properties**
 * **enum classes**
 * **annotation classes**
 * **Signatures of exported declarations must only contain \`exportable\` types:**
 * dynamic, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`,
 * `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`,
 * `Array<exportable-type>`
 * `Function types with exportable parameters and return type` and `@JsExport` classes and interfaces
 * **Nullable counterparts of types above**
 * **Unit return type. Must not be
 * nullable**
 * **This annotation is experimental, meaning that restrictions mentioned above are subject to change.
 */
@ExperimentalJsExport
@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)
@SinceKotlin("1.4")
@OptionalExpectation
public expect annotation class JsExport("", "/\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 */

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@SinceKotlin("1.4")
@OptionalExpectation
public expect annotation class SinceKotlin("1.4")

Open Source Used In DNAC Platform Antelope 9489
T: Iterable<T> {\n    /** Returns an iterator over the elements of this sequence that supports removing elements during iteration.\n     * @return the specified element is contained in this list,\n     * returns the index of the first occurrence of the specified element in the list, or -1 if the specified\n     * element is not contained in the list.\n     * @param E the type of elements contained in the collection.\n     * The collection is covariant in its element type.\n     */
    override fun iterator(): Iterator<T> {\n        return super.iterator()\n    }\n    // Query Operations\n    /** Returns the size of the collection.\n     * @return the number of elements in the collection.\n     */
    public fun size(): Int {\n        return list.size\n    }\n    // Bulk Operations\n    /** Checks if all elements in the specified collection are contained in this collection.\n     * @return \n     * Checks if all elements in the specified collection are contained in this collection.\n     * @return \n     */
    fun containsAll(elements: Collection<E>): Boolean {\n        return list.containsAll(elements)\n    }\n    /** Checks if all elements in the specified collection are contained in this collection.\n     * @return \n     */
    fun containsAll(elements: Collection<E>): Boolean {\n        return list.containsAll(elements)\n    }\n    // Bulk Operations\n    /** Checks if all elements in the specified collection are contained in this collection.\n     * @return \n     */
    fun containsAll(elements: Collection<E>): Boolean {\n        return list.containsAll(elements)\n    }\n    // Bulk Operations\n    /** Checks if all elements in the specified collection are contained in this collection.\n     * @return \n     */
    fun containsAll(elements: Collection<E>): Boolean {\n        return list.containsAll(elements)\n    }\n    // Modification Operations\n    /** Adds the specified element to the collection.\n     * Returns the index of the last occurrence of the specified\n     * element in the list, or -1 if the specified\n     * element is not contained in the list.\n     * @param E the type of elements contained in the collection.\n     * The collection is covariant in its element type.\n     */
    public fun add(element: E): Boolean {\n        return list.add(element)\n    }\n    /** Adds all of the elements of the specified collection to this collection.\n     * Returns the index of the last occurrence of the specified\n     * element in the list, or -1 if the specified\n     * element is not contained in the list.\n     * @param E the type of elements contained in the collection.\n     * The collection is covariant in its element type.\n     */
    public fun addAll(elements: Collection<E>): Boolean {\n        return list.addAll(elements)\n    }\n    // Modification Operations\n    /** Removes a single\n     * instance of the specified element from this collection,\n     * if it is present.\n     * @param E the type of elements contained in the collection.\n     * The collection is covariant in its element type.\n     */
    public fun remove(element: E): Boolean {\n        return list.remove(element)\n    }\n    /** Removes the specified element from the collection.\n     * @param E the type of elements contained in the collection.\n     * The collection is covariant in its element type.\n     */
    public fun remove(element: E): Boolean {\n        return list.remove(element)\n    }\n    // Positional Access Operations\n    /** Returns the element at the specified index in the list.\n     * @return \n     */
    fun get(index: Int): E {\n        return list.get(index)\n    }\n    // Positional Access Operations\n    /** Returns the index of the last occurrence of the specified\n     * element in the list, or -1 if the specified\n     * element is not contained in the list.\n     * @param E the type of elements contained in the collection.\n     * The collection is covariant in its element type.\n     */
    public fun lastIndexOf(element: E): Int {\n        return list.lastIndexOf(element)\n    }\n    // Positional Access Operations\n    /** Returns the view of the portion of this list\n     * between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).\n     * The returned list is backed by this\n     * list, so non-structural changes in the returned list are reflected in this list, and vice-versa.\n     * @param E the type of elements contained in the list.\n     * The mutable list is invariant in its element type.\n     */
    fun subList(fromIndex: Int, toIndex: Int): List<E> {\n        return list.subList(fromIndex, toIndex)\n    }\n}
```kotlin
internal constructor
    (
            start: Long,
            endInclusive: Long,
            step: Long
    ) :

    /**
     * The first element in the progression.
     */
    public val first: Long = start

    /**
     * The last element in the progression.
     */
    public val last: Long = getProgressionLastElement(start.toLong(), endInclusive.toLong(), step).toLong()

    /**
     * The step of the progression.
     */
    public val step: Long = step

    override fun iterator(): LongIterator =

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last

    override fun equals(other: Any?): Boolean =

    override fun hashCode(): Int =

    override fun toString(): String = if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"

    companion object {
        /**
         * Creates LongProgression with the specified bounds of a closed range.
         */
        public fun fromClosedRange(rangeStart: Long, rangeEnd: Long, step: Long): LongProgression = LongProgression(rangeStart, rangeEnd, step)
    }
}

/**
 * A progression of values of type `Long`.
 */
public open class LongProgression

```

The document contains code examples and explanations of classes and methods in the Kotlin programming language. It explains how to create progressions of values and how to check if a progression is empty or contains a specific value. The code is self-contained and provides examples of how to use the `IntProgression` and `LongProgression` classes, which are part of the Kotlin standard library. The explanations are clear and provide insights into the implementation details of these classes. The code examples are well-structured and easy to follow, making it accessible for developers familiar with the Kotlin language. The text is complete and provides a comprehensive understanding of the functionality provided by these classes.
fun toString() = 
"kotlin.Unit"

"/*
 * Copyright 2010-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 * implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package kotlin.annotation

import kotlin.annotation.AnnotationTarget.*

/**
 * Contains the list of code elements which are the possible annotation targets
 * Class, interface or object, annotation class is also included
 * Generic type parameter (unsupported yet)
 * Property
 * Field, including property's backing field
 * Local variable
 * Value parameter of a function or a constructor
 * Function (constructors are not included)
 * Property getter only
 * Property setter only
 * Any expression
 * File
 * Type alias
 * Determines how an annotation is stored in binary output
 * Annotation isn't stored in binary output
 * Annotation is stored in binary output, but invisible for reflection
 * Annotation is stored in binary output and visible for reflection (default retention)
 * This meta-annotation indicates the kinds of code elements which are possible targets of an annotation
 * If the target meta-annotation is not present on an annotation declaration, the annotation is applicable to the following elements:  
 * Class, [PROPERTY], [FIELD], [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION], [PROPERTY_GETTER], [PROPERTY_SETTER],
 * @property allowedTargets list of allowed annotation targets
 * @property value necessary annotation retention (RUNTIME, BINARY or SOURCE)
 * @property allowedTargets list of allowed annotation targets
 * @property annotationRetentionPolicy necessary annotation retention (RUNTIME, BINARY or SOURCE)
 * @property annotationRetentionPolicy
 * @property repeatable
 * @property mustBeDocumented
 */

/**
 * Specifies that the corresponding type parameter is not used for unsafe operations such as casts or 'is' checks
 * That means it's completely safe to use generic types as argument for such parameter.
 */

fun toString() = 
"kotlin.Unit"

"/*
 * Copyright 2010-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 * implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package kotlin.annotation

import kotlin.annotation.AnnotationTarget.*

/**
 * Contains the list of code elements which are the possible annotation targets
 * Class, interface or object, annotation class is also included
 * Generic type parameter (unsupported yet)
 * Property
 * Field, including property's backing field
 * Local variable
 * Value parameter of a function or a constructor
 * Function (constructors are not included)
 * Property getter only
 * Property setter only
 * Any expression
 * File
 * Type alias
 * Determines how an annotation is stored in binary output
 * Annotation isn't stored in binary output
 * Annotation is stored in binary output, but invisible for reflection
 * Annotation is stored in binary output and visible for reflection (default retention)
 * This meta-annotation indicates the kinds of code elements which are possible targets of an annotation
 * If the target meta-annotation is not present on an annotation declaration, the annotation is applicable to the following elements:  
 * Class, [PROPERTY], [FIELD], [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION], [PROPERTY_GETTER], [PROPERTY_SETTER],
 * @property allowedTargets list of allowed annotation targets
 * @property value necessary annotation retention (RUNTIME, BINARY or SOURCE)
 * @property allowedTargets list of allowed annotation targets
 * @property annotationRetentionPolicy necessary annotation retention (RUNTIME, BINARY or SOURCE)
 * @property annotationRetentionPolicy
 * @property repeatable
 * @property mustBeDocumented
 */

/**
 * Specifies that the corresponding type parameter is not used for unsafe operations such as casts or 'is' checks
 * That means it's completely safe to use generic types as argument for such parameter.
 */
'java/util/Map.remove(Ljava/lang/Object;Ljava/lang/Object;)Z' is defined in JDK (i.e. for major versions >= 8)

*/
@Target(AnnotationTarget.FUNCTION)
@Retention(AnnotationRetention.BINARY)
internal annotation class PlatformDependent\n
Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range \n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n * No validation on passed parameters is performed. The given parameters should satisfy the condition:\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n * @param start first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or difference of successive elements in the progression\n * @return the final element of the progression\n * @suppress
 */
@PublishedApi
internal fun getProgressionLastElement(start: Int, end: Int, step: Int): Int = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)
    else -> throw kotlin.IllegalArgumentException("Step is zero.\n")
}

@PublishedApi
internal fun getProgressionLastElement(start: Long, end: Long, step: Long): Long = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)
    else -> throw kotlin.IllegalArgumentException("Step is zero.\n")
}

@JsName("arrayIterator")
internal fun arrayIterator(array: dynamic, type: String?) = when (type) {
    null -> {
        val arr: Array<dynamic> = array
        object : Iterator<dynamic> {
            var index = 0
            override fun hasNext() = index < arr.size
            override fun next() = if (index < arr.size) arr[index++]
        }
    }
    "BooleanArray" -> booleanArrayIterator(array)
    "ByteArray" -> byteArrayIterator(array)
    "ShortArray" -> shortArrayIterator(array)
    "CharArray" -> charArrayIterator(array)
    "IntArray" -> intArrayIterator(array)
    "LongArray" -> longArrayIterator(array)
    "FloatArray" -> floatArrayIterator(array)
    "DoubleArray" -> doubleArrayIterator(array)
    else -> throw IllegalStateException("Unsupported type argument for arrayIterator: $type")
}

@JsName("booleanArrayIterator")
internal fun booleanArrayIterator(array: BooleanArray) = object : BooleanIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextBoolean() = if (index < array.size) array[index++]
}

@JsName("byteArrayIterator")
internal fun byteArrayIterator(array: ByteArray) = object : ByteIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextByte() = if (index < array.size) array[index++]
}

@JsName("shortArrayIterator")
internal fun shortArrayIterator(array: ShortArray) = object : ShortIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextShort() = if (index < array.size) array[index++]
}

@JsName("charArrayIterator")
internal fun charArrayIterator(array: CharArray) = object : CharIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextChar() = if (index < array.size) array[index++]
}

@JsName("IntArrayIterator")
internal fun intArrayIterator(array: IntArray) = object : IntIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextInt() = if (index < array.size) array[index++]
}

@JsName("LongArrayIterator")
internal fun longArrayIterator(array: LongArray) = object : LongIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextLong() = if (index < array.size) array[index++]
}

@JsName("FloatArrayIterator")
internal fun floatArrayIterator(array: FloatArray) = object : FloatIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextFloat() = if (index < array.size) array[index++]
}

@JsName("DoubleArrayIterator")
internal fun doubleArrayIterator(array: DoubleArray) = object : DoubleIterator() {
    var index = 0
    override fun hasNext() = index < array.size
    override fun nextDouble() = if (index < array.size) array[index++]
}
a.constructor(size)\n kotlin.copyArrayType(a, result)\n size = 0\n for (i in args.indices) \n val arr = args[i].asDynamic()\n for (j in 0 until arr.length) \n {\n result[size+i] = arr[j]\n }\n return result\n }\n \n@constructor(size)\n kotlin.copyArrayType(a, result)\n size = 0\n for (i in args.indices) \n val arr = args[i].asDynamic()\n for (j in 0 until arr.length) \n {\n result[size+i] = arr[j]\n }\n return result
}
internal abstract class CoroutineImpl(private val resultContinuation: Continuation<Any?>) : Continuation<Any?> {
    protected var state = 0
    protected var exceptionState = 0
    protected var result: Any? = null
    protected var exception: Throwable? = null
    protected var finallyPath: Array<Int>? = null

    public override val context: CoroutineContext = resultContinuation.context

    private var intercepted_: Continuation<Any?>? = null

    public fun intercepted(): Continuation<Any?> =
        intercepted_
            ?: (context[ContinuationInterceptor]!!.interceptContinuation(this) ?: this)

    override fun resumeWith(result: Result<Any?>) {
        var current = this
        var currentResult: Any? = result.getOrNull()
        var currentException: Throwable? = result.exceptionOrNull()

        // This loop unrolls recursion in current.resumeWith(param) to make saner and shorter stack traces on resume
        while (true) {
            with(current) {
                if (currentException == null) {
                    this.result = currentResult
                } else {
                    state = exceptionState
                    exception = currentException
                }

                try {
                    val outcome = doResume()
                    if (outcome === COROUTINE_SUSPENDED)
                        return
                    currentResult = outcome
                    currentException = null
                } catch (exception: dynamic) { // Catch all exceptions
                    currentResult = null
                    currentException = exception
                }

                releaseIntercepted() // this state machine instance is
                // unrolling recursion via loop

                if (completion is CoroutineImpl) {
                    // unrolling recursion via loop
                    current = completion
                } else {
                    // top-level completion reached -- invoke and return
                    currentException?.let {
                        completion.resumeWithException(it)
                    } ?: completion.resume(currentResult)
                    return
                }
            }
        }
    }

    private fun releaseIntercepted() {
        val intercepted = intercepted_
        if (intercepted != null && intercepted !== this) {
            context[ContinuationInterceptor]!!.releaseInterceptedContinuation(intercepted)
        }
        this.intercepted_ = CompletedContinuation // just in case
    }

    protected abstract fun doResume(): Any?
}

internal object CompletedContinuation : Continuation<Any?> {
    override val context: CoroutineContext
        get() = error("This continuation is already complete")

    override fun resumeWith(result: Result<Any?>) {
        error("This continuation is already complete")
    }

    override fun toString(): String = "This continuation is already complete"
}

@file:Suppress("UNCHECKED_CAST", "RedundantVisibilityModifier")
@SinceKotlin("1.3")
public inline class Result<out T> @PublishedApi internal constructor(  
    @PublishedApi
    internal val value: Any?
) : Serializable {
    // discovery

    /**
     * Returns `true` if this instance represents a successful outcome.
     * In this case `isFailure` returns `false`.
     */
    public val isSuccess: Boolean get() = value !is Failure

    $$
    * Returns `true` if this instance represents a failed outcome.
    * In this case `isSuccess` returns `false`.
    */
    public val isFailure: Boolean get() = value is Failure

    $$
    * Returns the encapsulated value if this instance represents [success][Result.isSuccess] or
    * `null` if it is [failure][Result.isFailure].
    * This function is a shorthand for `getOrElse { null }` (see [getOrElse]) or
    * `fold(onSuccess = { it },onFailure = { null })` (see [fold])
    */
    public val value: T

    $$
    * Returns a string `Success(v)` if this instance represents [success][Result.isSuccess] and `v` is a string representation of the value or a string `Failure(x)` if
    *
is [failure][isFailure] where 'x' is a string representation of the exception.

```kotlin
@Suppress("INAPPLICABLE_JVM_NAME")
@InlineOnly
@JvmName("success")
public inline fun <T> success(value: T): Result<T> =
    Result(value)

@Suppress("INAPPLICABLE_JVM_NAME")
@InlineOnly
@JvmName("failure")
public inline fun <T> failure(exception: Throwable): Result<T> =
    Result(createFailure(exception))
```

```kotlin
internal class Failure(@JvmField
        val exception: Throwable)
    : Serializable {
        override fun equals(other: Any?): Boolean = other is Failure && exception == other.exception
        override fun hashCode(): Int = exception.hashCode()
        override fun toString(): String = "Failure($exception)"
    }
```

```kotlin
internal fun createFailure(exception: Throwable): Any =
    Result.Failure(exception)

internal fun Result<*>.throwOnFailure() {
    if (value is Result.Failure) throw value.exception
}
```

```kotlin
public inline fun <R> runCatching(block: () -> R): Result<R> {
    return try {
        Result.success(block())
    } catch (e: Throwable) {
        Result.failure(e)
    }
}
```

```kotlin
public inline fun <T, R> T.runCatching(block: T.() -> R): Result<R> {
    return try {
        Result.success(block())
    } catch (e: Throwable) {
        Result.failure(e)
    }
}
```

```kotlin
public inline fun <T> Result<T>.getOrThrow(): T {
    throwOnFailure()
    return value as T
}
```

```kotlin
public inline fun <R, T : R> Result<T>.getOrDefault(defaultValue: R): R {
    if (isFailure) return defaultValue
    return value as T
}
```

The code provided is a simplified version of the `Result` class, a fundamental part of the Kotlin standard library, used for handling computations that may fail. The class encapsulates a result or an exception, allowing safe manipulation of potentially failed values. This is particularly useful in cases where operations may not always succeed, such as network requests or file I/O operations, ensuring that errors are handled gracefully and transparently.
return when (val exception = exceptionOrNull()) {
    null -> onSuccess(value as T)
    else -> onFailure(exception)
} }

@SinceKotlin("1.3")
public inline fun <R, T> Result<T>.map(transform: (value: T) -> R): Result<R> {
    contract {
        callsInPlace(transform, InvocationKind.AT_MOST_ONCE)
    } return when {
        isSuccess -> Result.success(transform(value as T))
        else -> Result(value)
    }
}

/**
 * Returns the encapsulated result of the given 
 * [transform] function applied to the encapsulated value
 * if this instance represents [success][Result.isSuccess] or the
 * original encapsulated [Throwable] exception if it is [failure][Result.isFailure].
 * Note, that this function rethrows any [Throwable] exception thrown by [transform] function and encapsulates it as a failure
 * See [mapCatching] for an alternative that
captures exceptions.
 */
@SinceKotlin("1.3")
public inline fun <R, T> Result<T>.mapCatching(transform: (value: T) -> R): Result<R> {
    return when {
        isSuccess -> runCatching { transform(value) }
        else -> Result(value)
    }
}

/**
 * Returns the encapsulated result of the given 
 * [transform] function applied to the encapsulated [Throwable] exception
 * if this instance represents [failure][Result.isFailure] or the
 * original encapsulated value if it is [success][Result.isSuccess].
 * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.
 * See [recoverCatching] for an alternative that captures exceptions.
 */
@SinceKotlin("1.3")
public inline fun <R, T : R> Result<T>.recover(transform: (exception: Throwable) -> R): Result<R> {
    return when (val exception = exceptionOrNull()) {
        null -> this
        else -> runCatching { transform(exception) }
    }
}

/**
 * Performs the given [action] on the encapsulated 
 * [Throwable] exception if this instance represents
 * [failure][Result.isFailure].
 * Returns the original `Result` unchanged.
 */
@SinceKotlin("1.3")
public inline fun <T> Result<T>.onFailure(action: (exception: Throwable) -> Unit): Result<T> {
    contract {
        callsInPlace(action, InvocationKind.AT_MOST_ONCE)
    } return when (val exception = exceptionOrNull()) {
        null -> this
        else -> onSuccess(action(value as T))
    }
}

/**
 * Performs the given [action] on the
 * encapsulated value if this instance represents 
 * [success][Result.isSuccess].
 * Returns the original `Result` unchanged.
 */
@SinceKotlin("1.3")
public inline fun <T> Result<T>.onSuccess(action: (value: T) -> Unit): Result<T> {
    contract {
        callsInPlace(action, InvocationKind.AT_MOST_ONCE)
    } return when (val exception = exceptionOrNull()) {
        null -> this
        else -> onFailure(action)
    }
}

// -------------------

Open Source Used In DNAC Platform Antelope 9501

/*
 * Copyright 2010-2020 JetBrains
 * s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0
 * license that can be found in the license/LICENSE.txt file.
 */

package kotlin.coroutines

import kotlin.contracts.*
import kotlin.coroutines.intrinsics.*
import kotlin.internal.InlineOnly

/**
 * Interface 
 * representing a continuation after a suspension point that returns a value of type `T`.
 */
@SinceKotlin("1.3")
public interface Continuation<in T> {
    /**
     * The context of the coroutine that corresponds to this continuation.
     */
    public val context: CoroutineContext

    /**
     * Continuation<in T> suspend
     * Resumes the execution of the corresponding coroutine passing a successful or failed [result] as the
     * return value of the last suspension point.
     */
    public infix fun continueWith(result: Result<T>): Result<T>
}

/**
 * Classes and interfaces marked with this annotation are restricted when used as receivers for extension
 * `suspend` functions.
 */
@SinceKotlin("1.3")
public interface Interface

extensions can only invoke other member or extension `suspend` functions on this particular *receiver* and are restricted from calling arbitrary suspension functions.

```kotlin
@SinceKotlin("1.3")
@Target(AnnotationTarget.CLASS)
@Retention(AnnotationRetention.BINARY)
public annotation class RestrictsSuspension

/**
 * Resumes the execution of the corresponding coroutine passing [value] as the return value of the last suspension point.
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun <T> Continuation<T>.resume(value: T): Unit =
    resumeWith(Result.success(value))

/**
 * Resumes the execution of the corresponding coroutine so that the [exception] is re-thrown right after the last suspension point.
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun <T> Continuation<T>.resumeWithException(exception: Throwable): Unit =
    resumeWith(Result.failure(exception))

/**
 * Creates a coroutine without a receiver and with result type [T].
 * This function creates and starts a new, fresh instance of suspendable computation every time it is invoked.
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <T> (suspend () -> T).startCoroutine(
    completion: Continuation<T>
) {
    createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)
}
```

```kotlin
/**
 * Obtains the current continuation instance inside suspend functions and suspends
 * the currently running coroutine. In this function both [Continuation.resume] and [Continuation.resumeWithException] can be used either synchronously or *asynchronously* from a different thread of execution. Subsequent invocation of any resume function will produce an [IllegalStateException].
 */
@SinceKotlin("1.3")
@InlineOnly
public suspend inline fun <T> suspendCoroutine(crossinline block: (Continuation<T>) -> Unit): T {
    contract { callsInPlace(block, InvocationKind.EXACTLY_ONCE) }
    return suspendCoroutineUninterceptedOrReturn { c: Continuation<T> -
        val safe = SafeContinuation(c.intercepted())
        block(safe)
        safe.getOrThrow()
    }
}
```

```kotlin
/**
 * Starts a coroutine without a receiver and with result type [T].
 * This function creates a coroutine with receiver type [R] and result type [T].
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <R, T> (suspend R.() -> T).createCoroutine(
    receiver: R,
    completion: Continuation<T>
): Continuation<Unit> =
    createCoroutineUnintercepted(receiver, completion).intercepted(), COROUTINE_SUSPENDED
```

```kotlin
/**
 * Creates a coroutine with receiver type [R] and result type [T].
 * This function creates a coroutine with receiver type [R] and result type [T].
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <R, T> (suspend R.() -> T).createCoroutine(
    receiver: R,
    completion: Continuation<T>
): Continuation<Unit> =
    createCoroutineUnintercepted(receiver, completion).intercepted(), resume(Unit)
Returns the context of the current coroutine.

@SinceKotlin("1.3")@Suppress("WRONG_MODIFIER_TARGET")@InlineOnly
public suspend inline
CoroutineContext get() { throw NotImplementedError("Implemented as intrinsic") } /* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. */

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

Starts an unintercepted coroutine without a receiver and with result type [T] and executes it until its first suspension.

- Returns the result of the coroutine or throws its exception if it does not suspend or [COROUTINE_SUSPENDED] if it suspends.
- In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception.
- The coroutine is started directly in the invoker's thread without going through the [ContinuationInterceptor] that might be present in the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation context is established.

This function is designed to be used from inside of [suspendCoroutineUninterceptedOrReturn] to resume the execution of the suspended coroutine using a reference to the suspending function.

Starts an unintercepted coroutine with receiver type [R] and result type [T] and executes it until its first suspension.

The coroutine is started directly in the invoker's thread without going through the [ContinuationInterceptor] that might be present in the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation context is established.

This function is designed to be used from inside of [suspendCoroutineUninterceptedOrReturn] to resume the execution of the suspended coroutine using a reference to the suspending function.

Creates unintercepted coroutine without receiver and with result type [T]. This function creates a new, fresh instance of suspendable computation every time it is invoked.

The coroutine is not emitted, but instead the continuation is created.

The continuation is invoked when the coroutine completes with a result or exception.

This function returns unintercepted continuation.

Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the [Continuation] that might be present in the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation context is established.

Note that [completion] of this function may get invoked in an arbitrary context.

[Continuation.intercepted] can be used to acquire the intercepted continuation.

Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of both the coroutine and [completion] happens in the invocation context established by the continuation.

Repeated invocation of any function on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or exception.

This function creates a new, fresh instance of suspendable computation every time it is invoked.

To start executing the created coroutine, invoke `resume(Unit)` on the returned [Continuation] instance.

Open Source Used In DNAC Platform Antelope 9503
coroutine completes with result or exception.

Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the\n
`[ContinuationInterceptor]` that might be present in the completion's `[CoroutineContext]`. It is the invoker's responsibility to ensure that a proper invocation context is established. Note that `[completion]` of this function may get invoked in an arbitrary context. \n
`[Continuation.intercepted]` can be used to acquire the intercepted continuation. \n
Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of the coroutine and [completion] happens in the invocation context established by the completion's `[CoroutineContext]`. Repeated invocation of any resume function on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or exception.

Relevant KDoc comments:

- `@SinceKotlin("1.3")` indicates since Kotlin version 1.3.

Other relevant KDoc comments:

- `internal inline fun <T> createCoroutineFromSuspendFunction(completion: Continuation<T>, crossinline block: () -> Any?): Continuation<Unit>`

License information:

- Kotlin
- Apache 2.0 license

Errors and exceptions:

- `public actual open class Error actual constructor(message: String?, cause: Throwable?): Error(message, cause)`
- `public actual open class Exception actual constructor(message: String?, cause: Throwable?): Exception(message, cause)`
- `public actual open class IllegalStateException actual constructor(message: String?, cause: Throwable?)`
Open Source Used In DNAC Platform Antelope 9505

```kotlin
RuntimeException(message, cause) {
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause, null)
    actual constructor(message: String?, cause: Throwable?) : this(cause, message)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class IndexOutOfBoundsException actual constructor(message: String?) : RuntimeException(message)
    actual constructor() : this(null)
}

public actual open class ConcurrentModificationException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause)
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class UnsupportedOperationException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause)
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class NumberFormatException actual constructor(message: String?) : IllegalArgumentException(message)
    actual constructor() : this(null)
}

public actual open class NullPointerException actual constructor(message: String?) : RuntimeException(message)
    actual constructor() : this(null)
}

public actual open class ClassCastException actual constructor(message: String?) : RuntimeException(message)
    actual constructor() : this(null)
}

public actual open class AssertionError
    actual constructor(message: String?, cause: Throwable?) : Error(message, cause)
    actual constructor(message: String?) : this(message)
    actual constructor(cause: Throwable?) : this(cause)

public actual open class NoSuchElementException actual constructor(message: String?) : RuntimeException(message)
    actual constructor() : this(null)
}

@SinceKotlin("1.3")
public actual open class ArithmeticException actual constructor(message: String?) : RuntimeException(message)
    actual constructor() : this(null)
}

public actual open class NoWhenBranchMatchedException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause)
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class UninitializedPropertyAccessException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause)
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. Use of this source code is
 * governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

@kotlin.internal.InlineOnly
internal inline fun jsDeleteProperty(obj: Any, property: Any) {
    js("delete obj[property]")
}

@kotlin.internal.InlineOnly
internal inline fun jsBitwiseOr(lhs: Any?, rhs: Any?): Int =
    js("lhs | rhs").unsafeCast<Int>()

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.math

/**
 * Returns this value with the sign bit same as of the 
 * `sign` value.
 * If `sign` is `NaN` the sign of the result is undefined.
 */
@SinceKotlin("1.2")
public actual fun Double.withSign(sign: Double): Double {
    val thisSignBit = js("Kotlin").doubleSignBit(this)
    val newSignBit = js("Kotlin").doubleSignBit(sign)
    return if (thisSignBit == newSignBit) this else -this
}

/**
 * Returns a bit representation of the specified floating-point value as `Long`
 * according to the IEEE 754 floating-point "double format" bit layout.
 */
@SinceKotlin("1.2")
@library("doubleToBits")
public actual fun Double.toBits(): Long =
    definedExternally

/**
 * Returns a bit representation of the specified floating-point value as `Long`
 * according to the IEEE 754 floating-point "double format" bit layout,
 * preserving 'NaN' values exact layout.
 */
@SinceKotlin("1.2")
@library("doubleToRawBits")
public actual fun Double.toRawBits(): Long =
    definedExternally

/**
 * Returns the [Double] value corresponding to a given bit representation.
 */
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual inline fun Double.Companion.fromBits(bits:
Long): Double = js("Kotlin").doubleFromBits(bits).unsafeCast<Double>()

/**
 * Returns a bit representation
 * of the specified floating-point value as [Int]*
 * according to the IEEE 754 floating-point "single format" bit
 * layout.
 * * Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent.
 * * so some [Float] values may overflow, underflow or loose their accuracy after conversion to bits and back.
 */
@SinceKotlin("1.2")
@library("floatToBits")
public actual fun Float.toBits(): Int =
    definedExternally

/**
 * Returns a bit representation of the specified floating-point value as [Int]*
 * according to the IEEE 754 floating-point "single format" bit layout,
 * preserving NaN values exact layout.
 * * Note that in Kotlin/JS [Float] range is wider than "single format" bit layout can represent.
 * * so some [Float] values
 * may overflow, underflow or loose their accuracy after conversion to bits and back.
 */
@SinceKotlin("1.2")
@library("floatToRawBits")
public actual fun Float.toRawBits(): Int =
    definedExternally

/**
 * Returns the [Float] value corresponding to a given bit representation.
 */
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual inline fun Float.Companion.fromBits(bits: Int): Float = js("Kotlin").floatFromBits(bits).unsafeCast<Float>()

internal inline fun Long(low: Int, high: Int) = js("Kotlin").Long.fromBits(low, high).unsafeCast<Long>()
internal inline val Long.low: Int get() =
    this.asDynamic().getLowBits().unsafeCast<Int>()
internal inline val Long.high: Int get() =
    this.asDynamic().getHighBits().unsafeCast<Int>()

import kotlin.reflect.KClass

@PublishedApi
internal fun <T : Annotation> KClass<*>.findAssociatedObject(annotationClass: KClass<T>): Any? {
    // This API is not supported in js-v1. Return null to be source-compatible with js-ir.
    return null
}

package kotlin.text

/**
 * Returns a string representation of this [Long] value in the specified [radix].
 * 
 * @throws IllegalArgumentException
 * when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.2")
public actual fun Long.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
https://github.com/JetBrains/kotlin/tree/master/master/libraries/stdlib
//

import kotlin.js.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Reverses elements in the list in-place.
 */
public actual fun <T> MutableList<T>.reverse(): Unit {
    val midPoint = (size / 2) - 1
    if (midPoint < 0) return
    var reverseIndex = lastIndex
    for (index in 0..midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

package kotlin.text

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
https://github.com/JetBrains/kotlin/tree/master/master/libraries/stdlib
//

import kotlin.js.*

/**
 * Returns a character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char
 * sequence.
 * * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun CharSequence.elementAt(index: Int): Char {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, length: $length") }
}

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
https://github.com/JetBrains/kotlin/tree/master/master/libraries/stdlib
//

import kotlin.js.*

/**
 * Returns an element at the given [index] if the [index] is out of bounds of this array.
 * * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun CharSequence.elementAtAt(index: Int): Char
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index") }
*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public actual fun UIntArray.elementAt(index: Int): UInt {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public actual fun ULongArray.elementAt(index: Int): ULong {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

*\@SinceKotlin("1.3")\@ExperimentalUnsignedTypes\public actual fun UByteArray.asList(): List<UByte> {
    return object : AbstractList<UByte>(), RandomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: UByte): Boolean = this@asList.contains(element)
        override fun get(index: Int): UByte {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: UByte): Int {
            if ((element as Any?) !is UByte) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: UByte): Int {
            if ((element as Any?) !is UByte) return -1
            return this@asList.lastIndexOf(element)
        }
    }
CONSTRUCTOR, PROPERTY_GETTER, PROPERTY_SETTER)\npublic actual annotation class JsName(actual
val name: String)\n\n/**\n * Denotes an `external` declaration that must be imported from native JavaScript
library.\n *\n * The compiler produces the code relevant for the target module system, for example, in case of
CommonJS,\n * it will import the declaration via the `require(...)` function.\n *\n * The annotation can be used on
top-level external declarations (classes, properties, functions) and files.\n * In case of file (which can't be `external`)
the following rule applies: all the declarations in\n * the file must be `external`. By applying `@JsModule(...)` on a
file you tell the compiler to import a JavaScript object\n * that contain all the declarations from the file.\n *\n *
Example:\n *\n * ``` kotlin\n * @JsModule(\"jquery\")\n * external abstract class JQuery() {\n * // some
declarations here\n * }\n *\n * @JsModule(\"jquery\")\n * external fun JQuery(element: Element): JQuery\n * ```\n
*\n * @property import name of a module to import declaration from.\n *
It is not interpreted by the Kotlin
compiler, it's passed as is directly to the target module system.\n *\n * @see JsNonModule\n
*/\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY, FUNCTION, FILE)\npublic
annotation class JsModule(val import: String)\n\n/**\n * Denotes an `external` declaration that can be used without
module system.\n *\n * By default, an `external` declaration is available regardless your target module system.\n *
However, by applying [JsModule] annotation you can make a declaration unavailable to *plain* module system.\n *
Some JavaScript libraries are distributed both as a standalone downloadable piece of JavaScript and as a module
available\n * as an npm package.\n * To tell the Kotlin compiler to accept both cases, you can augment [JsModule]
with the `@JsNonModule` annotation.\n *\n * For example:\n *\n * ``` kotlin\n * @JsModule(\"jquery\")\n *
@JsNonModule\n * @JsName(\"$\")\n * external abstract class JQuery() {\n * // some declarations here\n * }\n
*\n * @JsModule(\"jquery\")\n * @JsNonModule\n * @JsName(\"$\")\n * external fun JQuery(element: Element):
JQuery\n * ```\n *\n * @see JsModule\n */\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS,
PROPERTY, FUNCTION, FILE)\npublic annotation class JsNonModule\n\n/**\n * Adds prefix to `external`
declarations in a source file.\n *\n * JavaScript does not have concept of packages (namespaces). They are usually
emulated by nested objects.\n * The compiler turns references to `external` declarations either to plain unprefixed
names (in case of *plain* modules)\n * or to plain imports.\n * However, if a JavaScript library provides its
declarations in packages, you won't be satisfied with this.\n * You can tell the compiler to generate additional prefix
before references to `external` declarations using the `@JsQualifier(...)`\n * annotation.\n *\n * Note that a file
marked with the `@JsQualifier(...)` annotation can't contain non-`external` declarations.\n *\n * Example:\n *\n *
```\n * @file:JsQualifier(\"my.jsPackageName\")\n * package some.kotlinPackage\n *\n * external fun foo(x: Int)\n
*\n * external fun bar(): String\n * ```\n *\n * @property value the qualifier to add to the declarations in the
generated code.\n *
It must be a sequence of valid JavaScript identifiers separated by the `.` character.\n *
Examples of valid qualifiers are: `foo`, `bar.Baz`, `_.$0.f`.\n *\n * @see JsModule\n
*/\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.FILE)\npublic annotation class
JsQualifier(val value: String)\n\n/**\n * Exports top-level declaration on JS platform.\n *\n * Compiled module
exposes declarations that are marked with this annotation without name mangling.\n *\n * This annotation can be
applied to either files or top-level declarations.\n *\n * It is currently prohibited to export the following kinds of
declarations:\n *\n * * `expect` declarations\n * * inline functions with reified type parameters\n * * suspend
functions\n * * secondary constructors without `@JsName`\n * * extension properties\n * * enum classes\n * *
annotation classes\n *\n * Signatures of exported declarations must only contain \"exportable\" types:\n *\n * *
`dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`\n * * `BooleanArray`, `ByteArray`,
`ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`\n * * `Array<exportable-type>`\n * * Function types with
exportable parameters and return types\n * * `external` or `@JsExport` classes and interfaces\n * * Nullable
counterparts of types above\n * * Unit return type. Must not be nullable\n *\n * This annotation is experimental,
meaning that restrictions mentioned above are subject to change.\n
*/\n@ExperimentalJsExport\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY,
FUNCTION, FILE)\n@SinceKotlin(\"1.3\")\npublic actual annotation class JsExport\n","/*\n * Copyright 20102018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.jvm\n\n// these are

Open Source Used In DNAC Platform Antelope 9509


used in common generated code in stdlib

// TODO: find how to deprecate these

@Target(AnnotationTarget.FIELD)
@Retention(AnnotationRetention.SOURCE)
public actual

annotation class Volatile

@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY_SETTER)
@Retention(AnnotationRetention.SOURCE)
public actual

annotation class Synchronized

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.collections

/**
 * Provides a skeletal implementation of the
 * [MutableCollection] interface.
 * The collection is invariant in its element type.
 * * An actual abstract override fun add(element: E): Boolean
 * * @param E the type of elements contained in the collection.
 * * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * * license/LICENSE.txt file.
 */

package

/**
 * Provides a skeletal implementation of the
 * [MutableList] interface.
 * The list is invariant in its element type.
 * * An actual override fun add(element: E): Boolean
 * * @param E the type of elements contained in the list.
 * * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * * license/LICENSE.txt file.
 */

/*
 * Based on GWT AbstractList
 * Copyright 2007 Google Inc.
*/

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */
removeRange(fromIndex: Int, toIndex: Int) { 
  val iterator = listIterator(fromIndex) 
  repeat(toIndex - fromIndex) { 
    iterator.next() 
    iterator.remove() 
  } 
}

override fun removeAt(index: Int) { 
  list.removeAt(index) 
  _size-- 
}

override fun set(index: Int, element: E) { 
  list.add(fromIndex + index, element) 
  _size++ 
}

override fun addAt(index: Int) { 
  list.add(fromIndex + index, element) 
  _size++ 
}

override val size: Int get() = _size

internal override fun checkIsMutable(): Unit = checkIsMutable()

AbstractList.orderedEquals(this, other)
}

override fun equals(other: Any?): Boolean {
  if (other === this) return true
  if (other !is List<*>) return false

  return AbstractList.orderedEquals(this, other)
}

override fun hashCode(): Int = AbstractList.orderedHashCode(this)


package kotlin.collections

/**
 * Provides a [MutableList] implementation, which uses a resizable array as its backing storage.
 * This implementation doesn't provide a way to manage capacity, as backing JS array is resizeable itself.
 * There is no speed advantage to pre-allocating array sizes in JavaScript, so this implementation does not include any of the
 * capacity and "growth increment" concepts.
 */
public actual open class ArrayList<E> internal constructor(private var array: Array<Any?>) :
    AbstractMutableList<E>(), MutableList<E>, RandomAccess{
    private var isReadOnly: Boolean = false

    /** Creates an empty [ArrayList]. */
    public actual constructor() : this(emptyArray()) {}

    /** Creates an empty [ArrayList].
     * @param initialCapacity initial capacity (ignored)
     */
    @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
    public actual constructor(initialCapacity: Int = 0) : this(emptyArray()) {}

    /** Creates an [ArrayList] filled from the
     * [elements] collection.
     */
    public actual constructor(elements: Collection<E>) : this(elements.toTypedArray<Any?>()) {
        @PublishedApi
        internal fun build(): List<E> {
            checkIsMutable()
            isReadOnly = true
            return this
        }

        @Suppress("UNCHECKED_CAST")
        actual override fun get(index: Int): E = array[index] as E

        actual override fun set(index: Int, element: E): E {
            checkIsMutable()
            rangeCheck(index)
            @Suppress("UNCHECKED_CAST")
            return array[index].apply { array[index] = element } as E
        }

        actual override fun add(element: E): Boolean {
            checkIsMutable()
            array.asDynamic().push(element)
            modCount++
            return true
        }

        actual override fun add(index: Int, element: E): Unit {
            checkIsMutable()
            insertionRangeCheck(index)
            if (index == size) return addAll(elements)
            if (elements.isEmpty()) return false
            when (index) {
                size -> return addAll(elements)
                0 -> array.addAll(elements)
                else -> array = array.copyOfRange(0, index).asDynamic().concat(elements.toTypedArray<Any?>(), array.copyOfRange(index, size))
            }
            modCount++
            return true
        }

        actual override fun removeAt(index: Int): E {
            checkIsMutable()
            rangeCheck(index)
            modCount++
            return if (index == lastIndex)
                array.asDynamic().pop()
            else
                array.asDynamic().splice(index, 1)[0]
        }

        actual override fun remove(element: E): Boolean {
            checkIsMutable()
            for (index in array.indices) {
                if (array[index] == element) {
                    array.asDynamic().splice(index, 1)
                    modCount++
                    return true
                }
            }
            return false
        }

        override fun removeRange(fromIndex: Int, toIndex: Int) {
            checkIsMutable()
            modCount++
            array.asDynamic().splice(fromIndex, toIndex - fromIndex)
        }

        actual override fun clear() {
            checkIsMutable()
            array = emptyArray()
            modCount++
        }

        override fun toString() = arrayToString(array)
        override fun toArray(): Array<Any?> = js("[]").slice.call(array)

        internal override fun checkIsMutable() {
            if (isReadOnly) throw UnsupportedOperationException()
        }

        private fun rangeCheck(index: Int) = index.apply {
            AbstractList.checkElementIndex(index, size)
        }

        private fun insertionRangeCheck(index: Int) = index.apply {
            AbstractList.checkPositionIndex(index, size)
        }
}
internal fun <T> mergeSort(array: Array<out T>, start: Int, end: Int, comparator: Comparator<in T>): Array<T> {
    if (start == end) {
        return array
    }

    val median = (start + end) / 2
    val left = mergeSort(array, buffer, start, median, comparator)
    val right = mergeSort(array, buffer, median + 1, end, comparator)

    val target = if (left === buffer) array else buffer

    // Merge.
    var leftIndex = start
    var rightIndex = median + 1
    for (i in start..end) {
        when {
            leftIndex <= median && rightIndex <= end -> {
                val leftValue = left[leftIndex]
                val rightValue = right[rightIndex]
                if (comparator.compare(leftValue, rightValue) <= 0) {
                    target[i] = leftValue
                    leftIndex++
                } else {
                    target[i] = rightValue
                    rightIndex++
                }
            }
            leftIndex <= median -> {
                target[i] = left[leftIndex]
                leftIndex++
            }
            else /* rightIndex <= end */ -> {
                target[i] = right[rightIndex]
                rightIndex++
            }
        }
    }

    return target
}
Based on GWT AbstractHashMap

```
* Copyright 2008 Google Inc.
*/

package kotlin.collections

/**
 * Hash table based implementation of the [MutableMap] interface.
 * This implementation makes no guarantees regarding the order of enumeration of [keys], [values] and [entries] collections.
 */
Classes that extend HashMap and implement `build()` (freezing) operation
have to make sure mutating methods check `checkIsMutable`

```

```
open class HashMap<K, V> : AbstractMutableMap<K, V>, MutableMap<K, V> {

    private inner class EntrySet : AbstractMutableSet<MutableEntry<K, V>>() {

        override fun add(element: MutableEntry<K, V>): Boolean = throw UnsupportedOperationException("Add is not supported on entries")
        override fun clear() {
            this@HashMap.clear()
        }

        override operator fun contains(element: MutableEntry<K, V>): Boolean = containsEntry(element)
        override fun remove(element: MutableEntry<K, V>): Boolean {
            if (contains(element)) {
                this@HashMap.remove(element.key)
                return true
            }
            return false
        }

        override val size: Int get() = this@HashMap.size
    }

    /**
     * Internal implementation of the map: either string-based or hashcode-based.
     */
    private val internalMap: InternalMap<K, V>

    private val equality: EqualityComparator

    internal constructor(internalMap: InternalMap<K, V>) : super() {
        this.internalMap = internalMap
        this.equality = internalMap.equality
    }

    actual constructor() : this(InternalHashCodeMap(EqualityComparator.HashCode))

    actual constructor(initialCapacity: Int, loadFactor: Float = 0.0f) : this(initialCapacity, 0.0f)

    actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)

    /**
     * Constructs an instance of [HashMap] filled with the contents of the specified [original] map.
     */
    actual constructor(original: Map<out K, V>) :
```

```
```
open class HashSet<E> : AbstractMutableSet<E>, MutableSet<E> {

    internal val map: HashMap<E, Any>

    /**
     * Constructs a new [HashSet].
     */
    actual constructor() {
        map = HashMap<E, Any>()
    }

    /**
     * Constructs a new [HashSet] filled with the elements of the specified collection.
     */
    actual constructor(elements: Collection<E>) {
```

```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
/**
 * Constructs a new empty [HashSet].
 *
 * @param initialCapacity the initial capacity (ignored)
 * @param loadFactor the load factor (ignored)
 * @throws IllegalArgumentException if the initial capacity or load factor are negative
 */

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
actual constructor(initialCapacity: Int, loadFactor: Float = 0.0f) {
    map = HashMap<E, Any>(initialCapacity, loadFactor)
}

actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)

/**
 * Protected constructor to specify the underlying map. This is used by
 * * LinkedHashSet
 * * @param map underlying map to use.
 */
internal constructor(map: HashMap<E, Any>) {
    this.map = map
}

actual override fun add(element: E): Boolean {
    val old = map.put(element, this)
    return old == null
}

actual override fun clear() {
    map.clear()
}

actual override operator fun contains(element: E): Boolean = map.containsKey(element)

actual override fun isEmpty(): Boolean = map.isEmpty()

actual override fun iterator(): MutableIterator<E> = map.keys.iterator()

actual override fun remove(element: E): Boolean = map.remove(element) != null

actual override val size: Int get() = map.size

/**
 * Creates a new instance of the specialized implementation of [HashSet] with the specified [String] elements,
 * which elements the keys as properties of JS object without hashing them.
 */
public fun stringSetOf(vararg elements: String): HashSet<String> {
    return HashSet(stringMapOf<Any>()).apply { addAll(elements) }
}

/**
 * A simple wrapper around JavaScriptObject to provide [java.util.Map]-like semantics for any
 * key type.
 */
internal class InternalHashCodeMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {

    private var backingMap: dynamic = createJsMap()
    override var size: Int = 0
        private set

    override fun put(key: K, value: V): V? {
        val hashCode = equality.getHashCode(key)
        val chainOrEntry = getChainOrEntryOrNull(hashCode) ?: return null
        if (chainOrEntry !is Array<*>) {
            val entry: SimpleEntry<K, V> = chainOrEntry
            if (equality.equals(entry.key, key)) {
                return entry.setValue(value)
            } else {
                backingMap[hashCode] = arrayOf(entry, SimpleEntry(key, value))
                size++
                return null
            }
        } else {
            val chain: Array<MutableEntry<K, V>> = chainOrEntry
            for (index in chain.indices) {
                val entry = chain[index]
                if (equality.equals(key, entry.key)) {
                    if (chain.size == 1) {
                        chain.asDynamic().length = 0
                        jsDeleteProperty(backingMap, hashCode)
                    } else {
                        jsDeleteProperty(backingMap, hashCode)
                        size--
                        jsDeleteProperty(backingMap, hashCode)
                    }
                    return entry.setValue(value)
                } else {
                    return null
                }
            }
        }
        return null
    }

    override fun remove(key: K): V? {
        val hashCode = equality.getHashCode(key)
        val chainOrEntry = getChainOrEntryOrNull(hashCode) ?: return null
        if (chainOrEntry !is Array<*>) {
            val entry: MutableEntry<K, V> = chainOrEntry
            if (equality.equals(entry.key, key)) {
                return entry.setValue(value)
            } else {
                return null
            }
        } else {
            val chain = Array<MutableEntry<K, V>>
            for (index in chain.indices) {
                val entry = chain[index]
                if (equality.equals(key, entry.key)) {
                    if (chain.size == 1) {
                        chain.asDynamic().length = 0
                        jsDeleteProperty(backingMap, hashCode)
                        size--
                    } else {
                        jsDeleteProperty(backingMap, hashCode)
                    }
                }
            }
        }
        return null
    }

    override fun clear() {
        backingMap = createJsMap()
        size = 0
        override fun contains(key: K):
Boolean = getEntry(key) != null
override fun get(key: K): V? = getEntry(key) ?: value
private fun getEntry(key: K): MutableEntry<K, V>? {  
    val chainOrEntry = getChainOrEntryOrNull(quality.getHashCode(key)) ?: return null

    override fun get(key: K, V): chainOrEntry
    if (quality.equals(entry.key, key)) {  
        return entry
    } else {  
        return null
    } else {  
        val chain = Array<MutableEntry<K, V>>
        V>> = chainOrEntry
        return chain.findEntryInChain(key)
    }

    override fun iterator(): MutableIterator<MutableEntry<K, V>> {

        return object : MutableIterator<MutableEntry<K, V>> {
            private fun computeNext(): Int {
                if (chainOrEntry != null && isChain) {
                    val chainSize: Int = chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>().size
                    if (++itemIndex < chainSize)
                        return 0
                }

                if (++keyIndex < keys.size) {
                    chainOrEntry = backingMap[keys[keyIndex]]
                    isChain = chainOrEntry is Array<*>
                    itemIndex = 0
                    return 0
                } else {
                    chainOrEntry = null
                    return 1
                }
            }

            override fun hasNext(): Boolean {
                if (state == -1)
                    state = computeNext()
                return state == 0
            }

            override fun next(): MutableEntry<K, V> {
                if (!hasNext()) throw NoSuchElementException()
                val lastEntry = if (isChain) {
                    chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>()[itemIndex]
                } else {
                    chainOrEntry.unsafeCast<MutableEntry<K, V>>()
                }
                this.lastEntry = lastEntry
                state = -1
                return lastEntry
            }

            override fun remove() {
                checkNotNull(lastEntry)
                this@InternalHashCodeMap.remove(lastEntry!!.key)
                lastEntry = null
            }
        }
    }

    private fun getChainOrEntryOrNull(hashCode: Int): dynamic {
        val chainOrEntry = backingMap[hashCode]
        return if (chainOrEntry === undefined) null else chainOrEntry
    }
}

}
```kotlin
    override fun put(key: K, value: V): V? {
        require(key is String)
        val oldValue = backingMap[key]
        backingMap[key] = value
        if (oldValue !== undefined) {
            size++
            structureChanged(host)
            return oldValue.unsafeCast<V>()
        } else {
            size--
            structureChanged(host)
            return null
        }
    }

    override fun remove(key: K): V? {
        if (key !is String) return null
        val value = backingMap[key]
        if (value !== undefined) {
            jsDeleteProperty(backingMap, key)
            size--
            structureChanged(host)
            return value.unsafeCast<V>()
        } else {
            valueMod++
            return null
        }
    }

    override fun clear() {
        backingMap = createJsMap()
        size = 0
    }

    override fun iterator(): MutableIterator<MutableEntry<K, V>> {
        return object :
            MutableIterator<MutableEntry<K, V>> {
                private val keys: Array<String> = js("Object").keys(backingMap)
                private val iterator = keys.iterator()
                private var lastKey: String? = null

                override fun hasNext(): Boolean = iterator.hasNext()

                override fun next(): MutableEntry<K, V> {
                    val key = iterator.next()
                    lastKey = key
                    @Suppress("UNCHECKED_CAST")
                    return newMapEntry(key as K)
                }

                override fun remove() {
                    @Suppress("UNCHECKED_CAST")
                    this@InternalStringMap.remove(checkNotNull(lastKey) as K)
                }

                override fun override fun removeValue(key: K, value: V): V? {
                    if (key !== undefined) {
                        backingMap[key] = value
                        return oldValue
                    } else {
                        return null
                    }
                }
            }
        }
    }

    private fun newMapEntry(key: K): MutableEntry<K, V> = object : MutableEntry<K, V> {
        override val key: K get() = key
        override val value: V get() = this@InternalStringMap[key].unsafeCast<V>()

        override fun setValue(newValue: V): V = this@InternalStringMap.put(key, newValue).unsafeCast<V>()

        override fun hashCode(): Int = AbstractMap.entryHashCode(this)
        override fun toString(): String = AbstractMap.entryToString(this)
        override fun equals(other: Any?): Boolean = AbstractMap.entryEquals(this, other)
    }
```

override fun add(element: MutableEntry<K, V>): Boolean = throw
UnsupportedOperationException("Add is not supported on entries")
override fun clear() {
this@LinkedHashMap.clear()
}
override operator fun contains(element: MutableEntry<K, V>): Boolean = containsEntry(element)
override operator fun iterator(): MutableIterator<MutableEntry<K, V>> = EntryIterator()
override fun remove(element: MutableEntry<K, V>): Boolean {
    checkIsMutable()
    if (contains(element)) {
        this@LinkedHashMap.remove(element.key)
        return true
    } else {
        return false
    }
}
override fun checkIsMutable(): Unit =
this@LinkedHashMap.checkIsMutable()
}

/*
 * The head of the insert order chain, which is a doubly-linked circular
 * list.
 */
private var head: ChainEntry<K, V>? = null

/**
 * Add this node to the end of the chain.
 */
private fun ChainEntry<K, V>.addToEnd() {
    // This entry is not in the list.
    prev = null
    val _head = head
    if (_head == null) {
        head = this
        next = this
        prev = this
    } else {
        // Chain is valid.
        prev = _tail
        next = _head
        _head.prev = this
        _tail.next = this
    }
}

/**
 * Remove this node from the chain it is a part of.
 */
private fun ChainEntry<K, V>.remove() {
    if (this.next === this) {
        // if this is single element, remove head
        head = null
    } else {
        if (head === this) {
            // if this is first element, move head to next
            head = next
        }
        next!!.prev = prev
        prev!!.next = next
    }
}

/*
 * The hashmap that keeps track of our entries and
 * the chain. Note that we duplicate the key here to eliminate changes to HashMap and minimize the
 * code here, at the expense of additional space.
 */
private val map: HashMap<K, ChainEntry<K, V>>

private var isReadOnly: Boolean = false

/**
 * Constructs an empty [LinkedHashMap] instance.
 */
actual constructor() : super() {
    map = HashMap<K, ChainEntry<K, V>>()
}

internal constructor(backingMap: HashMap<K, Any>) : super() {
    @Suppress("UNCHECKED_CAST") // expected to work due to erasure
    map = backingMap as HashMap<K, ChainEntry<K, V>>
}

/**
 * Constructs an empty [LinkedHashMap] instance.
 * @param  initialCapacity the initial capacity
 * @param  loadFactor      the load factor (ignored)
 * @throws IllegalArgumentException if
 * the initial capacity or load factor are negative
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)

/**
 * Constructs an instance of [LinkedHashMap] filled with the contents of the specified [original] map.
 */
actual constructor(original: Map<out K, V>) {
    map = HashMap<K, ChainEntry<K, V>>()
    this.putAll(original)
}

@PublishedApi
internal fun build(): Map<K, V> {
    checkIsMutable()
    isReadOnly = true
    return this
}

actual override fun clear() {
    checkIsMutable()
    map.clear()
    head = null
}

//    override fun clone(): Any {
//        return LinkedHashMap(this)
//    }

actual override fun containsKey(key: K): Boolean = map.containsKey(key)

actual override fun containsValue(value: V): Boolean {
    var node: ChainEntry<K, V> = head
    do {
        if (node.value == value) return true
        node = node.next!
    } while (node !== head)
    return false
}

override fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()

actual override operator fun get(key: K): V? = map.get(key)

actual override fun put(key: K, value: V): V? {
    val newEntry = ChainEntry(key, value)
    map.put(key, newEntry)
    newEntry.addToEnd()
    return null
}

actual override fun remove(key: K): V? {
    val old = map.remove(key)
    if (old == null) return null
    entry.remove() else return old.setValue(value)
}

actual override fun size: Int get() = map.size

internal actual override fun checkIsMutable(): Unit =
if (isReadOnly) throw
UnsupportedOperationException()
    }
}

/**
 * Constructs the specialized implementation of
 * \[LinkedHashMap\] with \[String\] keys, which stores the keys as properties of \[JS\] object without hashing them.
 */
public fun <V> linkedStringMapOf(vararg pairs: Pair<String, V>): LinkedHashMap<String, V> {
    return
}

/**
 * Constructs a new empty \[LinkedHashSet\].
 */
actual constructor() : super(LinkedHashSet<String, Any>()))

/**
 * Constructs a new empty \[LinkedHashSet\].
 */
actual constructor(initialCapacity: Int, loadFactor: Float = 0.0f) {
    super(LinkedHashSet().build(), this(initialCapacity, loadFactor)
}

/**
 * Creates a new instance of the specialized implementation of
 * \[LinkedHashSet\] with the specified \[String\] elements, which elements the keys as properties of \[JS\] object without hashing them.
 */
public fun linkedStringSetOf(vararg elements: String): LinkedHashSet<String> {
    return
}

/**
 * Creates a new instance of the specialized implementation of
 * \[LinkedHashSet\] with \[String\] keys, which stores the keys as properties of \[JS\] object without hashing them.
 */
public fun <V> linkedStringMapOf(vararg pairs: Pair<String, V>): LinkedHashMap<String, V> {
    return
}

/**
 * Constructs a new empty \[LinkedHashSet\].
 */
actual constructor() : super(LinkedHashSet<String, Any>() +

/**
 * Constructs a new empty \[LinkedHashSet\].
 */
actual constructor(initialCapacity: Int) {
    this(initialCapacity, 0.0f)

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin

import

kotlin.

contracts.*


@kotlin.internal.InlineOnly
public actual inline fun <R> synchronized(lock: Any, block: ()

/*
 * Copyright 2010-2020
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.io

internal abstract class BaseOutput {
    open fun println() {
        print("\n")
    }

    open fun println(message: Any?) {
        print(message)
        println()
    }

    abstract fun print(message: Any?)

    open fun flush() {}
}

/** JsName used to make the declaration available outside of module to test it
 * and use at try.kotl.in
 */
@JsName("NodeJsOutput")
internal class NodeJsOutput(val outputStream: dynamic) : BaseOutput() {
    override fun print(message: Any?) {
        // TODO: Using local variable because of bug in block decomposition
        // lowering in IR backend
        val messageString = String(message)
        outputStream.write(messageString)
    }
}

/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in
 * and use at try.kotl.in
 */
@JsName("OutputToConsoleLog")
internal class OutputToConsoleLog : BaseOutput() {
    override fun println(message: Any?) {
        println(message)
    }

    override fun println() {
        console.log("")
    }

    override fun print(message: Any?) {
        console.log(message)
    }
}

/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in
 * and use at try.kotl.in
 */
@JsName("BufferedOutput")
internal open class BufferedOutput : BaseOutput() {
    var buffer = ""

    override fun print(message: Any?) {
        buffer += String(message)
    }

    override fun flush() {
        buffer = ""
    }
}

/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in
 * and use at try.kotl.in
 */
@JsName("BufferedOutputToConsoleLog")
internal class BufferedOutputToConsoleLog : BufferedOutput()

override fun print(message: Any?) {
    var s = String(message)
    val i = s.negativeLastIndexOf("\n", 0)
    if (i >= 0) {
        buffer += s.substring(0, i)
    }
    flush()
    buffer += s
}

override fun println(message: Any?) {
    println(buffer)
}

override fun println() {
    println(buffer)
}
"kotlinx.dom.hasClass")
)
@DeprecatedSinceKotlin(warningSince = "1.4")
inline fun
Element.hasClass(cssClass: String): Boolean = this.newHasClass(cssClass)

/**
 * Adds CSS class to element.
 * Has no effect if all specified classes are already in class attribute of the element
 * @return true if at least one class has been added
 */
@LowPriorityInOverloadResolution@Deprecated
inline fun Element.addClass(vararg cssClasses: String): Boolean = this.newAddClass(*cssClasses)

/**
 * Removes all [cssClasses] from element. Has no effect if all specified classes are missing in class attribute of the element
 * @return true if at least one class has been removed
 */
@LowPriorityInOverloadResolution@Deprecated
inline fun Element.removeClass(vararg cssClasses: String): Boolean = this.newRemoveClass(*cssClasses)

package kotlin.dom

import org.w3c.dom.Element
import org.w3c.dom.Node
import kotlinx.dom.isElement as newIsElement
import kotlinx.dom.isText as newIsText

/**
 * Gets a value indicating whether this node is a TEXT_NODE or a CDATA_SECTION_NODE.
 * @return true if this node is a [TEXT_NODE] or a [CDATA_SECTION_NODE]
 */
@LowPriorityInOverloadResolution@Deprecated
public val Node.isText: Boolean
    inline get() = this.newIsText

/**
 * Gets a value indicating whether this node is an [Element].
 * @return true if this node is an [Element]
 */
@LowPriorityInOverloadResolution@Deprecated
public val Node.isElement: Boolean
    inline get() = this.newIsElement

package org.w3c.dom.events

public fun EventListener(handler: (Event) -> Unit): EventListener = EventListenerHandler(handler)

private class EventListenerHandler(private val handler: (Event) -> Unit) : EventListener {
    public override fun handleEvent(event: Event) {
        handler(event)
    }

    public override fun toString(): String = "EventListenerHandler($handler)"
}

package org.w3c.dom

public external interface ItemArrayLike<out T> {
    val length: Int
    fun item(index: Int): T?
}

/**
 * Returns the view of this `ItemArrayLike<T>` collection as `List<T>`
 */
public fun <T> ItemArrayLike<T>.asList(): List<T> = object : AbstractList<T>() {
    override val size: Int get() = this@asList.length

    override fun get(index: Int): T = when (index) {
        in 0..lastIndex ->
            this@asList.item(index).unsafeCast<T>()
        else -> throw IndexOutOfBoundsException("index $index is not in range [0..$lastIndex]"
    }
}

package kotlin.dom

import org.w3c.dom.Element
import org.w3c.dom.Node
import kotlin.internal.LowPriorityInOverloadResolution
import kotlinx.dom.appendText as newAppendText
import kotlinx.dom.clear as newClear

/** Removes all the children from this node.
 */
@LowPriorityInOverloadResolution@Deprecated
inline fun Node.clear() = this.newClear()

/**
 * Creates text node and append it to the element.
 * @return this element
 */
@LowPriorityInOverloadResolution
@Deprecated
public fun Node.appendText(text: String): Element = this.newAppendText(text)
use 'kotlinx.dom.appendText' instead."

```kotlin
replaceWith = ReplaceWith("this.appendText(text)",
"kotlinx.dom.appendText")
```

@DeprecatedSinceKotlin(warningSince = "1.4")
inline fun Element.appendText(text: String): Element = this.newAppendText(text)

```/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

/**
 * Reinterprets this value as a value of the [dynamic type](/docs/reference/dynamic-type.html).
 */
@kotlin.internal.InlineOnly
public inline fun Any?.asDynamic(): dynamic = this

/**
 * Reinterprets this value as a value of the specified type [T] without any actual type checking.
 * @kotlin.internal.NoInfer T = this
 */
@kotlin.internal.InlineOnly
public inline fun <T> Any?.unsafeCast(): @kotlin.internal.NoInfer T = this

/**
 * Reinterprets this `dynamic` value as a value of the specified type [T] without any actual type checking.
 */
@kotlin.internal.DYNAMIC_EXTENSION
@kotlin.internal.InlineOnly
public inline fun <T> dynamic.unsafeCast(): @kotlin.internal.NoInfer T = this

/**
 * Allows to iterate this `dynamic` object in the following cases:
 * - when it has an `iterator` function,
 * - when it is an array
 * - when it is an instance of `{kotlin.collections.Iterable}
 */
@kotlin.internal.DYNAMIC_EXTENSION
public operator fun dynamic.iterator(): Iterator<dynamic> {
    val r: Any? = this

    return when {
        this["iterator"] != null ->
            r["iterator"]().iterator()
        isArray(r) ->
            r.unsafeCast<Array<*>>().iterator().iterator()
        else ->
            (r as Iterable<*>).iterator()
    }
}
```

```/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmName("GroupingKt")
@file:kotlin.jvm.JvmMultifileClass

package kotlin.collections

/**
 * Represents a source of elements with a `keyOf` function, which can be applied to each element to get its key.
 * A `Grouping` structure serves as an intermediate step in group-and-fold operations:
 * they group elements by their keys and then fold each group with some aggregating operation.
 * It is created by attaching a `keySelector: (T) -> K` function to a source of elements.
 * For the list of group-and-fold operations available, see the `extension functions` for `Grouping`.
 */
@SinceKotlin("1.1")
public interface Grouping<T, out K> {
    /**
     * Groups elements from the `Grouping` source by key and counts elements in each group.
     */
    fun sourceIterator(): Iterator<T>

    /**
     * Extracts the key of an `element`.
     */
    fun keyOf(element: T): K
}

/**
 * Groups elements from the `Grouping` source by key and applies `operation` to the elements of each group sequentially,
 * passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
 * The key for each element is provided by the `Grouping::keyOf` function.
 */
@SinceKotlin("1.1")
public fun Grouping<T, K>.groupBy(operation: (T, K, Int) -> Int): Map<K, Int> =
    fold(0) { acc, e, k, a -> acc + operation(e, k, a) }

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@package kotlin.js
```

```/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

/**
 * Represents a source of elements with a `keyOf` function, which can be applied to each element to get its key.
 * A `Grouping` structure serves as an intermediate step in group-and-fold operations:
 * they group elements by their keys and then fold each group with some aggregating operation.
 * It is created by attaching a `keySelector: (T) -> K` function to a source of elements.
 * For the list of group-and-fold operations available, see the `extension functions` for `Grouping`.
 */
@SinceKotlin("1.1")
public interface Grouping<T, out K> {
    /**
     * Groups elements from the `Grouping` source by key and counts elements in each group.
     */
    fun sourceIterator(): Iterator<T>

    /**
     * Extracts the key of an `element`.
     */
    fun keyOf(element: T): K
}

/**
 * Groups elements from the `Grouping` source by key and applies `operation` to the elements of each group sequentially,
 * passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
 * The key for each element is provided by the `Grouping::keyOf` function.
 */
@SinceKotlin("1.1")
public fun Grouping<T, K>.groupBy(operation: (T, K, Int) -> Int): Map<K, Int> =
    fold(0) { acc, e, k, a -> acc + operation(e, k, a) }

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@package kotlin.js
```

```/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

/**
 * Represents a source of elements with a `keyOf` function, which can be applied to each element to get its key.
 * A `Grouping` structure serves as an intermediate step in group-and-fold operations:
 * they group elements by their keys and then fold each group with some aggregating operation.
 * It is created by attaching a `keySelector: (T) -> K` function to a source of elements.
 * For the list of group-and-fold operations available, see the `extension functions` for `Grouping`.
 */
@SinceKotlin("1.1")
public interface Grouping<T, out K> {
    /**
     * Groups elements from the `Grouping` source by key and counts elements in each group.
     */
    fun sourceIterator(): Iterator<T>

    /**
     * Extracts the key of an `element`.
     */
    fun keyOf(element: T): K
}

/**
 * Groups elements from the `Grouping` source by key and applies `operation` to the elements of each group sequentially,
 * passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
 * The key for each element is provided by the `Grouping::keyOf` function.
 */
@SinceKotlin("1.1")
public fun Grouping<T, K>.groupBy(operation: (T, K, Int) -> Int): Map<K, Int> =
    fold(0) { acc, e, k, a -> acc + operation(e, k, a) }

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@package kotlin.js
```
@param operation function is invoked on each element with the following parameters:
- `key`: the key of the group this element belongs to;
- `accumulator`: the current value of the accumulator of the group, can be `null` if it's the first `element` encountered in the group;
- `element`: the element from the source being aggregated;
- `first`: indicates whether it’s the first `element` encountered in the group.

@return a [Map] associating the key of each group with the result of aggregation of the group elements.

samples.collections.Grouping.aggregateByRadix

@SinceKotlin("1.1")
public inline fun <T, K, R> Grouping<T, K>.aggregate(
    operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R
): Map<K, R> {
    return aggregateTo(mutableMapOf<K, R>(), operation)
}

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in the given [destination] map.
 * It's invoked with parameters:
 * - `key`: the key of the group this element belongs to;
 * - `accumulator`: the current value of the accumulator of the group, can be `null` if it's the first `element` encountered in the group;
 * - `element`: the element from the source being aggregated;
 * - `first`: indicates whether it's the first `element` encountered in the group.
 * If the [destination] map already has a value corresponding to some key, then the elements being aggregated for that key are never considered as `first`.

@return the [destination] map associating the key of each group with the result of aggregation of the group elements.

@sample samples.collections.Grouping.aggregateByRadixTo

@SinceKotlin("1.1")
public inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.aggregateTo(
    destination: M,
    operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R
): M {
    for (e in this.sourceIterator()) {
        val key = keyOf(e)
        val accumulator = destination[key]
        destination[key] = operation(key, accumulator, e, accumulator == null && !destination.containsKey(key))
    }
    return destination
}

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
 * An initial value of accumulator is provided by [initialValueSelector] function.

@SinceKotlin("1.1")
public inline fun <T, K, R> Grouping<T, K>.fold(
    initialValueSelector: (key: K, element: T) -> R,
    operation: (key: K, accumulator: R, element: T) -> R
): Map<K, R> =
    @Suppress("UNCHECKED_CAST")
    aggregate { key, acc, e -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }

@SinceKotlin("1.1")
public inline fun <T, K, R> Grouping<T, K>.foldByEvenLengthWithComputedInitialValue(
    initialValueSelector: (key: K, element: T) -> R,
    operation: (key: K, accumulator: R, element: T) -> R
): Map<K, R> =
    @Suppress("UNCHECKED_CAST")
    aggregate { key, acc, e, first -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }

@sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue

@sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValueTo

@SinceKotlin("1.1")
public inline fun <T, K, R> Grouping<T, K>.foldByEvenLengthWithComputedInitialValueTo(
    initialValueSelector: (key: K, element: T) -> R,
    operation: (key: K, accumulator: R, element: T) -> R
): Map<K, R> =
    @Suppress("UNCHECKED_CAST")
    aggregate { key, acc, e, first -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }
inline fun <T, K, R : MutableMap<in K, R>> Grouping<T, K>.foldTo(
@Suppress("UNCHECKED_CAST")
    aggregateTo(destination) { key, acc, e, first -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, \n * passing the previously accumulated value and the current element as arguments, \n * and stores the results in a new map. \n * An initial value of accumulator is the same \n * initialValue for each group. \n * @param operation a function that is invoked on each element with the following parameters: \n * - `accumulator`: the current value of the accumulator of the group; \n * - `element`: the element from the source being accumulated. \n * @return a [Map] associating the key of each group with the result of accumulating the group elements. \n * @sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue
 */
@SinceKotlin("1.1")
public inline fun <T, K, R> Grouping<T, K>.fold(
@Suppress("UNCHECKED_CAST")
    aggregate { _, acc, e, first -> operation(if (first) initialValue else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group
 * sequentially starting from the second element of the group, \n * passing the previously accumulated value and the current element as arguments, \n * and stores the results in a new map. \n * An initial value of accumulator is the first element of the group. \n * @param operation a function that is invoked on each subsequent element of the group with the following parameters: \n * - `key`: the key of the group this element belongs to; \n * - `accumulator`: the current value of the accumulator of the group; \n * - `element`: the element from the source being accumulated. \n * @return a [Map] associating the key of each group with the result of accumulating the group elements. \n * @sample samples.collections.Grouping.reduceByMaxVowels
 */
@SinceKotlin("1.1")
public inline fun <S, T : S, K> Grouping<T, K>.reduce(
    operation: (key: K, accumulator: S, element: T) -> S): Map<K, S> =
    aggregate { key, acc, e, first ->
@Suppress("UNCHECKED_CAST")
        if (first) e else operation(key, acc as S, e)
    }

/**
 * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group
 * sequentially starting from the second element of the group, \n * passing the previously accumulated value and the current element as arguments, \n * and stores the results in the given [destination] map. \n * An initial value of accumulator is the same \n * initialValue for each group. \n * @param operation a function that is invoked on each element with the following parameters: \n * - `accumulator`: the current value of the accumulator of the group; \n * - `element`: the element from the source being accumulated. \n * @return a [Map] associating the key of each group with the result of accumulating the group elements. \n * @sample samples.collections.Grouping.reduceByMaxVowelsTo
 */
@SinceKotlin("1.1")
public inline fun <S, T : S, K, M : MutableMap<in K, S>> Grouping<T, K>.reduceTo(
    destination: M, \n    operation: (key: K, accumulator: S, element: T) -> S) as M { \n        aggregateTo(destination) { key, acc, e, first ->
@Suppress("UNCHECKED_CAST")
        if (first) e else operation(key, acc as S, e)
    }

Open Source Used In DNAC Platform Antelope 9526
/**
 * Groups elements from the [Grouping] source by key and counts elements in each group to the given [destination] map.
 * If the [destination] map already has a value corresponding to the key of some group, that value is used as an initial value of the counter for that group.
 * @return the [destination] map associating the key of each group with the count of elements in the group.
 * @sample samples.collections.Grouping.countBy
 * @SinceKotlin("1.1")
 */
public fun <T, K, M : MutableMap<K, Int>> Grouping<T, K>.countByTo(destination: M): M =
    foldTo(destination, 0) { acc, _ -> acc + 1 }

/*
 /**
 * Groups elements from the [Grouping] source by key and sums values provided by the [valueSelector] function for elements in each group to the given [destination] map.
 * If the [destination] map already has a value corresponding to the key of some group, that value is used as an initial value of the sum for that group.
 * @return the [destination] map associating the key of each group with the sum of elements in the group.
 * @sample samples.collections.Grouping.sumByEach
 * @SinceKotlin("1.1")
 */
public fun <T, K, M : MutableMap<K, Int>> Grouping<T, K>.sumByEachTo(destination: M, valueSelector: (T) -> Int): M =
    foldTo(destination, 0) { acc, e -> acc + valueSelector(e) }

/*
 // TODO: sum by long and by double overloads

public inline fun <T, K, M : MutableMap<K, Long>> Grouping<T, K>.sumByEachByLongTo(destination: M, valueSelector: (T) -> Long): M =
    foldTo(destination, 0L) { acc, e -> acc + valueSelector(e) }

public inline fun <T, K> Grouping<T, K>.sumByEachByLong(valueSelector: (T) -> Long): Map<K, Long> =
    fold(0L) { acc, e -> acc + valueSelector(e) }

public inline fun <T, K, M : MutableMap<K, Double>> Grouping<T, K>.sumByEachByDoubleTo(destination: M, valueSelector: (T) -> Double): M =
    foldTo(destination, 0.0) { acc, e -> acc + valueSelector(e) }

public inline fun <T, K> Grouping<T, K>.sumByEachByDouble(valueSelector: (T) -> Double): Map<K, Double> =
    fold(0.0) { acc, e -> acc + valueSelector(e) }
*/

/*
 * Copyright 2010-2020
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

/**
 */
@Deprecated("Use top-level functions from kotlin.math package instead.", level = DeprecationLevel.ERROR)
public external object Math {
    @Deprecated("Use kotlin.math.PI instead.", ReplaceWith("PI", "kotlin.math.PI"), level = DeprecationLevel.ERROR)
    public val PI: Double

    @Deprecated("Use Random.nextDouble instead", ReplaceWith("kotlin.random.Random.nextDouble()", "kotlin.random")
    public fun random(): Double

    @Deprecated("Use kotlin.math.abs instead.", ReplaceWith("abs(value)", "kotlin.math.abs"), level = DeprecationLevel.ERROR)
    public fun abs(value: Double): Double

    @Deprecated("Use kotlin.math.acos instead.", ReplaceWith("acos(value)", "kotlin.math.acos"), level = DeprecationLevel.ERROR)
    public fun acos(value: Double): Double

    @Deprecated("Use kotlin.math.asin instead.", ReplaceWith("asin(value)", "kotlin.math.asin"), level = DeprecationLevel.ERROR)
    public fun asin(value: Double): Double

    @Deprecated("Use kotlin.math.atan instead.", ReplaceWith("atan(value)", "kotlin.math.atan"), level = DeprecationLevel.ERROR)
    public fun atan(value: Double): Double

    @Deprecated("Use kotlin.math.atan2 instead.", ReplaceWith("atan2(y, x)", "kotlin.math.atan2"), level = DeprecationLevel.ERROR)
    public fun atan2(y: Double, x: Double): Double

    @Deprecated("Use kotlin.math.cos instead.", ReplaceWith("cos(value)", "kotlin.math.cos"), level = DeprecationLevel.ERROR)
    public fun cos(value: Double): Double

    @Deprecated("Use kotlin.math.sin instead.", ReplaceWith("sin(value)", "kotlin.math.sin"), level = DeprecationLevel.ERROR)
    public fun sin(value: Double): Double

    @Deprecated("Use kotlin.math.exp instead.", ReplaceWith("exp(value)", "kotlin.math.exp"), level = DeprecationLevel.ERROR)
    public fun exp(value: Double): Double

    @Deprecated("Use maxOf or kotlin.math.max instead", level = DeprecationLevel.ERROR)
    public fun max(vararg values: Int): Int

    @Deprecated("Use maxOf or kotlin.math.max instead", level = DeprecationLevel.ERROR)
    public fun max(vararg values: Float): Float

    @Deprecated("Use maxOf or kotlin.math.max instead", level = DeprecationLevel.ERROR)
    public fun max(vararg values: Double): Double

    @Deprecated("Use minOf or kotlin.math.min instead", level = DeprecationLevel.ERROR)
    public fun min(vararg values: Int): Int

    @Deprecated("Use minOf or kotlin.math.min instead", level = DeprecationLevel.ERROR)
    public fun min(vararg values: Float): Float

    @Deprecated("Use minOf or kotlin.math.min instead", level = DeprecationLevel.ERROR)
    public fun min(vararg values: Double): Double

```kotlin
@Suppress("NOT_DOCUMENTED")
public external object JSON {
    public fun stringify(o: Any?): String
    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)?): String
    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)?): String
}
```

```kotlin
/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * LICENSE.txt file.
 */
@file:Suppress("DEPRECATION_ERROR")
package kotlin.math

import kotlin.internal.InlineOnly
import kotlin.js.Math as nativeMath

// region ================ Double Math

/** Computes the sine of the angle \([x]\) given in radians. *\n * Special cases:\n * - \(\sin(\text{NaN}+)\) is \(\text{NaN}\) *\n * - \(\sin(\text{NaN}-)\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sin(x: Double): Double = nativeMath.sin(x)

/** Computes the cosine of the angle \([x]\) given in radians. *\n * Special cases:\n * - \(\cos(\text{NaN}+)\) is \(\text{NaN}\) *\n * - \(\cos(\text{NaN}-)\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun cos(x: Double): Double = nativeMath.cos(x)

/** Computes the tangent of the angle \([x]\) given in radians. *\n * Special cases:\n * - \(\tan(\text{NaN}+)\) is \(\text{NaN}\) *\n * - \(\tan(\text{NaN}-)\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun tan(x: Double): Double = nativeMath.tan(x)

/** Computes the arc sine of the value \([x]\); the returned value is an *\n * angle in the range from \(-\pi/2\) to \(\pi/2\) radians. *\n * Special cases:\n * - \(\text{asin}(\text{NaN})\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun asin(x: Double): Double = nativeMath.asin(x)

/** Computes the arc cosine of the value \([x]\); the returned value is *\n * an angle in the range from \(0.0\) to \(\pi\) radians. *\n * Special cases:\n * - \(\text{acos}(\text{NaN})\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun acos(x: Double): Double = nativeMath.acos(x)

/** Computes the arc tangent of the value \([x]\); the returned value is *\n * an angle in the range from \(-\pi\) to \(\pi\) radians. *\n * Special cases:\n * - \(\text{atan}(\text{NaN})\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun atan(x: Double): Double = nativeMath.atan(x)

/** Computes the hyperbolic sine of the value \([x]\). *\n * Special cases:\n * - \(\sinh(\text{NaN})\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sinh(x: Double): Double = nativeMath.sinh(x)

/** Computes the hyperbolic cosine of the value \([x]\). *\n * Special cases:\n * - \(\cosh(\text{NaN})\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun cosh(x: Double): Double = nativeMath.cosh(x)

/** Computes the hyperbolic tangent of the value \([x]\). *\n * Special cases:\n * - \(\tanh(\text{NaN})\) is \(\text{NaN}\) *\n * @since 1.2 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun tanh(x: Double): Double = nativeMath.tanh(x)
```

Open Source Used In DNAC Platform Antelope 9529
value is positive 'y' such that \( \cosh(y) = x \). Special cases: if \( x = -\text{acosh}(\text{NaN}) \) is \( \text{NaN} \), when \( x < 1 \) or \( \text{acosh}(\text{+Inf}) \). The returned value is 'y' such that \( \tanh(y) = x \). Special cases: if \( x = \text{acosh}(\text{NaN}) \) is \( \text{NaN} \). When \( x > 1 \) or \( x < -1 \) and \( \text{tanh}(1.0) = \text{+Inf} \) and \( \text{tanh}(-1.0) = \text{-Inf} \), the other is not infinite. Special cases: if \( x = \text{acosh}(\text{NaN}) \) if any arguments is infinite. Special cases: if \( x = \text{acosh}(\text{NaN}) \) if any arguments is infinite. Special cases: if \( x = \text{acosh}(\text{NaN}) \) if any arguments is infinite.

```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun acosh(x: Double): Double = nativeMath.acosh(x)
```

** Computes the inverse hyperbolic tangent of the value \( x \).\n\n\* Special cases: if \( x = \text{acosh}(\text{NaN}) \) if any arguments is infinite. 4
\* Computes the inverse hyperbolic tangent of the value \( x \).\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun atanh(x: Double): Double = nativeMath.atanh(x)
```

- The returned value is 'y' such that \( \tanh(y) = x \). Special cases: if \( x = \text{atanh}(\text{NaN}) \) is \( \text{NaN} \). When \( x > 1 \) or \( x < -1 \), \( \text{atanh}(1.0) \). Special cases: if \( x = \text{atanh}(\text{NaN}) \) if any arguments is infinite.

```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun hypot(x: Double, y: Double): Double = nativeMath.hypot(x, y)
```

- Computes the positive square root of the value \( x \).\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sqrt(x: Double): Double = nativeMath.sqrt(x)
```

- Computes Euler's number \( e \) raised to the power of the value \( x \).\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun exp(x: Double): Double = nativeMath.exp(x)
```

- Computes \( e^x - 1 \) for values near zero.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun expm1(x: Double): Double = nativeMath.expm1(x)
```

- Computes the logarithm of the value \( x \) to the given base.\n\n```kotlin
public actual fun log(x: Double, base: Double): Double {
    if (base <= 0.0 || base == 1.0) return Double.NaN
    return nativeMath.log(x) / nativeMath.log(base)
}
```

- Computes \( \log_10(x) \) for values near zero.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log10(x: Double): Double = nativeMath.log10(x)
```

- Computes \( \log_2(x) \) for values near zero.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log2(x: Double): Double = nativeMath.log2(x)
```

- Computes \( \ln(1 + x) \) for values near zero.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun ln1p(x: Double): Double = nativeMath.log1p(x)
```

- Computes the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun ceil(x: Double): Double = nativeMath.ceil(x).unsafeCast<Double>() // TODO: Remove unsafe cast after removing public js.math\n```

- Computes the largest double value that is smaller than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun floor(x: Double): Double = nativeMath.floor(x).unsafeCast<Double>()
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun truncate(x: Double): Double = nativeMath.trunc(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun round(x: Double): Double = nativeMath.round(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun min(x: Double, y: Double): Double = nativeMath.min(x, y)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun max(x: Double, y: Double): Double = nativeMath.max(x, y)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun abs(x: Double): Double = nativeMath.abs(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sign(x: Double): Double = nativeMath.sign(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun expm1(x: Double): Double = nativeMath.expm1(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log1p(x: Double): Double = nativeMath.log1p(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sinh(x: Double): Double = nativeMath.sinh(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun cosh(x: Double): Double = nativeMath.cosh(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun tanh(x: Double): Double = nativeMath.tanh(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun acosh(x: Double): Double = nativeMath.acosh(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun atanh(x: Double): Double = nativeMath.atanh(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun hypot(x: Double, y: Double): Double = nativeMath.hypot(x, y)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sqrt(x: Double): Double = nativeMath.sqrt(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun exp(x: Double): Double = nativeMath.exp(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun expm1(x: Double): Double = nativeMath.expm1(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log1p(x: Double): Double = nativeMath.log1p(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log10(x: Double): Double = nativeMath.log10(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log2(x: Double): Double = nativeMath.log2(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log(x: Double): Double = nativeMath.log(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log10(x: Double): Double = nativeMath.log10(x)
```

- Returns the smallest double value that is greater than or equal to the given value \( x \) and is a mathematical integer.\n\n```kotlin
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun log2(x: Double): Double = nativeMath.log2(x)
```
the closest integer with ties rounded towards even integer.\n
*round(x)* is `x` where `x` is 'NaN' or '+Inf' or '-Inf' or already a mathematical integer.\n *round(x)* is `x` where `x` is 'NaN' or '+Inf' or '-Inf' or already a mathematici
Float Math

- Computes the sine of the angle [x] given in radians.
  \[ \sin(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the cosine of the angle [x] given in radians.
  \[ \cos(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the tangent of the angle [x] given in radians.
  \[ \tan(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the arc cosine of the value [x].
  \[ \arccos(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the arc tangent of the value [x].
  \[ \arctan(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the arc sine of the value [x].
  \[ \arcsin(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the hyperbolic sine of the value [x].
  \[ \sinh(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the hyperbolic cosine of the value [x].
  \[ \cosh(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the hyperbolic tangent of the value [x].
  \[ \tanh(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the inverse hyperbolic sine of the value [x].
  \[ \arcsinh(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the inverse hyperbolic cosine of the value [x].
  \[ \arccosh(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
- Computes the inverse hyperbolic tangent of the value [x].
  \[ \tanh^{-1}(x) = \frac{x}{\text{PI/2}} \times \frac{x}{\text{PI/2}} \]
`acosh(NaN)` is `NaN`
- `acosh(x)` is `NaN` when `x < 1`
- `acosh(+Inf)` is `+Inf`

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun acosh(x: Float): Float =
nativeMath.acosh(x.toDouble()).toFloat()

/**
 * Computes the inverse hyperbolic tangent of the value [x].
 *
 * Special cases:
 * - `tanh+y`) is `x`.
 * - `tanh(NaN)` is `NaN`
 * - `tanh(x)` is `NaN` when `x > 1` or `x < -1`
 * - `tanh(1.0)` is `+Inf`
 * - `tanh(-1.0)` is `-Inf`
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun atanh(x: Float): Float =
nativeMath.atanh(x.toDouble()).toFloat()

/**
 * Computes the logarithm of the value [x] to the given [base].
 *
 * Special cases:
 * - `ln(a + 1)` is `NaN`
 * - `ln(0.0)` is `-Inf`
 * - `ln(1)` is `0.0`
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun ln(x: Float): Float =
nativeMath.log(x.toDouble()).toFloat()
towards the closest integer with ties rounded towards even integer.

Special cases:
- `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.

    round(x.toDouble()).toFloat() Returns the absolute value of the given value [x].

Special cases:
- `abs(NaN)` is `NaN`.
- `round(-1.0)` if the value is negative.
- `round(0.0)` is `1.0`.
- `b.pow(0.0)` returns `NaN` for `b.pow(0.0)` is `0.0`.
- `b.pow(-1.0)` returns `NaN` for `b < 0`.

    nativeMath.pow(x.toDouble()).toFloat() Returns the value of this power [x].

Special cases:
- `b.pow(0.0)` is `1.0`.
- `b.pow(1.0)` is `b`.
- `b.pow(NaN)` is `NaN`.
- `b.pow(x)` is `NaN` for `x` is not an integer.

    nativeMath.pow(x.toDouble()).toFloat() Returns the absolute value of this value.

Special cases:
- `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow.
- `abs(NaN)` is `NaN`.

    nativeMath.abs(x.toDouble()).toFloat() Returns the absolute value of this power [x].

Special cases:
- `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow.
- `abs(NaN)` is `NaN`.

    withSign(sign: Float) Returns this value with the sign bit same as of the [sign] value.

If [sign] is `NaN` the sign of the result is undefined.

- `withSign(sign: Int)` Returns this value with the sign bit same as of the [sign] value.

Ties are rounded towards positive infinity.
- `x.roundToLong()` Returns this value with the sign bit same as of the [sign] value.

Ties are rounded towards positive infinity.
- `x.roundToLong()` is `Long.MAX_VALUE` when `x > Long.MAX_VALUE`.
- `x.roundToLong()` is `Long.MIN_VALUE` when `x < Long.MIN_VALUE`.

    nativeMath.min(a, b).toDouble().toFloat() Returns the greater of two values.

Special cases:
- `min(0.0)` is `0.0`.
- `min(-1.0)` if the value is negative.
- `min(1.0)` is `1.0`.

    nativeMath.min(a.toDouble(), b.toDouble()).toDouble() Returns the greater of two values.

Special cases:
- `min(0.0)` is `0.0`.
- `min(-1.0)` if the value is negative.
- `min(1.0)` is `1.0`.

    nativeMath.min(a.toDouble(), b.toDouble()).toDouble() Returns the greater of two values.

Special cases:
- `min(0.0)` is `0.0`.
- `min(-1.0)` if the value is negative.
- `min(1.0)` is `1.0`.
b)\n/*
 * Returns the absolute value of this value.
 */

@SinceKotlin("1.2")
@InlineOnly
public actual inline val Int.absoluteValue: Int get() = abs(this)

/**
 * Returns the sign of this value:
 *   - `-1` if the value is negative,
 *   - `0` if the value is zero,
 *   - `1` if the value is positive
 */
@SinceKotlin("1.2")
public actual val Int.sign: Int get() = when {
    this < 0 -> -1
    this > 0 -> 1
    else -> 0
}

/**
 * Returns the absolute value of the given value [n].
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline val Long.absoluteValue: Long get() = abs(this)

/**
 * Returns the sign of this value:
 *   - `-1` if the value is negative,
 *   - `0` if the value is zero,
 *   - `1` if the value is positive
 */
@SinceKotlin("1.2")
public actual val Long.sign: Int get() = when {
    this < 0 -> -1
    this > 0 -> 1
    else -> 0
}

/**
 * Counts the number of set bits in the binary representation of this [Int] number.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public actual fun Int.countOneBits(): Int {
    // Hacker's Delight 5-1 algorithm
    var v = this
    v = (v and 0x55555555) + (v.ushr(1) and 0x55555555)
    v = (v and 0x33333333) + (v.ushr(2) and 0x33333333)
    v = (v and 0x0F0F0F0F) + (v.ushr(4) and 0x0F0F0F0F)
    v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)
    return v
}

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public actual fun Int.countLeadingZeroBits(): Int =
    // Hacker's Delight 5-4 algorithm for expressing countTrailingZeroBits with countLeadingZeroBits
    Int.SIZE_BITS - (this or -this).inv().countLeadingZeroBits()

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Int] number.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public actual fun Int.countTrailingZeroBits(): Int =
    this or -this

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [Int] number,
 * or zero, if this number is zero.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public actual fun Int.takeHighestOneBit(): Int =
    if (this == 0) 0 else 1.shl(Int.SIZE_BITS - this)

Open Source Used In DNAC Platform Antelope 9535
*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.takeLowestOneBit(): Int =\n  // Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit this and - this\n  if (bitCount == 0) 0 else this.shr(bitCount) or this.ushr(-bitCount) * Rotating the number left by the specified [bitCount] number of bits.
  * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.
  * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count.
  * Rotating the number left by the specified [bitCount] number of bits.
  * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.
  * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count.
  * Rotating the number right by the specified [bitCount] number of bits.
  * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
  * number.rotateRight(-n) == number.rotateLeft(n) * Rotating by a multiple of [Int.SIZEBITS] (64) returns the same number, or more generally*
negative bit count is the same as rotating it left by the negated bit count:
\[ \text{number.rotateRight}(-n) == \text{number.rotateLeft}(n) \]
Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally:
\[ \text{number.rotateRight}(n) == \text{number.rotateRight}(n \% 64) \]

```kotlin
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public actual inline fun Long.rotateRight(bitCount: Int): Long = rotateLeft(-bitCount)
```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.internal

@package: Antelope

val @SinceKotlin("1.3")@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
val @package: Antelope

val @SinceKotlin("1.3")@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
val @package: Antelope

```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

import kotlin.internal.LowPriorityInOverloadResolution

/**
 * Exposes the JavaScript [Promise object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Promise) to Kotlin.
 */
@Suppress("NOT_DOCUMENTED")
public open external class Promise<out T>(executor: (resolve: (T) -> Unit, reject: (Throwable) -> Unit) -> Unit) {
    @LowPriorityInOverloadResolution
    public open fun <S>
        then(onFulfilled: ((T) -> S)?): Promise<S>

    @LowPriorityInOverloadResolution
    public open fun <S>
        then(onFulfilled: ((T) -> S)?, onRejected: ((Throwable) -> S)?): Promise<S>

    public open fun <S>
        catch(onRejected: (Throwable) -> S?): Promise<S>

    companion object {
        public fun <S> all(promise: Array<out Promise<S>>): Promise<Array<out S>>

        public fun <S> race(promise: Array<out Promise<S>>): Promise<S>

        public fun reject(e: Throwable): Promise<Nothing>

        public fun <S> resolve(e: S): Promise<S>
        public fun <S> resolve(e: Promise<S>): Promise<S>
    }
}

// It's workaround for KT-19672 since we can fix it properly until KT-11265 isn't fixed.
inline fun <T, S> Promise<Promise<T>>.then(
    noinline onFulfilled: ((T) -> S)?
): Promise<S> {
    return this.unsafeCast<Promise<T>>().then(onFulfilled)
}

inline fun <T, S> Promise<Promise<T>>.then(
    noinline onFulfilled: ((T) -> S)?,
    noinline onRejected: ((Throwable) -> S)?
): Promise<S> {
    return this.unsafeCast<Promise<T>>().then(onFulfilled, onRejected)
}

```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.random

import kotlin.math.pow

internal actual fun defaultPlatformRandom(): Random =
    Random(js("(Math.random() * Math.pow(2, 32)) | 0").unsafeCast<Int>())

private val INV_2_26: Double = 2.0.pow(-26)
private val INV_2_53: Double = 2.0.pow(-53)
internal actual fun doubleFromParts(hi26: Int, low27: Int): Double =
    hi26 * INV_2_26 + low27 * INV_2_53

```

```
/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.reflect

import findAssociatedObject

/**
 * The experimental marker for associated objects API.
 */
@RequiresOptIn(level = RequiresOptIn.Level.ERROR)
@Retention(value = AnnotationRetention.BINARY)
public annotation class ExperimentalAssociatedObjects

/**
 * Makes the annotated annotation class an associated object key.
 */
@ExperimentalAssociatedObjects
@Retention(AnnotationRetention.BINARY)
@Target(AnnotationTarget.ANNOTATION_CLASS)
public annotation class AssociatedObjectKey

/**
 * If T is an @AssociatedObjectKey-annotated annotation class and S::class is annotated wit
 * Any usage of a declaration annotated with
 * `@ExperimentalAssociatedObjects` must be accepted either by `null` * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalAssociatedObjects::class)`, `null` * or by using the compiler argument `-Xopt-in=kotlin.reflect.ExperimentalAssociatedObjects`.
 */
@ExperimentalAssociatedObjects
@Retention(AnnotationRetention.BINARY)
public inline fun <reified T : Annotation> KClass<*>.findAssociatedObject(): Any? * Makes the annotated annotation class an associated object key.

```

```
/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

import getKClass
import kotlin.reflect.KClass
import kotlin.reflect.js.internal.KClassImpl

/**
 * Represents the constructor of a class. Instances of `JsClass` can be passed to JavaScript APIs that expect a constructor reference.
 */
@Target(ElementType.TYPE)
@Retention(AnnotationRetention.RUNTIME)
public annotation class JsClass

```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package com.google.solutions.dnac.platform.antelope

import com.google.solutions.dnac.platform.antelope.OpenSource

```

```
Open Source Used In DNAC Platform Antelope 9537
val name: String

/**
 * Obtains a constructor reference for the given `KClass`.
 */
val <T : Any> KClass<T>.js: JsClass<T>
    get() = (this as KClassImpl<T>).jClass

/**
 * Obtains a `KClass` instance for the given constructor reference.
 */
val <T : Any> KClass<T>.kotlin: KClass<T>
    get() = getKClass(this)

package kotlin.reflect.js

internal class KClassImpl<T : Any>(
    internal open val jClass: JsClass<T>
) : KClass<T> {

    override val qualifiedName: String?
        get() = TODO()

    override fun equals(other: Any?): Boolean {
        return other
    }

    override fun hashCode(): Int = simpleName?.hashCode() ?: 0

    override fun toString(): String {
        return "class $simpleName"
    }
}

internal class SimpleKClassImpl<T : Any>(jClass: JsClass<T>) :
    KClassImpl<T>(jClass) {
    override val simpleName: String? = jClass.asDynamic().`$metadata$`?.simpleName.unsafeCast<String?>()

    override fun isInstance(value: Any?): Boolean {
        return jsIsType(value, jClass)
    }
}

internal class PrimitiveKClassImpl<T : Any>(
    jClass: JsClass<T>,
    private val givenSimpleName: String,
    private val isInstanceFunction: (Any?) -> Boolean
) : KClassImpl<T>(jClass) {
    override fun equals(other: Any?): Boolean {
        if (other !is PrimitiveKClassImpl<*>) return false
        return givenSimpleName == other.givenSimpleName
    }

    override val simpleName: String? = givenSimpleName

    override fun isInstance(value: Any?): Boolean {
        return isInstanceFunction(value)
    }
}

internal object NothingKClassImpl : KClassImpl<Nothing>(js("Object")) {
    override val simpleName: String = "Nothing"

    override fun isInstance(value: Any?): Boolean = false

    override val jClass:
        JsClass<Nothing>
        get() = throw UnsupportedOperationException("There's no native JS class for Nothing type")

    override fun equals(other: Any?): Boolean = other === this

    override fun hashCode(): Int = 0
}

internal class ErrorKClass : KClass<Nothing> {
    override val simpleName: String? = error("Unknown simpleName for ErrorKClass")
    override val qualifiedName: String? = error("Unknown qualifiedName for ErrorKClass")

    override fun isInstance(value: Any?): Boolean = error("Can't check isInstance on ErrorKClass")

    override fun equals(other: Any?): Boolean = other === this

    override fun hashCode(): Int = 0
}
package kotlin.reflect.js.internal

import kotlin.reflect.*

internal class KTypeImpl(
    override val classifier: KClassifier,
    override val arguments: List<KTypeProjection>,
    override val isMarkedNullable: Boolean
) : KType {
    override fun equals(other: Any?): Boolean =
        other is KTypeImpl &&
        classifier == other.classifier && arguments == other.arguments &&
        isMarkedNullable == other.isMarkedNullable

    override fun hashCode(): Int =
        (classifier.hashCode() * 31 + arguments.hashCode()) * 31 + isMarkedNullable.hashCode()

    override fun toString(): String {
        val kClass = (classifier as? KClass<*>)
        val classifierName = when {
            kClass == null ->
                classifier.toString()
            kClass.simpleName != null -> kClass.simpleName
            else -> "(non-denotable type)"
        }

        val args =
            if (arguments.isEmpty()) ""
            else arguments.joinToString(", ", "<", ">") { it.asString() }
        val nullable = if (isMarkedNullable) "?" else ""

        return classifierName
            + args + nullable
    }

    // TODO: this should be the implementation of KTypeProjection.toString, see KT-30071
    private fun KTypeProjection.asString(): String {
        if (variance == null) return "*"
        return
            variance.prefixString() + type.toString()
    }
}

internal fun KVariance.prefixString() =
    when (this) {
        KVariance.INVARIANT -> ""
        KVariance.IN -> "in "
        KVariance.OUT -> "out "
    }

package kotlin.reflect.js.internal

import kotlin.js.JsClass

@JsName("PrimitiveClasses")
internal object PrimitiveClasses {
    @JsName("anyClass")
    val anyClass =
        PrimitiveKClassImpl(js("Object").unsafeCast<JsClass<Any>>(), "Any", { it is Any })

    @JsName("numberClass")
    val numberClass =
        PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Number>>(), "Number", { it is Number })

    @JsName("nothingClass")
    val nothingClass = NothingKClassImpl

    @JsName("booleanClass")
    val booleanClass =
        PrimitiveKClassImpl(js("Boolean").unsafeCast<JsClass<Boolean>>(), "Boolean", { it is Boolean })

    @JsName("byteClass")
    val byteClass =
        PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Byte>>(), "Byte", { it is Byte })

    @JsName("shortClass")
    val shortClass =
        PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Short>>(), "Short", { it is Short })

    @JsName("intClass")
    val intClass =
        PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Int>>(), "Int", { it is Int })

    @JsName("floatClass")
    val floatClass =
        PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Float>>(), "Float", { it is Float })

    @JsName("doubleClass")
    val doubleClass =
        PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Double>>(), "Double", { it is Double })

    @JsName("arrayClass")
    val arrayClass =
        PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<*>>(), "Array", { it is Array })

    @JsName("stringClass")
    val stringClass =
        PrimitiveKClassImpl(js("String").unsafeCast<JsClass<String>>(), "String", { it is String })

    @JsName("throwableClass")
    val throwableClass =
        PrimitiveKClassImpl(js("Throwables").unsafeCast<JsClass<Throwables>>(), "Throwables", { it is Throwables })
}
primitiveKClassImpl(js("Error").unsafeCast<JsClass<Throwable>>(), "Throwable", { it is Throwable })

@JsName("booleanArrayClass")
    val booleanArrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<BooleanArray>>(), "BooleanArray", { it is BooleanArray })

    @JsName("charArrayClass")
    val charArrayClass = PrimitiveKClassImpl(js("Int16Array").unsafeCast<JsClass<CharArray>>(), "CharArray", { it is CharArray })

    @JsName("byteArrayClass")
    val byteArrayClass = PrimitiveKClassImpl(js("Int8Array").unsafeCast<JsClass<ByteArray>>(), "ByteArray", { it is ByteArray })

    @JsName("shortArrayClass")
    val shortArrayClass = PrimitiveKClassImpl(js("Int16Array").unsafeCast<JsClass<ShortArray>>(), "ShortArray", { it is ShortArray })

    @JsName("intArrayClass")
    val intArrayClass = PrimitiveKClassImpl(js("Int32Array").unsafeCast<JsClass<IntegerArray>>(), "IntArray", { it is IntArray })

    @JsName("longArrayClass")
    val longArrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<LongArray>>(), "LongArray", { it is LongArray })

    @JsName("floatArrayClass")
    val floatArrayClass = PrimitiveKClassImpl(js("Float32Array").unsafeCast<JsClass<FloatArray>>(), "FloatArray", { it is FloatArray })

    @JsName("doubleArrayClass")
    val doubleArrayClass = PrimitiveKClassImpl(js("Float64Array").unsafeCast<JsClass<DoubleArray>>(), "DoubleArray", { it is DoubleArray })

    @JsName("functionClass")
    fun functionClass(arity: Int): KClassImpl<Any> {
        return functionClasses.get(arity) ?: run {
            val result = PrimitiveKClassImpl(js("Function").unsafeCast<JsClass<Any>>(), "Function$arity", { jsTypeOf(it) === "function" && it.asDynamic().length === arity })
            functionClasses.asDynamic()[arity] = result
            result
        }
    }

private val functionClasses = arrayOfNulls<KClassImpl<Any>>(0)

@JsName("getKClass")
internal fun <T : Any> getKClass(jClass: Any /* JsClass<T> | Array<JsClass<T>> */): KClass<T> {
    return if (js("Array").isArray(jClass)) {
        getKClassM(jClass.unsafeCast<Array<JsClass<T>>>())
    } else {
        getKClass1(jClass.unsafeCast<JsClass<T>>() as KClass<T>)
    }
}

@JsName("getKClassM")
internal fun <T : Any> getKClassM(jClasses: Array<JsClass<T>>): KClass<T> = when (jClasses.size) {
    1 -> getKClass1(jClasses[0])
    0 -> NothingKClassImpl.unsafeCast<KClass<T>>()
    else -> ErrorKClass().unsafeCast<KClass<T>>()
}

@JsName("getKClassFromExpression")
internal fun <T : Any> getKClassFromExpression(e: T): KClass<T> =
    when (jsTypeOf(e)) {
        "string" -> PrimitiveClasses.stringClass
        "number" -> if (jsBitwiseOr(e, 0).asDynamic() === e)
            PrimitiveClasses.intClass
        else PrimitiveClasses.doubleClass
        "boolean" -> PrimitiveClasses.booleanClass
        "function" -> PrimitiveClasses.functionClass(e.asDynamic().length)
        else -> KClass::class
    }

    .unsafeCast<KClass<T>>()
}

constructor

val constructor = js("Object").getPrototypeOf(e).constructor
when { e when { e is BooleanArray -> PrimitiveClasses.booleanClass
    e is CharArray -> PrimitiveClasses.charArrayClass
    e is ByteArray -> PrimitiveClasses.byteArrayClass
    e is LongArray -> PrimitiveClasses.longArrayClass
    e is FloatArray -> PrimitiveClasses.floatArrayClass
    e is DoubleArray -> PrimitiveClasses.doubleArrayClass
    e is KClass<*> -> KClass.class
    e is Array<*> -> PrimitiveClasses.arrayClass
    else -> { val constructor = js("Object").getPrototypeOf(e).constructor
        when { e === null constructor === js("Error") -> PrimitiveClasses.throwableClass
            else -> { val jsClass: JsClass<T> = constructor
                getKClass1(jsClass) }
        } unsafeCast<KClass<T>>()
    }
    @JsName("getString")()} unsafeCast<KClass<T>>()
PrimitiveClasses.stringClass.unsafeCast<KClass<T>>()
val metadata = jClass.asDynamic().`$metadata$`
return if (metadata != null) {
    if (metadata.`$kClass$` == null) {
        val kClass =
SimpleKClassImpl(jClass)
        metadata.`$kClass$` = kClass
    } else {
        SimpleKClassImpl(jClass)
    }
} else {
    null
}

return if (metadata != null) {
    if (metadata.`$kClass$` == null) {
        val kClass =
SimpleKClassImpl(jClass)
        metadata.`$kClass$` = kClass
        kClass
    } else {
        metadata.`$kClass$`
    }
} else {
    SimpleKClassImpl(jClass)
}
}

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

/**
 * Exposes
 * the JavaScript 
RegExp object
 * to Kotlin.
 */
@Suppress("NOT_DOCUMENTED")
public external class RegExp(pattern: String, flags: String? = definedExternally) {

    public fun test(str: String): Boolean

    public fun exec(str: String): RegExpMatch?

    public override fun toString(): String

    /**
     * The lastIndex is a read/write integer property of regular
     * expressions that specifies the index at which to start the next match.
     */
    public var lastIndex: Int

    public
    val global: Boolean
    val ignoreCase: Boolean
    val multiline: Boolean
}

/**
 * Resets the
 * regular expression so that subsequent
 * [RegExp.test] and
 * [RegExp.exec] calls will match starting with the beginning of
 * the input string.
 */
public fun RegExp.reset() {
    lastIndex = 0
}

// TODO: Inherit from array or
introduce asArray() extension
*/*

public external interface RegExpMatch {
    public val index: Int
    public val input: String
    public val length: Int
}

/**
 * Returns the entire text matched by
 * [RegExp.exec] if
 * the [index] parameter is 0, or the text matched by each capturing parenthesis
 * at the given index.
 */
public inline

/**
 * Converts the result of
 * [RegExp.exec] to an array where the first element contains the entire matched text and each subsequent
 * element
 * is the text matched by each capturing parenthesis.
 */
public inline fun RegExpMatch.asArray(): Array<out
String?> = unsafeCast<Array<out String?>>()

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.sequences

internal actual class ConstrainedOnceSequence<T> actual constructor(sequence: Sequence<T>) : Sequence<T> {
    private var sequenceRef: Sequence<T>? = sequence

    actual override fun iterator(): Iterator<T> {
        val sequence =
        sequenceRef ?: throw IllegalStateException("This sequence can be consumed only once.")
        sequenceRef =
        null
        return sequence.iterator()
    }
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.text

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform
 * multiple string manipulation operations.
 * It contains the same characters as the specified
 * [content] char sequence.
 */
public actual class StringBuilder actual constructor(content: String) :
Appendable, CharSequence {
    private var sequenceRef: Sequence<String> =
    SequenceRef?
    = sequence
    actual override fun iterator(): Iterator<String> {
        return sequence.iterator()
    }

    actual override fun append(value: Char): StringBuilder {
        string += value
        return this
    }

    actual override fun get(index: Int): Char {
        string.get(index)
    }

    actual override fun toString(): String {
        return string
    }
}

/**
 * Character coding exception.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public actual open class CharacterCodingException(message: String?) : Exception(message) {
    actual constructor() : this(null)
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

Open Source Used In DNAC Platform Antelope 9541
override fun append(value: CharSequence?): StringBuilder {
    string += value.toString()
    return this
}
the entire supplementary character.

*param index the index of `Char` to remove.

* @throws
IndexOutOfBoundsException if [index] is out of bounds of this string builder.

* @throws
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)

public fun deleteAt(index: Int): StringBuilder {
    AbstractList.checkElementIndex(index, length)
    string = string.substring(0, index) + string.substring(index + 1)
    return this
}

/**
   * Removes characters in the specified range from this string builder and
   * returns this instance.

   * @param startindex the beginning (inclusive) of the range to remove.

   * @param endindex the end (exclusive) of the range to remove.

   * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startindex] is not out of range of this string builder indices or when `startIndex > endIndex`.

   * @since Kotlin("1.4")

   * @sinceExperimental(ExperimentalStdlibApi::class)

   public fun deleteRange(startIndex: Int, endIndex: Int): StringBuilder {
        checkReplaceRange(startIndex, endIndex, length)

        string = string.substring(0, startIndex) + string.substring(endIndex)
        return this
    }

    /**
     * Copies characters from this string builder into the [destination] character array.

     * @param destination the array to copy to, 0 by default.

     * @param destinationOffset the position in the array to copy to, 0 by default.

     * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.

     * @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.

     * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.

     * @since Kotlin("1.4")

     * @sinceExperimental(ExperimentalStdlibApi::class)

     public fun toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) {
            AbstractList.checkBoundsIndexes(startIndex, endIndex, length)

            var dstIndex = destinationOffset
            for (index in startIndex until endIndex) {
                destination[dstIndex++] = string[index]
            }
        }

    /**
     * Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance.

     * Characters are appended in order, starting at specified [startIndex].

     * @param value the array from which characters are appended.

     * @param startIndex the beginning (inclusive) of the subarray to append.

     * @param endIndex the end (exclusive) of the subarray to append.

     * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.

     * @since Kotlin("1.4")

     * @sinceExperimental(ExperimentalStdlibApi::class)

     public fun appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {
            @OptIn(ExperimentalStdlibApi::class)
            string += value.concatToString(startIndex, endIndex)
            return this
        }

    /**
     * Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance.

     * @param value the character sequence from which a subsequence is appended.

     * @param startIndex the beginning (inclusive) of the subsequence to append.

     * @param endIndex the end (exclusive) of the subsequence to append.

     * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.

     * @since Kotlin("1.4")

     * @sinceExperimental(ExperimentalStdlibApi::class)

     public fun appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder {
            val stringCsq = value.toString()

            AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)

            string += stringCsq.substring(startIndex, endIndex)
            return this
        }

    /**
     * Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance.

     * The inserted characters go in same order as in the [value] array, starting at [index].

     * @param index the position in this string builder to insert at.

     * @param value the array from which characters are inserted.

     * @param startIndex the beginning (inclusive) of the subarray to insert.

     * @param endIndex the end (exclusive) of the subarray to insert.

     * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.

     */

}
insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {
    AbstractList.checkPositionIndex(index, this.length)

    @OptIn(ExperimentalStdlibApi::class)
    string =
    string.substring(0, index) + value.concatToString(startIndex, endIndex) + string.substring(index)
    return this
}

inStringCsq = value.toString()
AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)

string = string.substring(0, index) + stringCsq.substring(startIndex, endIndex) + string.substring(index)
return this
}

/**
 * Clears the content of this string builder making it empty and returns this instance.
 *
 * @sample samples.text.Strings.clearStringBuilder
 */
@SinceKotlin("1.3")
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.clear(): StringBuilder = this.clear()

/**
 * Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance.
 *
 * @param startIndex the beginning (inclusive) of the range to replace.
 * @param endIndex the end (exclusive) of the range to replace.
 * @param value the string to replace with.
 *
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder or when `startIndex > endIndex`.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder =
    this.setRange(startIndex, endIndex, value)

/**
 * Removes the character at the specified [index] from this string builder and returns this instance.
 *
 * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.
 *
 * @param index the index of `Char` to remove.
 *
 * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun deleteAt(index: Int): StringBuilder = this.deleteAt(index)

/**
 * Removes characters in the specified range from this string builder and returns this instance.
 *
 * @param startIndex the beginning (inclusive) of the range to remove.
 * @param endIndex the end (exclusive) of the range to remove.
 *
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun deleteRange(startIndex: Int, endIndex: Int): StringBuilder =
    this.deleteRange(startIndex, endIndex)
StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder = this.deleteRange(startIndex, endIndex)

StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) =
    this.toCharArray(destination, destinationOffset, startIndex, endIndex)

StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =
this.appendRange(value, startIndex, endIndex)

StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =
this.appendRange(value, startIndex, endIndex)

StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =
this.insertRange(index, value, startIndex, endIndex)
`StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =
  this.insertRange(index, value, startIndex, endIndex)

public actual fun Char.isWhitespace(): Boolean = toString().matches("[\s\xA0]")

@kotlin.internal.InlineOnly
public actual inline fun Char.toLowerCase(): Char =
  js("String.fromCharCode")((toInt()).toLowerCase().charCodeAt(0).unsafeCast<Int>().toChar())

@kotlin.internal.InlineOnly
public actual inline fun Char.toUpperCase(): Char =
  js("String.fromCharCode")((toInt()).toUpperCase().charCodeAt(0).unsafeCast<Int>().toChar())

/**
 * Returns `true` if the content of this string is equal to the word "true", ignoring case, and `false` otherwise.
 */
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@DeprecatedSinceKotlin(hiddenSince = "1.4")
public actual inline fun String.toBoolean(): Boolean =
  this.toString().matches("\btrue\b")

/**
 * Parses the string as a signed `Byte` number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
public actual fun String.toByte(): Byte = toByteOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a signed `Byte` number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
public actual fun String.toByte(radix: Int): Byte = toByteOrNull(radix) ?: numberFormatError(this)

/**
 * Parses the string as an `Int` number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
public actual fun String.toInt(): Int = toIntOrNull() ?: numberFormatError(this)

/**
 * Parses the string as an `Int` number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
public actual fun String.toInt(radix: Int): Int = toIntOrNull(radix) ?: numberFormatError(this)
number

/*
 * Public actual function String.toLong(): Long = toLongOrNull() ?: numberFormatException(this)
 *
 * Parses the string as a [Long] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toLong(): Long = toLongOrNull() ?: numberFormatException(this)

/**
 * Parses the string as a [Long] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toLong(radix: Int): Long = toLongOrNull(radix) ?:

/**
 * Parses the string as a [Double] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toDouble(): Double = (+(this.asDynamic())).unsafeCast<Double>().also {
    if (it.isNaN() && !this.isNaN() ||
        it == 0.0 && this.isNotBlank())
        numberFormatException(this)
}

/**
 * Parses the string as a [Float] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
@kotlin.internal.InlineOnly
public actual inline fun String.toFloat(): Float =

toDouble().unsafeCast<Float>().also {
    if (it.isNaN() && !this.isNaN() || it == 0.0 &&
        this.isNotBlank())
        numberFormatException(this)
}

/**
 * Parses the string as a [Double] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toDoubleOrNull(): Double? =

/**
 * Parses the string as a [Float] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
@kotlin.internal.InlineOnly
public actual inline fun String.toFloatOrNull(): Float? = toDoubleOrNull().unsafeCast<Float?>()

/**
 * Returns a string representation of this [Byte] value in the specified [radix].
 */
@PublishedApi
internal actual fun Byte.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Short] value in the specified [radix].
 */
@PublishedApi
internal actual fun Short.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Int] value in the specified [radix].
 */
public actual fun Int.toString(radix: Int): String =

private fun String.isNaN(): Boolean = when (this.toLowerCase()) {
    "nan", "+nan", "-nan" -> true
    else -> false
}

/**
 * Checks whether the given [radix] is valid radix for number to string conversion.
 */
@PublishedApi
internal actual fun checkRadix(radix: Int): Int {
    if (radix !in 2..36) {
        throw IllegalArgumentException("radix $radix was not in valid range 2..36")
    }
    return radix
}

internal actual fun digitOf(char: Char, radix: Int): Int = when {
    char >= '0' && char <= '9' -> char - '0'
    char >= 'A' && char <= 'Z' -> char - 'A' + 10
    char >= 'a' && char <= 'z' -> char - 'a' + 10
    else -> -1
}.let { if (it >= radix) -1 else it }

package kotlin.text

import kotlin.js.RegExp

/**
 * Provides enumeration values to use to set regular expression options.
 */
public actual enum class RegexOption(val value: String) {
    /** Enables case-insensitive matching. */
    IGNORE_CASE("i"),
    /** Enables multiline mode.
     *
     * In multiline mode the expressions `^` and `$` match just after or just before,
     * respectively, a line terminator or the end of the input sequence.
     */
    MULTILINE("m")
}

/**
 * Represents the results from a single capturing group within a [MatchResult] of [Regex].
 *
 * @param value The value of captured group.
 */
public actual data class MatchGroup(actual val value: String)

/**
 * Represents a compiled regular expression.
 * Provides functions to match strings in text with a pattern, replace the found occurrences and
 * split text around matches.
 *
 * For pattern syntax reference see [MDN RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp#Special_characters_meaning_in_regular_expressions)
 * and
 * [http://www.w3schools.com/jsref/jsref_obj_regex.asp](https://www.w3schools.com/jsref/jsref_obj_regex.asp).
 */
@PublishedApi
public actual class Regex actual constructor(pattern: String, options: Set<RegexOption>) {

    /** Creates a regular expression from the specified pattern and the specified set of [options].
     * @param options The value of captured group.
     */
    @constructor Creates a regular expression from the specified [pattern] string and the specified set of options.
    * * */
    public actual class Regex actual constructor(pattern: String, options: Set< RegexOption >): String

    /** Creates a regular expression from the specified [pattern] string and the specified single [option].
     * * */
    public actual constructor(pattern: String, option: RegexOption) : this(pattern, setOf(option))

    /** Enables case-insensitive matching. */
    IGNORE_CASE("i"),
    /** Enables multiline mode. */
    MULTILINE("m")

    fun * */
from the specified [pattern] string and the default options. */
    public actual constructor(pattern: String) :
        this(pattern, emptySet()) /* The pattern string of this regular expression. */
        public actual val pattern:
            String = pattern
        /* The set of options that were used to create this regular expression. */
        public actual val options: Set<RegexOption> = options.toSet()
    }

    /**
     * Indicates whether the regular expression matches the entire [input]. */
    public actual infix fun matches(input: CharSequence): Boolean {
        nativePattern.reset()
        val match = nativePattern.exec(input.toString())
        return match != null && match.index == 0 &&
                nativePattern.lastIndex == input.length
    }

    /** Indicates whether the regular expression can find at least one
     * match in the specified [input]. */
    public actual fun containsMatchIn(input: CharSequence): Boolean {
        nativePattern.reset()
        return nativePattern.test(input.toString())
    }

    /** Returns the first match of a regular expression in the [input],
     * beginning at the specified [startIndex]. */
    public actual fun find(input: CharSequence, startIndex: Int = 0): MatchResult? {
        if (startIndex < 0 || startIndex > input.length) {
            throw IndexOutOfBoundsException("Start index out of bounds: $startIndex, input length: ${input.length}")
        }
        return nativePattern.findNext(input.toString(), startIndex)
    }

    /** Returns a sequence of all occurrences of a regular expression
     * within the [input] string, beginning at the specified [startIndex].
     */
    public actual fun findAll(input: CharSequence, startIndex: Int = 0): Sequence<MatchResult> {
        if (startIndex < 0 || startIndex > input.length) {
            throw IndexOutOfBoundsException("Start index out of bounds: $startIndex, input length: ${input.length}")
        }
        return generateSequence({ find(input, startIndex) }, { match -> match.next() })
    }

    /** Attempts to match the entire [input] CharSequence against the pattern.
     * */
    public actual fun matchEntire(input: CharSequence): MatchResult? {
        if (pattern.startsWith('^') && pattern.endsWith('$'))
            return find(input)
        else
            return Regex("^${pattern.trimStart('^').trimEnd('$')}$", options).find(input)
    }

    /** Replaces all occurrences of this regular expression in the specified [input]
     * with specified [replacement] expression. */
    public actual fun replace(input: CharSequence, replacement: String): String = input.toString().nativeReplace(nativePattern, replacement)

    /** Replaces all occurrences of this regular expression in the specified [input] string with
     * the result of [transform] that takes [MatchResult] and returns a string to be used as a
     * replacement for that match. */
    public actual fun replace(input: CharSequence, transform: (MatchResult) -> CharSequence): String {
        var match = find(input)
        if (match == null) return input.toString()
        val lastStart = 0
        val length = input.length
        val sb = StringBuilder(length)
        do {
            if (match == null) return sb.toString()
            sb.append(transform(match))
            lastStart = match.range.start
            match = find(input)
        } while (lastStart < length && match != null)
        sb.append(escapeWithBackslash(input, lastStart, foundMatch.range.endInclusive))
        return sb.toString()
    }

    /** Replaces the first occurrence of this regular expression in the specified [input] string
     * with specified [replacement] expression. */
    public actual fun replaceFirst(input: CharSequence, replacement: String): String {
        val nonGlobalOptions =
options.map { it.value }.joinToString(separator = "\"\")
    return input.toString().nativeReplace(Regexp(pattern, nonGlobalOptions), replacement)  
  }  
}  
/**  
 * Splits the [input] CharSequence around matches of this  
 * regular expression.  
 * @param limit Non-negative value specifying the maximum number of substrings  
 * the string can be split to.  
 * @param limit Limit must be non-negative, but was \"limit\"  
 * val matches = findAll(input).let { if (limit == 0) it else it.take(limit - 1) }  
 * var result =  
 * mutableListOf<String>()  
 * var lastStart = 0  
 * for (match in matches) {  
 *    result.add(input.substring(lastStart, match.range.start).toString())  
 *    lastStart = match.range.endInclusive + 1  
 * }  
 * result.add(input.substring(lastStart, input.length).toString())
 * return result
 */

/**  
 * Returns the string representation of this regular expression, namely the [pattern] of this regular  
 * expression.  
 */
/**  
 * Returns a regular expression pattern string that matches the specified [literal] string literally.  
 * No characters of that string will have special meaning when searching for an occurrence of the regular expression.  
 */
public override fun toString(): String = nativePattern.toString()

actual companion object {
    /**  
     * Returns a regular expression that  
     * matches the specified [literal] string literally.  
     * No characters of that string will have special meaning when it is used as a  
     * replacement string in [Regex.replace] function.  
     */
    public actual fun fromLiteral(literal: String): Regex = Regex(escape(literal))

    /**  
     * Returns a literal replacement expression for the  
     * specified [literal] string.  
     * No characters of that string will have special meaning when it is used as a  
     * replacement string.  
     */
    public actual fun escapeReplacement(literal: String): String = literal.nativeReplace(replacementEscape, "$$")

    private val patternEscape = RegExp("""[\-\\^$*+?.()|\[\]{}]\""", "g")
    private val replacementEscape = RegExp("""\$""", "g")
}

/**
 * Creates a regular expression from the specified [pattern] string and the specified single [option].
 */
@Deprecated("Provided for binary compatibility", level = DeprecationLevel.HIDDEN)
@JsName("Regex\$3q2\$")
public fun Regex_0(pattern: String, option: RegexOption): Regex = Regex(pattern, setOf(option))

/** Creates a regular expression from the specified [pattern] string and the default options.  
 */
@Deprecated("Provided for binary compatibility", level = DeprecationLevel.HIDDEN)
@JsName("Regex_61zpoe\$")
public fun Regex_1(pattern: String): Regex = Regex(pattern, emptySet())
newChar.toString()*/n","*/n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.
 */n */npackage kotlin.text
*/n/**
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */n
package kotlin.text

/** Returns the negative [size] if [throwOnMalformed] is false, throws [CharacterCodingException] otherwise.
 * @throws CharacterCodingException if the sequence is malformed and [throwOnMalformed] is true.
 */
@OptIn(ExperimentalStdlibApi::class)
private fun malformed(size: Int, index: Int, throwOnMalformed: Boolean): Int {
    if (throwOnMalformed) throw CharacterCodingException("Malformed sequence starting at $index")
    return -size
}

/**
 * Returns code point corresponding to UTF-8 sequence of two bytes, where the first byte of the sequence is the
 * [byte1] and the second byte is in the [bytes] array at the [index].
 * * Returns zero if the sequence is malformed and [throwOnMalformed] is false.
 * @throws CharacterCodingException if the sequence is malformed and [throwOnMalformed] is true.
 */
private fun codePointFrom2(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (byte1 and 0x1E == 0 || index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }
    val byte2 = bytes[index].toInt()
    if (byte2 and 0xC0 != 0x80) {
        return malformed(0, index, throwOnMalformed)
    }
    return (byte1 shl 6) xor byte2 xor -0x1E080
}

/**
 * Returns code point corresponding to UTF-8 sequence of three bytes, where the first byte of the sequence is the
 * [byte1] and the others are in the [bytes] array starting from the [index].
 * * Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence
 * if the sequence is malformed and [throwOnMalformed] is false.
 * @throws CharacterCodingException if the sequence is malformed and [throwOnMalformed] is true.
 */
private fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }

    val byte2 = bytes[index].toInt()
    if (byte1 and 0xF == 0) {
        if (byte2 and 0xF0 <= 0x80) {
            return malformed(0, index, throwOnMalformed)
        }
    } else if (byte1 and 0xF == 0x4) {
        if (byte2 and 0xF0 != 0x80) {
            return malformed(0, index, throwOnMalformed)
        }
    } else if (byte1 and 0xF > 0x4) {
        return malformed(0, index, throwOnMalformed)
    }

    if (index + 1 == endIndex) {
        return malformed(1, index, throwOnMalformed)
    }
    val byte3 = bytes[index + 1].toInt()
    if (byte3 and 0xC0 != 0x80) {
        return malformed(1, index, throwOnMalformed)
    }

    if (index + 2 == endIndex) {
        return malformed(2, index, throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte3 and 0x80 != 0) {
        return malformed(2, index, throwOnMalformed)
    }

    return (byte1 shl 12) xor (byte2 shl 6) xor byte3 xor -0x1F80
}

/**
 * Returns code point corresponding to UTF-8 sequence of four bytes, where the first byte of the sequence is the
 * [byte1] and the others are in the [bytes] array starting from the [index].
 * * Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence
 * if the sequence is malformed and [throwOnMalformed] is false.
 * @throws CharacterCodingException if the sequence is malformed and [throwOnMalformed] is true.
 */
    if (index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }

    val byte2 = bytes[index].toInt()
    if (byte1 and 0xF == 0) {
        if (byte2 and 0xF0 <= 0x80) {
            return malformed(0, index, throwOnMalformed)
        }
    } else if (byte1 and 0xF == 0x4) {
        if (byte2 and 0xF0 != 0x80) {
            return malformed(0, index, throwOnMalformed)
        }
    } else if (byte1 and 0xF > 0x4) {
        return malformed(0, index, throwOnMalformed)
    }

    if (index + 1 == endIndex) {
        return malformed(1, index, throwOnMalformed)
    }
    val byte3 = bytes[index + 1].toInt()
    if (byte3 and 0xC0 != 0x80) {
        return malformed(1, index, throwOnMalformed)
    }

    if (index + 2 == endIndex) {
        return malformed(2, index, throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte3 and 0x80 != 0) {
        return malformed(2, index, throwOnMalformed)
    }

    return (byte1 shl 12) xor (byte2 shl 6) xor byte3 xor (byte4 and 0x80)
}
throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte4 and 0xC0 != 0x80) {
        return malformed(2, index, throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte4 and 0xC0 != 0x80) {
        return malformed(2, index, throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte4 and 0xC0 != 0x80) {
        return malformed(2, index, throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte4 and 0xC0 != 0x80) {
        return malformed(2, index, throwOnMalformed)
    }
    val byte4 = bytes[index + 2].toInt()
    if (byte4 and 0xC0 != 0x80) {
        return malformed(2, index, throwOnMalformed)
    }

    /**
     * Maximum number of bytes needed to encode a single char.
     * @param string the string to encode.
     * @param startCharIndex the start offset (inclusive) of the substring to
     * @param endCharIndex the end offset (exclusive) of the substring to encode.
     * @param throwOnMalformed whether to throw on malformed char sequence or replace
     * by the replacement value.
     */
    private fun encodeUtf8(string: String, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): String {
        require(startIndex >= 0 && endIndex <= string.length && startIndex <= endIndex)

        var byteIndex = 0
        var charIndex = startIndex
        while (charIndex < endIndex) {
            val code = string[charIndex++].toInt()
            when {
                code < 0x80 ->
                    bytes[byteIndex++] = code.toByte()

                code < 0x800 -> {
                    bytes[byteIndex++] = ((code shr 6) or 0xC0).toByte()
                    bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()
                }

                code < 0xD800 || code >= 0xE000 -> {
                    bytes[byteIndex++] = ((code shr 12) or 0xE0).toByte()
                    bytes[byteIndex++] = (((code shr 6) and 0x3F) or 0x80).toByte()
                    bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()
                }

                else -> { // Surrogate char value
                    val codePoint = codePointFromSurrogate(string, code, charIndex, endIndex, throwOnMalformed)
                    if (codePoint <= 0) {
                        bytes[byteIndex++] = REPLACEMENT_CHAR
                    } else {
                        bytes[byteIndex++] = (codePoint shr 18) or 0xF0
                        bytes[byteIndex++] = (((codePoint shr 12) and 0x3F) or 0x80).toByte()
                        bytes[byteIndex++] = ((codePoint shr 6) and 0x3F) or 0x80).toByte()
                    }
                }
            }
        }

        return if (bytes.size == byteIndex) { string } else { bytes.toByteArray() }
    }

    /**
     * The character a malformed UTF-8 byte sequence is replaced by.
     */
    private const val REPLACEMENT_CHAR = '\uFFFD'

    /**
     * Decodes the UTF-8 [bytes] array and returns the resulting [String].
     * @param bytes the byte array to decode.
     * @param startIndex the start offset (inclusive) of the array to be decoded.
     * @param endIndex the end offset (exclusive) of the array to be encoded.
     * @param throwOnMalformed whether to throw on malformed byte sequence or replace
     * by the replacement value.
     */
    internal fun decodeUtf8(bytes: ByteArray, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): String {
        require(startIndex >= 0 && endIndex <= bytes.size && startIndex <= endIndex)

        var byteIndex = startIndex
        val stringBuilder = StringBuilder()
        while (byteIndex < endIndex) {
            val byte = bytes[byteIndex++].toInt()
            when {
                byte >= 0 ->
                    stringBuilder.append(byte.toChar())

                byte shr 5 == -2 -> {
                    val code = codePointFrom2(bytes, byte, byteIndex, endIndex, throwOnMalformed)
                    if (code <= 0) {
                        stringBuilder.append(REPLACEMENT_CHAR)
                    } else {
                        stringBuilder.append(code.toChar())
                        byteIndex += 1
                    }
                }

                byte shr 4 == -2 -> {
                    val code = codePointFrom3(bytes, byte, byteIndex, endIndex, throwOnMalformed)
                    if (code <= 0) {
                        stringBuilder.append(REPLACEMENT_CHAR)
                    } else {
                        stringBuilder.append(code.toChar())
                        byteIndex += 2
                    }
                }

                else -> {
                    when (byte and 0x80) {
                        if (byte and 0x40 != 0x00) {
                            val codePoint = codePointFrom1(bytes, byte, byteIndex, endIndex, throwOnMalformed)
                            if (codePoint <= 0) {
                                stringBuilder.append(REPLACEMENT_CHAR)
                            } else {
                                stringBuilder.append(code.toChar())
                                byteIndex += 1
                            }
                        } else {
                            val codePoint = codePointFrom0(bytes, byte, byteIndex, endIndex, throwOnMalformed)
                            if (codePoint <= 0) {
                                stringBuilder.append(REPLACEMENT_CHAR)
                            } else {
                                stringBuilder.append(code.toChar())
                                byteIndex += 2
                            }
                        }
                    }
                }
            }
        }

        return stringBuilder.toString()
    }
}
byte shr 3 == -2 -> {
    val code = codePointFrom4(bytes, byte, byteIndex, endIndex, throwOnMalformed)
    if (code <= 0) {StringBuilder.append(REPLACEMENT_CHAR)
        byteIndex += -code
    } else {
        val high = (code - 0x10000) shr 10 or 0xD800
        val low = (code and 0x3FF) or 0xDC00
        stringBuilder.append(high.toChar())
        stringBuilder.append(low.toChar())
        byteIndex += 3
    }
}
else -> {
    malformed(0, byteIndex, throwOnMalformed)
    stringBuilder.append(REPLACEMENT_CHAR)
}

return stringBuilder.toString()
json("maximumFractionDigits" to decimals)).unsafeCast<String>().unsafeCast<Number>().unsafeCast<Double>().formatScientific(value = Double) return value.asDynamic().toExponential(2).unsafeCast<String>().unsafeCast<Number>().unsafeCast<Double>().formatScientific(value = Double)

internal actual fun formatScientific(value: Double): String {
    return value.asDynamic().toExponential(2).unsafeCast<String>()
}

@SinceKotlin("1.4")
public fun Document.createElement(name: String, init: Element.() -> Unit): Element {
    contract { callsInPlace(init, InvocationKind.EXACTLY_ONCE) }
    return createElement(name).apply(init)
}

@SinceKotlin("1.4")
public fun Element.appendElement(name: String, init: Element.() -> Unit): Element {
    contract { callsInPlace(init, InvocationKind.EXACTLY_ONCE) }
    return ownerDocument!!.createElement(name, init).also {
        appendChild(it)
    }
}

@SinceKotlin("1.4")
fun Element.hasClass(cssClass: String): Boolean = className.matches("\"(^|.*\s+)$cssClass($|\s+.*)\"\".toRegex())

@SinceKotlin("1.4")
fun Element.addClass(vararg cssClasses: String): Boolean {
    val missingClasses = cssClasses.filterNot { hasClass(it) }
    if (missingClasses.isNotEmpty()) {
        val presentClasses = className.trim()
        className = buildString {
            append(presentClasses)
            if (!presentClasses.isEmpty()) {
                append(" ")
            }
            missingClasses.joinTo(this, " ")
        }
        return true
    }

    return false
}

@SinceKotlin("1.4")
fun Element.removeClass(vararg cssClasses: String): Boolean {
    if (cssClasses.any { hasClass(it) }) {
        val toBeRemoved = cssClasses.toSet()
        className = className.trim().split("\\s+".toRegex()).filter { it !in toBeRemoved }.joinToString(" ")
        return true
    }

    return false
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform multiple string manipulation operations.
 */
expect class StringBuilder : Appendable, CharSequence {
    /** Constructs an empty string builder. */
    constructor()

    /** Constructs an empty string builder with the specified initial [capacity]. */
    constructor(capacity: Int) **/ Constructs a string builder that contains the same characters as the specified [content] char sequence. */
    constructor(content: CharSequence) **/ Constructs a string builder that contains the same characters as the specified [content] string. */
    constructor(content: String) **/ Constructs a string builder with the specified initial [capacity]. */
    constructor(capacity: Int) **/ Converts the string into a regular expression [Regex] with the default options. */
    constructor(regExp: String) **/ Converts the string into a regular expression [Regex] with the specified single [option]. */
    constructor(regExp: Regex, option: RegexOption) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any, options: Set<RegexOption>) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

import kotlin.contracts.*

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform multiple string manipulation operations.
 */
expect class StringBuilder : Appendable, CharSequence {
    /** Constructs an empty string builder. */
    constructor()

    /** Constructs an empty string builder with the specified initial [capacity]. */
    constructor(capacity: Int) **/ Constructs a string builder that contains the same characters as the specified [content] char sequence. */
    constructor(content: CharSequence) **/ Constructs a string builder that contains the same characters as the specified [content] string. */
    constructor(content: String) **/ Converts the string into a regular expression [Regex] with the default options. */
    constructor(regExp: String) **/ Converts the string into a regular expression [Regex] with the specified single [option]. */
    constructor(regExp: Regex, option: RegexOption) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any, options: Set<RegexOption>) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

import kotlin.contracts.*

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform multiple string manipulation operations.
 */
expect class StringBuilder : Appendable, CharSequence {
    /** Constructs an empty string builder. */
    constructor()

    /** Constructs an empty string builder with the specified initial [capacity]. */
    constructor(capacity: Int) **/ Constructs a string builder that contains the same characters as the specified [content] char sequence. */
    constructor(content: CharSequence) **/ Constructs a string builder that contains the same characters as the specified [content] string. */
    constructor(content: String) **/ Converts the string into a regular expression [Regex] with the default options. */
    constructor(regExp: String) **/ Converts the string into a regular expression [Regex] with the specified single [option]. */
    constructor(regExp: Regex, option: RegexOption) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any, options: Set<RegexOption>) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

import kotlin.contracts.*

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform multiple string manipulation operations.
 */
expect class StringBuilder : Appendable, CharSequence {
    /** Constructs an empty string builder. */
    constructor()

    /** Constructs an empty string builder with the specified initial [capacity]. */
    constructor(capacity: Int) **/ Constructs a string builder that contains the same characters as the specified [content] char sequence. */
    constructor(content: CharSequence) **/ Constructs a string builder that contains the same characters as the specified [content] string. */
    constructor(content: String) **/ Converts the string into a regular expression [Regex] with the default options. */
    constructor(regExp: String) **/ Converts the string into a regular expression [Regex] with the specified single [option]. */
    constructor(regExp: Regex, option: RegexOption) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any, options: Set<RegexOption>) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

import kotlin.contracts.*

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform multiple string manipulation operations.
 */
expect class StringBuilder : Appendable, CharSequence {
    /** Constructs an empty string builder. */
    constructor()

    /** Constructs an empty string builder with the specified initial [capacity]. */
    constructor(capacity: Int) **/ Constructs a string builder that contains the same characters as the specified [content] char sequence. */
    constructor(content: CharSequence) **/ Constructs a string builder that contains the same characters as the specified [content] string. */
    constructor(content: String) **/ Converts the string into a regular expression [Regex] with the default options. */
    constructor(regExp: String) **/ Converts the string into a regular expression [Regex] with the specified single [option]. */
    constructor(regExp: Regex, option: RegexOption) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any, options: Set<RegexOption>) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StringsKt")

package kotlin.text

import kotlin.contracts.*

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform multiple string manipulation operations.
 */
expect class StringBuilder : Appendable, CharSequence {
    /** Constructs an empty string builder. */
    constructor()

    /** Constructs an empty string builder with the specified initial [capacity]. */
    constructor(capacity: Int) **/ Constructs a string builder that contains the same characters as the specified [content] char sequence. */
    constructor(content: CharSequence) **/ Constructs a string builder that contains the same characters as the specified [content] string. */
    constructor(content: String) **/ Converts the string into a regular expression [Regex] with the default options. */
    constructor(regExp: String) **/ Converts the string into a regular expression [Regex] with the specified single [option]. */
    constructor(regExp: Regex, option: RegexOption) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
    constructor(content: Any, options: Set<RegexOption>) **/ Converts the string into a regular expression [Regex] with the specified set of [options]. */
Surrogate pairs included in this string builder are treated as single characters. Therefore, the order of the high-
low surrogates is never reversed. Note that the reverse operation may produce new surrogate pairs that
were unpaired low-surrogates and high-surrogates before the operation. For example, reversing
`"\u00DD\u00CD"` produces `"\u00CD\u00DD"` which is a valid surrogate pair. fun reverse():

The overall effect is exactly as if the [value] were converted to a string by the
`value.toString()` method, and then that string was appended to this string builder. fun append(value: Any?): StringBuilder

The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method, and then that string was appended to this string builder..fun append(value: Boolean): StringBuilder

`null` produces `"null"` which is a valid surrogate pair.

Characters are appended in order, starting at the index 0. fun ensureCapacity(minimumCapacity: Int): StringBuilder

* If [value] is `null`, then the four characters `"null\000"` are appended.

* The capacity is the maximum length this string builder can have before an allocation occurs.

* If the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.

* Otherwise, this method takes no action and simply returns.

* Returns `-1` if the specified [string] does not occur in this string builder.

* Returns the index within this string builder of the first occurrence of the specified [string].

* Returns `-1` if the specified [string] does not occur in this string builder starting at the specified [startIndex].

* The last occurrence of empty string `"\000"` is considered to be at the index equal to `this.length - 1` if the specified [string] does not occur in this string builder.

* Returns the index within this string builder of the last occurrence of the specified [string].

* Inserts the string representation of the specified boolean [value] into this string builder and returns this instance.

* The overall effect is as if the [value] were converted to a string by the `value.toString()` method, and then that string was inserted into this string builder at the specified [index].

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.

* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.

* Returns this instance.
with characters in the specified string [value] and returns this instance.

*\n* @param startIndex the beginning (inclusive) of the range to replace.
*\n* @param endIndex the end (exclusive) of the range to replace.
*\n* @throws IndexOutOfBoundsException or [IllegalStateException] if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.

*\n* @SinceKotlin("1.4")\n*\n* @WasExperimental(ExperimentalStdlibApi::class)\n*\n* @public expect fun

StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder

/**
 * Removes the character at the specified [index] from this string builder and returns this instance.
*\n* @param index the index of `Char` to remove.
*\n* @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.

*\n* @SinceKotlin("1.4")\n*\n* @WasExperimental(ExperimentalStdlibApi::class)\n*\n* @public expect fun

StringBuilder.deleteAt(index: Int): StringBuilder

/**
 * Removes characters in the specified range from this string builder and returns this instance.
*\n* @param startIndex the beginning (inclusive) of the range to remove.
*\n* @param endIndex the end (exclusive) of the range to remove.
*\n* @throws IndexOutOfBoundsException or [IllegalStateException] when [startIndex] is out of bounds of this string builder indices or when `startIndex > endIndex`.

*\n* @SinceKotlin("1.4")\n*\n* @WasExperimental(ExperimentalStdlibApi::class)\n*\n* @public expect fun

StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder

/**
 * Copies characters from this string builder into the [destination] character array.
*\n* @param destination the array to copy to.
*\n* @param startIndex the beginning (inclusive) of the range to copy, 0 by default.
*\n* @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.
*\n* @throws IndexOutOfBoundsException or [IllegalStateException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.

*\n* @SinceKotlin("1.4")\n*\n* @WasExperimental(ExperimentalStdlibApi::class)\n*\n* @public expect fun

StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length): CharArray

/**
 * Appends characters in a subarray of the specified character array [value] to this string builder
*\n* @param value the array from which characters are appended.
*\n* @param startIndex the beginning (inclusive) of the subarray to append.
*\n* @throws IndexOutOfBoundsException or [IllegalStateException] when [startIndex] is out of bounds of the [value] array indices or when `startIndex > endIndex`.

*\n* @SinceKotlin("1.4")\n*\n* @WasExperimental(ExperimentalStdlibApi::class)\n*\n* @public expect fun

StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder

/**
 * Appends a subsequence of the specified character sequence [value] to this string builder
*\n* @param value the character sequence from which a subsequence is appended.
*\n* @throws IndexOutOfBoundsException or [IllegalStateException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.

*\n* @SinceKotlin("1.4")\n*\n* @WasExperimental(ExperimentalStdlibApi::class)\n*\n* @public expect fun

StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder

/**
 * Inserts characters in a subarray of the specified character array [value] into this string builder
*\n* @param value the character sequence from which a subsequence is appended.
*\n* @param startIndex the beginning (inclusive) of the subsequence to append.
*\n* @param endIndex the end (exclusive) of the subsequence to append.
*\n* @throws IndexOutOfBoundsException or [IllegalStateException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.

*\n* @SinceKotlin("1.4")\n*\n* @WasExperimental(ExperimentalStdlibApi::class)\n*\n* @public expect fun

StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder

/**
 * Inserts characters in a subarray of the specified character array [value] into this string builder
*\n* @param value the character sequence from which a subsequence is appended.
*\n* @param startIndex the beginning (inclusive) of the subsequence to append.
*\n* @param endIndex the end (exclusive) of the subsequence to append.
*\n* @throws IndexOutOfBoundsException or [IllegalStateException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.

*\n* @SinceKotlin("1.4")\n*\n* @WasExperimental(ExperimentalStdlibApi::class)\n*\n* @public expect fun

StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public expect fun StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public expect fun StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder

@kotlin.internal.InlineOnly
@Suppress("EXTENSION_SHADOWED_BY_MEMBER")
@Deprecated("Use append(value: Any?) instead", ReplaceWith("append(value = obj)"), DeprecationLevel.WARNING)
public inline fun StringBuilder.append(obj: Any?): StringBuilder = this.append(obj)

public fun StringBuilder.append(vararg value: String?): StringBuilder {
    for (item in value)
        append(item)
    return this
}

public fun StringBuilder.append(vararg value: Any?): StringBuilder {
    for (item in value)
        append(item)
    return this
}

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun StringBuilder.appendLine(): StringBuilder = append('
')

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun StringBuilder.appendLine(value: CharSequence?): StringBuilder = append(value).appendLine()

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun StringBuilder.appendLine(value: String?): StringBuilder = append(value).appendLine()

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun StringBuilder.appendLine(value: Any?): StringBuilder = append(value).appendLine()
org.w3c.dom.*

/**
 * Gets a value indicating whether this node is a TEXT_NODE or a CDATA_SECTION_NODE.
 */
@SinceKotlin("1.4")
public val Node.isText: Boolean
    get() = nodeType == Node.TEXT_NODE || nodeType == Node.CDATA_SECTION_NODE

/**
 * Gets a value indicating whether this node is an [Element].
 */
@SinceKotlin("1.4")
public val Node.isElement: Boolean
    get() = nodeType == Node.ELEMENT_NODE

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlinx.dom

import org.w3c.dom.*

/** Removes all the children from this node. */
@SinceKotlin("1.4")
public fun Node.clear() {
    while (hasChildNodes()) {
        removeChild(firstChild!!)
    }
}

/**
 * Creates text node and append it to the element.
 * @return this element
 */
@SinceKotlin("1.4")
fun Element.appendText(text: String): Element {
    appendChild(ownerDocument!!.createTextNode(text))
    return this
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.w3c.dom

@Deprecated("Use UnionMessagePortOrWindowProxy instead.", ReplaceWith("UnionMessagePortOrWindowProxy"))
typealias UnionMessagePortOrWindow = UnionMessagePortOrWindowProxy

@Deprecated("Use `as` instead.", ReplaceWith("`as`"))
var HTMLLinkElement.as_
    get() = `as`
    set(value) {
        `as` = value
    }

@Deprecated("Use `is` instead.", ReplaceWith("`is`"))
var ElementCreationOptions.is_
    get() = `is`
    set(value) {
        `is` = value
    }

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming
 * Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.khronos.webgl

import kotlin.js.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

public external interface WebGLContextAttributes {
    var alpha: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var depth: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var stencil: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var antialias: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var premultipliedAlpha: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var preserveDrawingBuffer: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var preferLowPowerToHighPerformance: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var failIfMajorPerformanceCaveat: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["alpha"] = alpha
    o["depth"] = depth
    o["stencil"] = stencil
    o["antialias"] = antialias
    o["premultipliedAlpha"] = premultipliedAlpha
    o["preserveDrawingBuffer"] = preserveDrawingBuffer
    o["preferLowPowerToHighPerformance"] = preferLowPowerToHighPerformance
    o["failIfMajorPerformanceCaveat"] = failIfMajorPerformanceCaveat
    return o
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.dom

public external interface WebGLObject

/**
 * Exposes the JavaScript [WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin
 */
public external class WebGLBuffer

/**
 * Exposes the JavaScript [WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin
 */
public external class WebGLFramebuffer

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.dom

public external interface WebGLObject

/**
 * Exposes the JavaScript [WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin
 */
public external class WebGLBuffer

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.dom

public external interface WebGLObject

/**
 * Exposes the JavaScript [WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin
 */
public external class WebGLBuffer

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.dom

public external interface WebGLObject

/**
 * Exposes the JavaScript [WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin
 */
public external class WebGLBuffer

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.dom

public external interface WebGLObject

/**
 * Exposes the JavaScript [WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin
 */
public external class WebGLBuffer

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.dom

public external interface WebGLObject

/**
 * Exposes the JavaScript [WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin
 */
public external class WebGLBuffer

Open Source Used In DNAC Platform Antelope 9562
Open Source Used In DNAC Platform Antelope

[WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin

[WebGLRendererbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRendererbuffer) to Kotlin

[WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin

[WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin

[WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin

[WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin

[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to Kotlin

```kotlin
public external abstract class WebGLProgram : WebGLObject

/**
 * Exposes the JavaScript
[WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin

[WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin

[WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin

[WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin

[WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin

[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to Kotlin
```

```kotlin
/**
 * Exposes the JavaScript

[WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin

[WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin

[WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin

[WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin

[WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin

[WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to Kotlin
```
Int, attachment: Int, texture: WebGLTexture?, level: Int)
fun frontFace(mode: Int)
fun generateMipmap(target: Int)
fun getActiveAttrib(program: WebGLProgram?, index: Int): WebGLActiveInfo?
fun getActiveUniform(program: WebGLProgram?, index: Int): WebGLActiveInfo?
fun getAttachedShaders(program: WebGLProgram?): Array<WebGLShader>?
fun getAttribLocation(program: WebGLProgram?, name: String): Int
fun getBufferParameter(target: Int, pname: Int): Any?
fun getParameter(pname: Int): Any?
fun getError(): Int
fun getFramebufferAttachmentParameter(target: Int, attachment: Int, pname: Int): Int?
fun getProgramParameter(program: WebGLProgram?, pname: Int): Any?
fun getProgramInfoLog(program: WebGLProgram?): String?
fun getRenderbufferParameter(target: Int, pname: Int): Any?
fun getShaderParameter(shader: WebGLShader?, pname: Int): Any?
fun getTexParameter(target: Int, pname: Int): Any?
fun getUniform(program: WebGLProgram?, location: WebGLUniformLocation?): Any?
fun getUniformLocation(program: WebGLProgram?, name: String): WebGLUniformLocation?
fun getVertexAttrib(index: Int, pname: Int): Any?
fun getVertexAttribOffset(index: Int, pname: Int): Int
fun hint(target: Int, mode: Int)
fun isEnabled(cap: Int): Boolean
fun isBuffer(buffer: WebGLBuffer?): Boolean
fun isFramebuffer(framebuffer: WebGLFramebuffer?): Boolean
fun isProgram(program: WebGLProgram?): Boolean
fun isRenderbuffer(renderbuffer: WebGLRenderbuffer?): Boolean
fun isShader(shader: WebGLShader?): Boolean
fun isTexture(texture: WebGLTexture?): Boolean
fun lineWidth(width: Float)
fun linkProgram(program: WebGLProgram?)
fun pixelStorei(pname: Int, param: Int)
fun polygonOffset(factor: Float, units: Float)
fun readPixels(x: Int, y: Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?)
fun renderbufferStorage(target: Int, internalformat: Int, width: Int, height: Int)
fun sampleCoverage(value: Float, invert: Boolean)
fun scissor(x: Int, y: Int, width: Int, height: Int)
fun shaderSource(shader: WebGLShader?, source: String)
fun stencilFunc(func: Int, ref: Int, mask: Int)
fun stencilFuncSeparate(face: Int, func: Int, ref: Int, mask: Int)
fun stencilMask(mask: Int)
fun stencilMaskSeparate(face: Int, mask: Int)
fun stencilOp(fail: Int, zfail: Int, zpass: Int)
fun texImage2D(target: Int, level: Int, internalformat: Int, format: Int, type: Int, pixels: ArrayBufferView?)
fun texImage2D(target: Int, level: Int, format: Int, type: Int, source: TexImageSource?)
fun texParameterf(target: Int, pname: Int, param: Float)
fun texParameteri(target: Int, pname: Int, param: Int)
fun uniform1f(location: WebGLUniformLocation?, x: Float)
fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform1fv(location: WebGLUniformLocation?, v: Array<Float>)
fun uniform1i(location: WebGLUniformLocation?, x: Int)
fun uniform1iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform1iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform2f(location: WebGLUniformLocation?, x: Float, y: Float)
fun uniform2fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform2fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform2iv(location: WebGLUniformLocation?, x: Int, y: Int)
fun uniform2iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform2iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float)
fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform3iv(location: WebGLUniformLocation?, x: Int, y: Int, z: Int)
fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float, w: Float)
fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform4iv(location: WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int)
fun uniform4iv(location: WebGLUniformLocation?, v: Int32Array)
useProgram(program: WebGLProgram?)
fun validateProgram(program: WebGLProgram?)

fun vertexAttrib1f(index: Int, x: Float)
fun vertexAttrib1fv(index: Int, values: dynamic)

fun vertexAttrib2f(index: Int, x: Float, y: Float)
fun vertexAttrib2fv(index: Int, values: dynamic)

fun vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)
fun vertexAttrib3fv(index: Int, values: dynamic)

fun vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)
fun vertexAttrib4fv(index: Int, values: dynamic)


fun viewport(x: Int, y: Int, width: Int, height: Int)
fun companion object {
    val DEPTH_BUFFER_BIT: Int
    val STENCIL_BUFFER_BIT: Int
    val COLOR_BUFFER_BIT: Int
    val POINTS: Int
    val LINES: Int
    val LINE_LOOP: Int
    val LINE_STRIP: Int
    val TRIANGLES: Int
    val TRIANGLE_STRIP: Int
    val TRAVERSE: Int
    val ZERO: Int
    val ONE: Int
    val SRC_COLOR: Int
    val ONE_MINUS_SRC_COLOR: Int
    val SRC_ALPHA: Int
    val ONE_MINUS_SRC_ALPHA: Int
    val DST_ALPHA: Int
    val ONE_MINUS_DST_ALPHA: Int
    val DST_COLOR: Int
    val ONE_MINUS_DST_COLOR: Int
    val BLEND_EQUATION: Int
    val BLEND_EQUATION_RGB: Int
    val BLEND_EQUATION_ALPHA: Int
    val BLEND_SRC_RGB: Int
    val BLEND_SRC_ALPHA: Int
    val BLEND_DST_RGB: Int
    val BLEND_DST_ALPHA: Int
    val BLEND_COLOR: Int
    val ARRAY_BUFFER: Int
    val ELEMENT_ARRAY_BUFFER: Int
    val ARRAY_BUFFER_BINDING: Int
    val ELEMENT_ARRAY_BUFFER_BINDING: Int
    val STREAM_DRAW: Int
    val STATIC_DRAW: Int
    val DYNAMIC_DRAW: Int
    val BUFFER_SIZE: Int
    val BUFFER_USAGE: Int
    val CURRENT_VERTEX_ATTRIB: Int
    val FRONT: Int
    val BACK: Int
    val FRONT_AND_BACK: Int
    val CULL_FACE: Int
    val BLEND: Int
    val DITHER: Int
    val STENCIL_TEST: Int
    val DEPTH_TEST: Int
    val POLYGON_OFFSET_FILL: Int
    val SAMPLE_ALPHA_TO_COVERAGE: Int
    val SAMPLE_COVERAGE: Int
    val NO_ERROR: Int
    val INVALID_ENUM: Int
    val INVALID_VALUE: Int
    val INVALID_OPERATION: Int
    val OUT_OF_MEMORY: Int
    val CW: Int
    val CCW: Int
    val LINE_WIDTH: Int
    val ALIASED_POINT_SIZE_RANGE: Int
    val ALIASED_LINE_WIDTH_RANGE: Int
    val CULL_FACE_MODE: Int
    val FRONT_FACE: Int
    val DEPTH_RANGE: Int
    val DEPTH_WRITEMASK: Int
    val DEPTH_CLEAR_VALUE: Int
    val DEPTH_FUNC: Int
    val STENCIL_CLEAR_VALUE: Int
    val STENCIL_FUNC: Int
    val STENCIL_FAIL: Int
    val STENCIL_PASS_DEPTH_FAIL: Int
    val STENCIL_PASS_DEPTH_PASS: Int
    val STENCIL_REF: Int
    val STENCIL_VALUE_MASK: Int
    val STENCIL_BACK_FUNC: Int
    val STENCIL_BACK_FAIL: Int
    val STENCIL_BACK_PASS_DEPTH_FAIL: Int
    val STENCIL_BACK_PASS_DEPTH_PASS: Int
    val STENCIL_BACK_REF: Int
    val STENCIL_BACK_VALUE_MASK: Int
    val VIEWPORT: Int
    val SCISSOR_BOX: Int
    val COLOR_CLEAR_VALUE: Int
    val COLOR_WRITEMASK: Int
    val UNPACK_ALIGNMENT: Int
    val PACK_ALIGNMENT: Int
    val MAX_TEXTURE_SIZE: Int
    val MAX_VIEWPORT_DIMS: Int
    val SUBPIXEL_BITS: Int
    val RED_BITS: Int
    val GREEN_BITS: Int
    val BLUE_BITS: Int
    val ALPHA_BITS: Int
    val DEPTH_BITS: Int
    val STENCIL_BITS: Int

POLYGON_OFFSET_UNITS: Int  val POLYGON_OFFSET_FACTOR: Int  val
TEXTURE_BINDING_2D: Int  val SAMPLE_BUFFERS: Int  val SAMPLES: Int  val
SAMPLE_COVERAGE_VALUE: Int  val SAMPLE_COVERAGE_INVERT: Int  val
COMPRESSED_TEXTURE_FORMATS: Int  val DONT_CARE: Int  val FASTEST: Int  val
NICEST: Int  val GENERATE_MIPMAP_HINT: Int  val BYTE: Int  val UNSIGNED_BYTE:
Int  val SHORT: Int  val UNSIGNED_SHORT: Int  val INT: Int  val UNSIGNED_INT: Int  val
FLOAT: Int  val DEPTH_COMPONENT: Int  val ALPHA: Int  val RGB: Int  val RGBA:
Int  val LUMINANCE: Int  val LUMINANCE_ALPHA: Int  val UNSIGNED_SHORT_4_4_4_4:
Int  val UNSIGNED_SHORT_5_5_5_1: Int  val UNSIGNED_SHORT_5_6_5: Int  val FRAGMENT_SHADER:
Int  val VERTEX_SHADER: Int  val MAX_VERTEX_ATTRIBS: Int  val MAX_VERTEX_UNIFORM_VECTORS:
Int  val MAX_VARYING_VECTORS: Int  val MAX_COMBINED_TEXTURE_IMAGE_UNITS:
Int  val MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int  val MAX_TEXTURE_IMAGE_UNITS:
Int  val SHADER_TYPE: Int  val DELETE_STATUS: Int  val VALIDATE_STATUS: Int  val
ATTACHED_SHADERS: Int  val ACTIVE_UNIFORMS: Int  val ACTIVE_ATTRIBUTES: Int  val
SHADING_LANGUAGE_VERSION: Int  val CURRENT_PROGRAM: Int  val NEVER: Int  val
DEC: Int  val INVERT: Int  val INCR_WRAP: Int  val DECR_WRAP: Int  val
VENDOR: Int  val RENDERER: Int  val VERSION: Int  val LINEAR: Int  val
NEAREST_MIPMAP_NEAREST: Int  val LINEAR_MIPMAP_NEAREST: Int  val
NEAREST_MIPMAP_LINEAR: Int  val LINEAR_MIPMAP_LINEAR: Int  val
TEXTURE_MAG_FILTER: Int  val TEXTURE_MIN_FILTER: Int  val TEXTURE_WRAP_S:
Int  val TEXTURE_WRAP_T: Int  val TEXTURE_2D: Int  val TEXTURE: Int  val
TEXTURE_CUBE_MAP: Int  val TEXTURE_BINDING_CUBE_MAP: Int  val
TEXTURE_CUBE_MAP_POSITIVE_X: Int  val TEXTURE_CUBE_MAP_NEGATIVE_X: Int  val
TEXTURE_CUBE_MAP_POSITIVE_Y: Int  val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int  val
TEXTURE_CUBE_MAP_POSITIVE_Z: Int  val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int  val
MAX_CUBE_MAP_TEXTURE_SIZE: Int  val TEXTURE0: Int  val TEXTURE1: Int  val
TEXTURE2: Int  val TEXTURE3: Int  val TEXTURE4: Int  val TEXTURE5: Int  val
TEXTURE6: Int  val TEXTURE7: Int  val TEXTURE8: Int  val TEXTURE9: Int  val
TEXTURE14: Int  val TEXTURE15: Int  val TEXTURE16: Int  val TEXTURE17: Int  val
TEXTURE30: Int  val TEXTURE31: Int  val TEXTURE32: Int  val TEXTURE33: Int  val
TEXTURE34: Int  val TEXTURE35: Int  val TEXTURE36: Int  val TEXTURE37: Int  val
TEXTURE38: Int  val TEXTURE39: Int  val TEXTURE40: Int  val TEXTURE41: Int  val
TEXTURE42: Int  val TEXTURE43: Int  val TEXTURE44: Int  val TEXTURE45: Int  val
TEXTURE50: Int  val TEXTURE51: Int  val TEXTURE52: Int  val TEXTURE53: Int  val
TEXTURE54: Int  val TEXTURE55: Int  val TEXTURE56: Int  val TEXTURE57: Int  val
TEXTURE58: Int  val TEXTURE59: Int  val TEXTURE60: Int  val TEXTURE61: Int  val
CLAMP_TO_EDGE: Int  val MIRRORED_REPEAT: Int  val FLOAT_VEC2: Int  val
FLOAT_VEC3: Int  val FLOAT_VEC4: Int  val INT_VEC2: Int  val INT_VEC3: Int  val
BOOL_VEC4: Int  val FLOAT_MAT2: Int  val FLOAT_MAT3: Int  val FLOAT_MAT4: Int  val
SAMPLER_2D: Int  val SAMPLER_CUBE: Int  val VERTEX_ATTRIB_ARRAY_SIZE:
Int  val VERTEX_ATTRIB_ARRAY_TYPE: Int  val VERTEX_ATTRIB_ARRAY_STRIDE:
Int  val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int  val
VERTEX_ATTRIB_ARRAY_POINTER: Int  val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING:
Int  val IMPLEMENTATION_COLOR_READ_TYPE: Int  val
IMPLEMENTATION_COLOR_READ_FORMAT: Int  val COMPIL
OPEN_SOURCE_USED_IN_DNAC_PLATFORM_ANTELope
DEPTH_FUNC: Int
val STENCIL_CLEAR_VALUE: Int
val STENCIL_FUNC: Int
val STENCIL_FAIL: Int
val STENCIL_PASS_DEPTH_FAIL: Int
val STENCIL_PASS_DEPTH_PASS: Int
val STENCIL_REF: Int
val STENCIL_VALUE_MASK: Int
val STENCIL_WRITEMASK: Int
val STENCIL_BACK_FUNC: Int
val STENCIL_BACK_FAIL: Int
val STENCIL_BACK_PASS_DEPTH_FAIL: Int
val STENCIL_BACK_PASS_DEPTH_PASS: Int
val STENCIL_BACK_REF: Int
val STENCIL_BACK_VALUE_MASK: Int
val VIEWPORT: Int
val SCISSOR_BOX: Int
val COLOR_CLEAR_VALUE: Int
val COLOR_WRITEMASK: Int
val UNPACK_ALIGNMENT: Int
val PACK_ALIGNMENT: Int
val MAX_TEXTURE_SIZE: Int
val MAX_VIEWPORT_DIMS: Int
val SUBPIXEL_BITS: Int
val RED_BITS: Int
val GREEN_BITS: Int
val BLUE_BITS: Int
val ALPHA_BITS: Int
val DEPTH_BITS: Int
val STENCIL_BITS: Int
val POLYGON_OFFSET_UNITS: Int
val POLYGON_OFFSET_FACTOR: Int
val TEXTURE_BINDING_2D: Int
val SAMPLE_BUFFERS: Int
val SAMPLES: Int
val SAMPLE_COVERAGE_VALUE: Int
val SAMPLE_COVERAGE_INVERT: Int
val COMPRESSED_TEXTURE_FORMATS: Int
val DONT_CARE: Int
val FASTEST: Int
val NICEST: Int
val GENERATE_MIPMAP_HINT: Int
val BYTE: Int
val UNSIGNED_BYTE: Int
val SHORT: Int
val UNSIGNED_SHORT: Int
val INT: Int
val UNSIGNED_INT: Int
val FLOAT: Int
val DEPTH_COMPONENT: Int
val RGB: Int
val LUMINANCE: Int
val LUMINANCE_ALPHA: Int
val UNSIGNED_SHORT_4_4_4_4: Int
val UNSIGNED_SHORT_5_5_5_1: Int
val UNSIGNED_SHORT_5_6_5: Int
val FRAGMENT_SHADER: Int
val VERTEX_SHADER: Int
val MAX_VERTEX_ATTRIBS: Int
val MAX_VERTEX_UNIFORM_VECTORS: Int
val MAX_VARYING_VECTORS: Int
val MAX_COMBINED_TEXTURE_IMAGE_UNITS: Int
val MAX_TEXTURE_IMAGE_UNITS: Int
val MAX_FRAGMENT_UNIFORM_VECTORS: Int
val SHADER_TYPE: Int
val DELETE_STATUS: Int
val LINK_STATUS: Int
val VALIDATE_STATUS: Int
val ATTACHED_SHADERS: Int
val ACTIVE_UNIFORMS: Int
val ACTIVE_ATTRIBUTES: Int
val SHADING_LANGUAGE_VERSION: Int
val CURRENT_PROGRAM: Int
val NEVER: Int
val LESS: Int
val EQUAL: Int
val LEQUAL: Int
val GREATER: Int
val NOTEQUAL: Int
val ALWAYS: Int
val KEEP: Int
val REPLACE: Int
val INCR: Int
val DECR: Int
val INVERT: Int
val INCR_WRAP: Int
val DECR_WRAP: Int
val VENDOR: Int
val RENDERER: Int
val VERSION: Int
val NEAREST: Int
val LINEAR: Int
val NEAREST_MIPMAP_NEAREST: Int
val LINEAR_MIPMAP_NEAREST: Int
val TEXTURE_CUBE_MAP_LINEAR: Int
val TEXTURE_CUBE_MAP: Int
val TEXTURE_CUBE_MAP_POSITIVE_X: Int
val TEXTURE_CUBE_MAP_NEGATIVE_X: Int
val TEXTURE_CUBE_MAP_POSITIVE_Y: Int
val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int
val TEXTURE_CUBE_MAP_POSITIVE_Z: Int
val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int
val TEXTURE_CUBE_MAP_POSITIVE_X: Int
val TEXTURE_CUBE_MAP_NEGATIVE_X: Int
val TEXTURE_CUBE_MAP_POSITIVE_Y: Int
val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int
val TEXTURE_CUBE_MAP_POSITIVE_Z: Int
val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int
val MAX_CUBE_MAP_TEXTURE_SIZE: Int
val TEXTURE0: Int
val TEXTURE1: Int
val TEXTURE2: Int
val TEXTURE3: Int
val TEXTURE4: Int
val TEXTURE5: Int
val TEXTURE6: Int
val TEXTURE7: Int
val TEXTURE8: Int
val TEXTURE9: Int
val TEXTURE10: Int
val TEXTURE11: Int
val TEXTURE12: Int
val TEXTURE13: Int
val TEXTURE14: Int
val TEXTURE15: Int
val TEXTURE16: Int
val TEXTURE17: Int
val TEXTURE18: Int
val TEXTURE19: Int
val TEXTURE20: Int
val TEXTURE21: Int
val TEXTURE22: Int
val TEXTURE23: Int
val TEXTURE24: Int
val TEXTURE25: Int
val TEXTURE26: Int
val TEXTURE27: Int
val TEXTURE28: Int
val TEXTURE29: Int
textContentStart

/**
 * Exposes the JavaScript WebGLContextEvent to Kotlin
 */
public external open class WebGLContextEvent(type: String, eventInit: WebGLContextEventInit = definedExternally) : Event {
    open val statusMessage: String

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    fun get() = definedExternally
    fun set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["statusMessage"] = statusMessage
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript ArrayBuffer to Kotlin
 */
public external open class ArrayBuffer(length: Int) : BufferDataSource {
    open val byteLength: Int
    fun slice(begin: Int, end: Int = definedExternally): ArrayBuffer

    companion object {
        fun isView(value: Any?): Boolean
    }
}

/**
 * Exposes the JavaScript ArrayBufferView to Kotlin
 */
public
external interface ArrayBufferView : BufferDataSource {
    val buffer: ArrayBuffer
    val byteOffset: Int
    val byteLength: Int
}

/**
 * Exposes the JavaScript [Int8Array](https://developer.mozilla.org/en/docs/Web/API/Int8Array) to Kotlin
 */
public external open class Int8Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Int8Array)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun set(array: Int8Array, offset: Int = definedExternally)
    fun set(array: Array<Byte>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Int8Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Int8Array.get(index: Int): Byte = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Int8Array.set(index: Int, value: Byte) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript [Uint8Array](https://developer.mozilla.org/en/docs/Web/API/Uint8Array) to Kotlin
 */
public external open class Uint8Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint8Array)
    constructor(array: Array<Byte>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun set(array: Uint8Array, offset: Int = definedExternally)
    fun set(array: Array<Byte>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint8Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint8Array.get(index: Int): Byte = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint8Array.set(index: Int, value: Byte) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript [Uint8ClampedArray](https://developer.mozilla.org/en/docs/Web/API/Uint8ClampedArray) to Kotlin
 */
public external open class Uint8ClampedArray : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint8ClampedArray)
    constructor(array: Array<Byte>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun set(array: Uint8ClampedArray, offset: Int = definedExternally)
    fun set(array: Array<Byte>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint8ClampedArray

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint8ClampedArray.get(index: Int): Byte = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint8ClampedArray.set(index: Int, value: Byte) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript [Int16Array](https://developer.mozilla.org/en/docs/Web/API/Int16Array) to Kotlin
 */
public external open class Int16Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Int16Array)
    constructor(array: Array<Short>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun set(array: Int16Array, offset: Int = definedExternally)
    fun set(array: Array<Short>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Int16Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Int16Array.get(index: Int): Short = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Int16Array.set(index: Int, value: Short) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript [Uint16Array](https://developer.mozilla.org/en/docs/Web/API/Uint16Array) to Kotlin
 */
public external open class Uint16Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint16Array)
    constructor(array: Array<Short>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    fun set(array: Uint16Array, offset: Int = definedExternally)
    fun set(array: Array<Short>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint16Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint16Array.get(index: Int): Short = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint16Array.set(index: Int, value: Short) { asDynamic()[index] = value }
Open Source Used In DNAC Platform Antelope 9571

[Int32Array](https://developer.mozilla.org/en/docs/Web/API/Int32Array) to Kotlin

```kotlin
open class Int32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Int32Array)
    constructor(array: Array<Int>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = 0, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int
    override val byteLength: Int
    fun set(array: Int32Array, offset: Int = 0)
    fun set(array: Array<Int>, offset: Int = 0)
    fun subarray(start: Int, end: Int): Int32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Int32Array.get(index: Int): Int = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Int32Array.set(index: Int, value: Int) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript [Uint32Array](https://developer.mozilla.org/en/docs/Web/API/Uint32Array) to Kotlin
 */
public class Uint32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint32Array)
    constructor(array: Array<Int>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = 0, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int
    override val byteLength: Int
    fun set(array: Uint32Array, offset: Int = 0)
    fun set(array: Array<Int>, offset: Int = 0)
    fun subarray(start: Int, end: Int): Uint32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint32Array.get(index: Int): Int = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint32Array.set(index: Int, value: Int) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript [Float32Array](https://developer.mozilla.org/en/docs/Web/API/Float32Array) to Kotlin
 */
public class Float32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Float32Array)
    constructor(array: Array<Float>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = 0, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int
    override val byteLength: Int
    fun set(array: Float32Array, offset: Int = 0)
    fun set(array: Array<Float>, offset: Int = 0)
    fun subarray(start: Int, end: Int): Float32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Float32Array.get(index: Int): Float = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Float32Array.set(index: Int, value: Float) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript [Float64Array](https://developer.mozilla.org/en/docs/Web/API/Float64Array) to Kotlin
 */
public class Float64Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Float64Array)
    constructor(array: Array<Double>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = 0, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int
    override val byteLength: Int
    fun set(array: Float64Array, offset: Int = 0)
    fun set(array: Array<Double>, offset: Int = 0)
    fun subarray(start: Int, end: Int): Float64Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Float64Array.get(index: Int): Double = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Float64Array.set(index: Int, value: Double) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript [DataView](https://developer.mozilla.org/en/docs/Web/API/DataView) to Kotlin
 */
public class DataView(buffer: ArrayBuffer, byteOffset: Int = 0, byteLength: Int = definedExternally) : ArrayBufferView {
    override val buffer: ArrayBuffer
    override val byteOffset: Int
    override val byteLength: Int
    fun getInt8(offset: Int): Byte
    fun getUint8(offset: Int): Uint8
    fun getInt16(offset: Int, littleEndian: Boolean = true): Short
    fun getUint16(offset: Int, littleEndian: Boolean = true): Uint16
    fun getInt32(offset: Int, littleEndian: Boolean = true): Int
    fun getUint32(offset: Int, littleEndian: Boolean = true): Uint32
    fun getFloat32(offset: Int, littleEndian: Boolean = true): Float
    fun getFloat64(offset: Int, littleEndian: Boolean = true): Float64
    fun setInt8(offset: Int, value: Byte): Unit
    fun setUint8(offset: Int, value: Uint8): Unit
    fun setInt16(offset: Int, value: Short, littleEndian: Boolean = true): Unit
    fun setUint16(offset: Int, value: Uint16, littleEndian: Boolean = true): Unit
    fun setInt32(offset: Int, value: Int, littleEndian: Boolean = true): Unit
    fun setUint32(offset: Int, value: Uint32, littleEndian: Boolean = true): Unit
    fun setFloat32(offset: Int, value: Float, littleEndian: Boolean = true): Unit
    fun setFloat64(offset: Int, value: Float64, littleEndian: Boolean = true): Unit
}
```
fun setInt32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally)
fun setUint32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally)
fun setFloat32(byteOffset: Int, value: Float, littleEndian: Boolean = definedExternally)
fun setFloat64(byteOffset: Int, value: Double, littleEndian: Boolean = definedExternally)

public external interface BufferDataSource

public external interface TexImageSource

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!/n
// See github.com/kotlin/dukat for details

package org.w3c.dom.clipboard

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

public external interface ClipboardEventInit : EventInit {
    var clipboardData: DataTransfer? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["clipboardData"] = clipboardData
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [ClipboardEvent](https://developer.mozilla.org/en/docs/Web/API/ClipboardEvent) to Kotlin
 */
public external open class ClipboardEvent(type: String, eventInitDict: ClipboardEventInit = definedExternally) : Event {
    open var clipboardData: DataTransfer?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

/**
 * Exposes the JavaScript [Clipboard](https://developer.mozilla.org/en/docs/Web/API/Clipboard) to Kotlin
 */
public external abstract class Clipboard : EventTarget {
    fun read(): Promise<DataTransfer>
    fun readText(): Promise<String>
    fun write(data: DataTransfer): Promise<Unit>
    fun writeText(data: String): Promise<Unit>
}

public external interface ClipboardPermissionDescriptor {
    var allowWithoutGesture: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ClipboardPermissionDescriptor(allowWithoutGesture: Boolean? = false): ClipboardPermissionDescriptor {
    val o = js("({})")
    o["allowWithoutGesture"] = allowWithoutGesture
    return o
}
abstract class CSSStyleSheet : StyleSheet {
    open val ownerRule: CSSRule?
    open val cssRules: CSSRuleList
    fun insertRule(rule: String, index: Int): Int
    fun deleteRule(index: Int)
}

/**
 * Exposes the JavaScript [StyleSheetList](https://developer.mozilla.org/en/docs/Web/API/StyleSheetList) to Kotlin
 */
public external abstract class StyleSheetList : ItemArrayLike<StyleSheet> {
    override fun item(index: Int): StyleSheet?
}

@kotlin.internal.InlineOnly
public inline operator fun StyleSheetList.get(index: Int): StyleSheet? = asDynamic()[index]

/**
 * Exposes the JavaScript [LinkStyle](https://developer.mozilla.org/en/docs/Web/API/LinkStyle) to Kotlin
 */
public external interface LinkStyle {
    val sheet: StyleSheet?
        get() = definedExternally
}

/**
 * Exposes the JavaScript [CSSRuleList](https://developer.mozilla.org/en/docs/Web/API/CSSRuleList) to Kotlin
 */
public external abstract class CSSRuleList : ItemArrayLike<CSSRule> {
    override fun item(index: Int): CSSRule?
}

@kotlin.internal.InlineOnly
public inline operator fun CSSRuleList.get(index: Int): CSSRule? = asDynamic()[index]

/**
 * Exposes the JavaScript [CSSRule](https://developer.mozilla.org/en/docs/Web/API/CSSRule) to Kotlin
 */
public external abstract class CSSRule {
    open val type: Short
    open var cssText: String
    open val parentRule: CSSRule?
    open val parentStyleSheet: CSSStyleSheet?

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSStyleRule](https://developer.mozilla.org/en/docs/Web/API/CSSStyleRule) to Kotlin
 */
public external abstract class CSSStyleRule : CSSRule {
    open var selectorText: String
    open val style: CSSStyleDeclaration

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

public external abstract class CSSImportRule : CSSRule {
    open val href: String
    open val media: MediaList
    open val styleSheet: CSSStyleSheet

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSGroupingRule](https://developer.mozilla.org/en/docs/Web/API/CSSGroupingRule) to Kotlin
 */
public external abstract class CSSGroupingRule : CSSRule {
    open val cssRules: CSSRuleList
    fun insertRule(rule: String, index: Int): Int
    fun deleteRule(index: Int)

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin
 */
public external abstract class CSSMediaRule : CSSGroupingRule {
    open val media: MediaList

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin
 */
public external abstract class CSSPageRule : CSSGroupingRule {
    open var selectorText: String
    open val style: CSSStyleDeclaration

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

public external abstract class CSSMarginRule : CSSRule {
    open val name: String
    open val style: CSSStyleDeclaration

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val NAMESPACE_RULE: Short
    }
}
Open Source Used In DNAC Platform Antelope  9574

/**
 * Exposes the JavaScript
 [CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin
 */
public external abstract class CSSNamespaceRule : CSSRule {
    open val namespaceURI: String
    open val prefix:

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript
 [CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin
 */
public external abstract class CSSStyleDeclaration : ItemArrayLike<String> {
    open var cssText: String
    open val parentRule: CSSRule?
    open var cssFloat: String
    open var alignContent: String
    open var alignItems:
    open var alignSelf: String
    open var animation: String
    open var animationDelay: String
    open var animationDirection: String
    open var animationDuration: String
    open var animationFillMode: String
    open var animationIterationCount: String
    open var animationName: String
    open var animationPlayState:
    open var animationTimingFunction: String
    open var backgroundPosition: String
    open var backgroundAttachment: String
    open var backgroundClip: String
    open var backgroundColor: String
    open var backgroundImage: String
    open var backgroundOrigin: String
    open var backgroundPosition: String
    open var backgroundSize: String
    open var border: String
    open var borderBottom: String
    open var borderBottomColor: String
    open var borderBottomLeftRadius: String
    open var borderBottomRightRadius: String
    open var borderBottomStyle:
    open var borderBottomWidth: String
    open var borderCollapse: String
    open var borderColor:
    open var borderImage: String
    open var borderImageOutset: String
    open var borderImageRepeat:
    open var borderImageSlice: String
    open var borderImageSource: String
    open var borderImageWidth: String
    open var borderLeft: String
    open var borderLeftColor: String
    open var borderLeftStyle:
    open var borderLeftWidth: String
    open var borderRight: String
    open var borderRightColor: String
    open var borderRightStyle:
    open var borderRightWidth: String
    open var borderTop: String
    open var borderTopColor: String
    open var borderTopLeftRadius: String
    open var borderTopRightRadius: String
    open var borderTopStyle:
    open var borderTopWidth: String
    open var bottom: String
    open var boxDecorationBreak: String
    open var boxSizing: String
    open var breakBefore: String
    open var breakInside: String
    open var captionSide: String
    open var clear: String
    open var clip:
    open var color: String
    open var columnCount: String
    open var columnFill: String
    open var columnGap: String
    open var columnRule: String
    open var columnRuleColor: String
    open var columnRuleStyle:
    open var columnRuleWidth:
    open var columns: String
    open var counterIncrement: String
    open var counterReset:
    open var cursor: String
    open var display:
    open var emptyCells: String
    open var filter:
    open var flex:
    open var flexBasis:
    open var flexDirection:
    open var flexGrow:
    open var flexShrink:
    open var flexWrap:
    open var font:
    open var fontFamily:
    open var fontFeatureSettings:
    open var fontLanguageOverride:
    open var fontStyle:
    open var fontStretch:
    open var fontStyle:
    open var fontSynthesis:
    open var fontWeight:
    open var fontVariant:
    open var fontVariantAlternates:
    open var fontVariantEastAsian:
    open var fontVariantLigatures:
    open var fontVariantNumeric:
    open var fontVariantPosition:
    open var fontWeight:
    open var hyphens:
    open var imageRendering:
    open var imageResolution:
    open var imeMode:
    open var justifyContent:
    open var letterSpacing:
    open var lineBreak:
    open var
org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

public external interface MediaKeySystemConfiguration {
    var label: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var initDataTypes: Array<String>? /* = arrayOf() */
        get() = definedExternally
        set(value) = definedExternally
    var audioCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() */
        get() = definedExternally
        set(value) = definedExternally
    var videoCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() */
        get() = definedExternally
        set(value) = definedExternally
    var distinctiveIdentifier: MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL */
        get() = definedExternally
        set(value) = definedExternally
    var persistentState: MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL */
        get() = definedExternally
        set(value) = definedExternally
    var sessionTypes: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["label"] = label
    o["initDataTypes"] = initDataTypes
    o["audioCapabilities"] = audioCapabilities
    o["videoCapabilities"] = videoCapabilities
    o["distinctiveIdentifier"] = distinctiveIdentifier
    o["persistentState"] = persistentState
    o["sessionTypes"] = sessionTypes
    return o
}

public external interface MediaKeySystemMediaCapability {
    var contentType: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var robustness: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun MediaKeySystemMediaCapability(contentType: String? = "", robustness: String? = ""): MediaKeySystemMediaCapability {
    val o = js("({})")
    o["contentType"] = contentType
    o["robustness"] = robustness
    return o
}

public external abstract class MediaKeySystemAccess {
    open val keySystem: String
    fun getConfiguration(): MediaKeySystemConfiguration
    fun createMediaKeys(): Promise<MediaKeys>
}

public external abstract class MediaKeys {
    fun createSession(sessionType: MediaKeySessionType = definedExternally): MediaKeySession
}

public external abstract class MediaKeySession : EventTarget {
    open val sessionId: String
    open val expiration: Double
    open val closed: Promise<Unit>
    open val keyStatuses: MediaKeyStatusMap
    open var onkeystatuseschange: ((Event) -> dynamic)?
    open var onmessage: ((MessageEvent) -> dynamic)?
    fun generateRequest(initDataType: String, initData: dynamic): Promise<Unit>
    fun load(sessionId: String): Promise<Boolean>
    fun update(response: dynamic): Promise<Unit>
    fun remove(): Promise<Unit>
    fun close(): Promise<Unit>
    fun remove(): Promise<Unit>
}

public external abstract class MediaKeyStatusMap {
    open val size: Int
    fun has(keyId: dynamic): Boolean
    fun get(keyId: dynamic): Any?
}

public external open class MediaKeyMessageEvent(type: String, eventInitDict: MediaKeyMessageEventInit) : Event {
    open val generateRequest(initDataType: String, initData: dynamic): Promise<Unit>
    fun load(sessionId: String): Promise<Unit>
    fun update(response: dynamic): Promise<Unit>
    fun remove(): Promise<Unit>
}

public external open class MediaEncryptedEvent(type: String, eventInitDict: MediaEncryptedEventInit) : Event {
    open val generateRequest(initDataType: String, initData: dynamic): Promise<Unit>
    fun load(sessionId: String): Promise<Unit>
    fun update(response: dynamic): Promise<Unit>
    fun remove(): Promise<Unit>
}

open class MediaKeyMessageEventInit {
    var messageType: MediaKeyMessageType
    var message: ArrayBuffer
    var companion object
}

open class MediaEncryptedEventInit {
    var messageType: MediaKeyMessageType
    var message: ArrayBuffer
    var companion object
}

open class MediaKeyStatusMap {
    open val size: Int
    fun has(keyId: dynamic): Boolean
    fun get(keyId: dynamic): Any?
}

open class MediaKeyMessageEvent {
    fun generateRequest(initDataType: String, initData: dynamic): Promise<Unit>
    fun load(sessionId: String): Promise<Unit>
    fun update(response: dynamic): Promise<Unit>
    fun remove(): Promise<Unit>
}

open class MediaEncryptedEvent {
    fun generateRequest(initDataType: String, initData: dynamic): Promise<Unit>
    fun load(sessionId: String): Promise<Unit>
    fun update(response: dynamic): Promise<Unit>
    fun remove(): Promise<Unit>
}

open class MediaKeyStatusMap {
    open val size: Int
    fun has(keyId: dynamic): Boolean
    fun get(keyId: dynamic): Any?
}

open class MediaKeyMessageEvent {
    fun generateRequest(initDataType: String, initData: dynamic): Promise<Unit>
    fun load(sessionId: String): Promise<Unit>
    fun update(response: dynamic): Promise<Unit>
    fun remove(): Promise<Unit>
}

open class MediaEncryptedEvent {
    fun generateRequest(initDataType: String, initData: dynamic): Promise<Unit>
    fun load(sessionId: String): Promise<Unit>
    fun update(response: dynamic): Promise<Unit>
    fun remove(): Promise<Unit>
}
eventInitDict: MediaEncryptedEventInit = definedExternally) : Event {
    open val initDataType: String
    open val initData: ArrayBuffer
    companion object
}

public external interface MediaEncryptedEventInit : Event {
    open val initDataType: String
    open val initData: ArrayBuffer?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface MediaEncryptedEventInit : EventInit {
    var initDataType: String? /* = "" */
        get() = definedExternally
    var initData: ArrayBuffer? /* = null */
        get() = definedExternally
    }

@kotlin.internal.InlineOnly
public inline fun MediaEncryptedEventInit(initDataType: String? = "", initData: ArrayBuffer? = null, bubbles: Boolean? = false,
    val o = js("({})")
    o["initDataType"] = initDataType
    o["initData"] = initData
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
} /* please, don't implement this interface! */

@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface MediaKeysRequirement {
    companion object
}


/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface MediaKeySessionType {
    companion object
}

public inline val MediaKeySessionType.Companion.TEMPORARY: MediaKeySessionType get() = "temporary".asDynamic().unsafeCast<MediaKeySessionType>()

public inline val MediaKeySessionType.Companion.PERSISTENT_LICENSE: MediaKeySessionType get() = "persistent-license".asDynamic().unsafeCast<MediaKeySessionType>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface MediaKeyStatus {
    companion object
}

public inline val MediaKeyStatus.Companion.USABLE: MediaKeyStatus get() = "usable".asDynamic().unsafeCast<MediaKeyStatus>()

public inline val MediaKeyStatus.Companion.EXPIRED: MediaKeyStatus get() = "expired".asDynamic().unsafeCast<MediaKeyStatus>()

public inline val MediaKeyStatus.Companion.RELEASED: MediaKeyStatus get() = "released".asDynamic().unsafeCast<MediaKeyStatus>()

public inline val MediaKeyStatus.Companion.OUTPUT_RESTRICTED: MediaKeyStatus get() = "output-restricted".asDynamic().unsafeCast<MediaKeyStatus>()

public inline val MediaKeyStatus.Companion.OUTPUT_DOWNSCALED: MediaKeyStatus get() = "output-downscaled".asDynamic().unsafeCast<MediaKeyStatus>()

public inline val MediaKeyStatus.Companion.STATUS_PENDING: MediaKeyStatus get() = "status-pending".asDynamic().unsafeCast<MediaKeyStatus>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface MediaKeyMessageType {
    companion object
}

public inline val MediaKeyMessageType.Companion.LICENSE_REQUEST: MediaKeyMessageType get() = "license-request".asDynamic().unsafeCast<MediaKeyMessageType>()

public inline val MediaKeyMessageType.Companion.LICENSE_RENEWAL: MediaKeyMessageType get() = "license-renewal".asDynamic().unsafeCast<MediaKeyMessageType>()

public inline val MediaKeyMessageType.Companion.LICENSE_RELEASE: MediaKeyMessageType get() = "license-release".asDynamic().unsafeCast<MediaKeyMessageType>()
package org.w3c.dom.events

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript [UIEvent](https://developer.mozilla.org/en/docs/Web/API/UIEvent) to Kotlin
 */
public external open class UIEvent(type: String, eventInitDict: UIEventInit = definedExternally) : Event {
    open var view: Window?
    open var detail: Int

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface UIEventInit : EventInit {
    var view: Window? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var detail: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [FocusEvent](https://developer.mozilla.org/en/docs/Web/API/FocusEvent) to Kotlin
 */
public external open class FocusEvent(type: String, eventInitDict: FocusEventInit = definedExternally) : UIEvent {
    open var relatedTarget: EventTarget?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface FocusEventInit : UIEventInit {
    var relatedTarget: EventTarget? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["relatedTarget"] = relatedTarget
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [MouseEvent](https://developer.mozilla.org/en/docs/Web/API/MouseEvent) to Kotlin
 */
public external open class MouseEvent(type: String, eventInitDict: MouseEventInit = definedExternally) : UIEvent, UnionElementOrMouseEvent {
    open var screenX: Int
    open var screenY: Int
    open var clientX: Int
    open var clientY: Int
    open var ctrlKey: Boolean
    open var shiftKey: Boolean
    open var altKey: Boolean
    open var metaKey: Boolean
    open var button: Short
    open var buttons: Short
    open var relatedTarget: EventTarget?
    open var region: String?
    open var pageX: Double
    open var pageY: Double
    open var offsetX: Double
    open var offsetY: Double
    fun getModifierState(keyArg: String): Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface MouseEventInit : EventModifierInit {
    var screenX: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var screenY: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var clientX: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var clientY: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var button: Short? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var buttons: Short? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var relatedTarget: EventTarget?
        get() = null
        set(value) = null
        var region: String? /* = null */
        get() = null
        set(value) = null
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["screenX"] = screenX
    o["screenY"] = screenY
    o["clientX"] = clientX
    o["clientY"] = clientY
    o["button"] = button
    o["buttons"] = buttons
    o["relatedTarget"] = relatedTarget
    o["region"] = region
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

public external interface EventModifierInit : UIEventInit {
    var ctrlKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var shiftKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var altKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var metaKey: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierAltGraph: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierCapsLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierFn: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierNumLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierScrollLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierSuper: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierSymbol: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var modifierSymbolLock: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["modifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
[WheelEvent](https://developer.mozilla.org/en/docs/Web/API/WheelEvent) to Kotlin
 */
public external open class WheelEvent(type: String, eventInitDict: WheelEventInit = definedExternally) : MouseEvent { /*...*/
```kotlin
* get() = definedExternally
    set(value) = definedExternally
    var code: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var location: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var repeat: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var isComposing: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally

@kotlin.internal.InlineOnly
    isComposing: Boolean? = false, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false,
    metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false,
    modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false,
    modifierNumLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbolLock: Boolean? = false,
    val o = js("({})")
    o["key"] = key
    o["code"] = code
    o["location"] = location
    o["repeat"] = repeat
    o["isComposing"] = isComposing
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierFnLock"] = modifierFnLock
    o["modifierHyper"] = modifierHyper
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 */
public external open class CompositionEvent(type: String, eventInitDict: CompositionEventInit = definedExternally) :
    UIEvent {
    open val data: String

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface CompositionEventInit : UIEventInit {
    var data: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun CompositionEventInit(data: String? = "", view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false,
    cancelable: Boolean? = false, composed: Boolean? = false): CompositionEventInit {
    val o = js("({})")
    o["data"] = data
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 */
public external open class Event(type: String, eventInitDict: EventInit = definedExternally) {
    open val type: String
    open val target:
    open val currentTarget: EventTarget?
    open val eventPhase: Short
    open val bubbles: Boolean
    open val cancelable: Boolean
    open val isTrusted: Boolean
    open val timeStamp: Number
    fun composedPath(): Array<EventTarget>
    fun stopPropagation(): Unit
    fun stopImmediatePropagation(): Unit
    fun preventDefault(): Unit
    fun initEvent(type: String, bubbles: Boolean, cancelable: Boolean) {
    val o = js("({})")
    o["data"] = data
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 */
public external abstract class EventTarget {
    fun addEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)
    fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)
    fun removeEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)
    fun removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)
    fun dispatchEvent(event: Event): Boolean
}

/**
 * Exposes the JavaScript
 */
public external interface EventListener {
    fun handleEvent(event: Event)
}
**Exposes the JavaScript [DOMStringMap](https://developer.mozilla.org/en/docs/Web/API/DOMStringMap) to Kotlin
*/
public external abstract class DOMStringMap

@kotlin.internal.InlineOnly
public inline operator fun DOMStringMap.get(name: String): String? = asDynamic()[name]

@kotlin.internal.InlineOnly
public inline operator fun DOMStringMap.set(name: String, value: String) { asDynamic()[name] = value }

/**
 * Exposes the JavaScript [HTMLHtmlElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHtmlElement) to Kotlin
 */
public external abstract class HTMLHtmlElement : HTMLElement {
    open var version: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLHeadElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadElement) to Kotlin
 */
public external abstract class HTMLHeadElement : HTMLElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTitleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTitleElement) to Kotlin
 */
public external abstract class HTMLTitleElement : HTMLElement {
    open var text: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLBaseElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBaseElement) to Kotlin
 */
public external abstract class HTMLBaseElement : HTMLElement {
    open var href: String
    open var target: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
Open Source Used In DNAC Platform Antelope

```kotlin
COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLLinkElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLinkElement) to Kotlin
 */
public external abstract class HTMLLinkElement : HTMLElement, LinkStyle {
    open var href: String
    open var crossOrigin: String?
    open var rel: String
    open var `as`: RequestDestination
    open val relList: DOMTokenList
    open var media: String
    open var nonce: String
    open var hreflang: String
    open var type: String
    open var rev: String
    open var target: String
    open var scope: String
    open var workerType: WorkerType

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLMetaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMetaElement) to Kotlin
 */
public external abstract class HTMLMetaElement : HTMLElement {
    open var name: String
    open var httpEquiv: String
    open var content: String
    open var scheme: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript [HTMLStyleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLStyleElement) to Kotlin
 */
public external abstract class HTMLStyleElement : HTMLElement, LinkStyle {
    open var media: String
    open var nonce: String
    open var type: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript [HTMLBodyElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBodyElement) to Kotlin
 */
public external abstract class HTMLBodyElement : HTMLElement, WindowEventHandlers {
    open var text: String
```
EXECUTIVE SUMMARY

The Executive Summary provides an overview of the key findings and recommendations from the project. It highlights the main objectives, the methodology used, and the key outcomes.

1. BACKGROUND

The Background section sets the stage for the project by providing contextual information and the rationale behind the study. It includes a brief history of the issue, relevant literature, and the specific context in which the project was undertaken.

2. METHODOLOGY

The methodology section details the approach used to conduct the research. It describes the methods employed, such as data collection techniques, sample size, and statistical analysis.

3. RESULTS

The Results section presents the findings of the study. It includes the data collected, the analysis performed, and the conclusions drawn from the data.

4. DISCUSSION

The Discussion section interprets the results, compares them with previous findings, and addresses the implications of the study. It also highlights any limitations of the research and suggests areas for future investigation.

5. CONCLUSIONS

The Conclusions section summarizes the key findings and their implications. It provides recommendations for future research and highlights the potential impact of the study on the field.

6. REFERENCES

The References section lists all the sources cited in the report. It follows a standardized format to ensure credibility and provides a comprehensive view of the literature relevant to the study.

7. APPENDICES

The Appendices section contains supplementary material, such as additional data, detailed tables, or raw data sets, which support the main findings of the report.
DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/\
* Exposes the JavaScript [HTMLTimeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTimeElement) to Kotlin
*/
public external abstract class HTMLTimeElement : HTMLElement {
    open var dateTime: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/\
* Exposes the JavaScript [HTMLSpanElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSpanElement) to Kotlin
*/
public external abstract class HTMLSpanElement : HTMLElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/\
* Exposes the JavaScript [HTMLBRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBRElement) to Kotlin
*/
public external abstract class HTMLBRElement : HTMLElement {
    open var clear: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/\
* Exposes the JavaScript [HTMLHyperlinkElementUtils](https://developer.mozilla.org/en/docs/Web/API/HTMLHyperlinkElementUtils) to Kotlin
*/
public external interface HTMLHyperlinkElementUtils {
    var href: String
    val origin: String
    var protocol: String
    var username: String
    var password: String
    var host: String
    var hostname: String
    var port: String
    var pathname: String
    var search: String
    var hash: String
}

/\
* Exposes the JavaScript [HTMLModElement](https://developer.mozilla.org/en/docs/Web/API/HTMLModElement) to Kotlin
*/
public external abstract class HTMLModElement : HTMLElement {
    open var cite: String
    open var dateTime: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
/**
 * Exposes the JavaScript [HTMLPictureElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPictureElement) to Kotlin
 */
public external abstract class HTMLPictureElement : HTMLElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLSourceElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSourceElement) to Kotlin
 */
public external abstract class HTMLSourceElement : HTMLElement {
    open var src: String
    open var type: String
    open var srcset: String
    open var sizes: String
    open var media: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLImageElement](https://developer.mozilla.org/en/docs/Web/API/HTMLImageElement) to Kotlin
 */
public external abstract class HTMLImageElement : HTMLElement, HTMLOrSVGImageElement, TexImageSource {
    open var alt: String
    open var src: String
    open var srcset: String
    open var sizes: String
    open var crossOrigin: String?
    open var useMap: String
    open var isMap: Boolean
    open var width: Int
    open var height: Int
    open val naturalWidth: Int
    open val naturalHeight: Int
    open val complete: Boolean
    open val currentSrc: String
    open var referrerPolicy: String
    open var name: String
    open var lowsrc: String
    open var align: String
    open var hspace: Int
    open var vspace: Int
    open var longDesc: String
    open var border: String
    open val x: Int
    open val y: Int

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLIFrameElement](https://developer.mozilla.org/en/docs/Web/API/HTMLIFrameElement) to Kotlin
 */
public external abstract class HTMLIFrameElement : HTMLElement {
    open var src: String
    open var
srcdoc: String
open var name: String
open var sandbox: DOMTokenList
open var allowFullscreen: Boolean
open var allowUserMedia: Boolean
open var width: String
open var height: String
open var allowReferrerPolicy: String
open val contentDocument: Document?
open val contentWindow: Window?
on open var align: String
open var scrolling: String
open var frameBorder: String
open var longDesc: String
open var marginHeight: String
open var marginWidth: String
fun getSVGDocument(): Document? companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLEmbedElement](https://developer.mozilla.org/en/docs/Web/API/HTMLEmbedElement) to Kotlin
 */
public external abstract class HTMLEmbedElement : HTMLElement {
open var src: String
open var type: String
open var width: String
open var height: String
open var align: String
open var name: String
fun getSVGDocument(): Document?

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLObjectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLObjectElement) to Kotlin
 */
public external abstract class HTMLObjectElement : HTMLElement {
open var data: String
open var type: String
open var typeMustMatch: Boolean
open var name: String
open var useMap: String
open val form: HTMLFormElement?
open var width: String
open var height: String
open val contentDocument: Document?
open val contentWindow: Window?
open val willValidate: Boolean
open val validity: ValidityState
open val validationMessage: String
open var align: String
open var archive: String
open var code: String
open var declare: Boolean
open var hspace: Int
open var standby: String
open var vspace: Int
open var codeBase: String
open var codeType: String
open var border: String
fun getSVGDocument(): Document?
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLParamElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParamElement) to Kotlin
 */
public external abstract class HTMLParamElement : HTMLElement {
open var name: String
open var

value: String
open var type: String
open var value

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLVideoElement](https://developer.mozilla.org/en/docs/Web/API/HTMLVideoElement) to Kotlin
 */
public external abstract class HTMLVideoElement : HTMLMediaElement, CanvasImageSource, TexImageSource {
open var width: Int
open var height: Int
open val videoWidth: Int
open val videoHeight: Int
open var poster: String
open var playsInline: Boolean

companion object {
val NETWORK_EMPTY: Short
val NETWORK_IDLE: Short
val NETWORK_LOADING: Short
val NETWORK_NO_SOURCE: Short
val HAVE_NOTHING: Short
val HAVE_METADATA: Short
val HAVE_CURRENT_DATA: Short
val HAVE_FUTURE_DATA: Short
val HAVE_ENOUGH_DATA: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLAudioElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAudioElement) to Kotlin
 */
public external abstract class HTMLAudioElement : HTMLMediaElement {

companion object {
val NETWORK_EMPTY: Short
val NETWORK_IDLE: Short
val NETWORK_LOADING: Short
val NETWORK_NO_SOURCE: Short
val HAVE_NOTHING: Short
val HAVE_METADATA: Short
val HAVE_CURRENT_DATA: Short
val HAVE_FUTURE_DATA: Short
val HAVE_ENOUGH_DATA: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLTrackElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTrackElement) to Kotlin
 */
public external abstract class HTMLTrackElement : HTMLElement {
open var kind: String
open var src: String
open var srclang: String
open var default: Boolean
open val readyState: Short
open val track: TextTrack

companion object {
val NONE: Short
val LOADING: Short
val LOADED: Short
val ERROR: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

Open Source Used In DNAC Platform Antelope 9591
/**
 * Exposes the JavaScript [HTMLMediaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement) to Kotlin
 */
public external abstract class HTMLMediaElement : HTMLElement {
    open val error: MediaError?
    open var src: String
    open var srcObject: MediaProvider?
    open val currentSrc: String
    open var crossOrigin: String?
    open val networkState: Short
    open var preload: String
    open val buffered: TimeRanges
    open val readyState: Short
    open var seeking: Boolean
    open var currentTime: Double
    open val duration: Double
    open var paused: Boolean
    open var defaultPlaybackRate: Double
    open var loop: Boolean
    open var controls: Boolean
    open var volume: Double
    open var muted: Boolean
    open var defaultMuted: Boolean
    open var onencrypted: (Event) -> dynamic
    open var onwaitingforkey: (Event) -> dynamic
    fun load()
    fun canPlayType(type: String): CanPlayTypeResult
    fun fastSeek(time: Double)
    fun getStartDate(): dynamic
    fun play(): Promise<Unit>
    fun pause()
    fun addTextTrack(kind: TextTrackKind, label: String = definedExternally, language: String = definedExternally): TextTrack
    fun setMediaKeys(mediaKeys: MediaKeys?): Promise<Unit>

    companion object {
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [MediaError](https://developer.mozilla.org/en/docs/Web/API/MediaError) to Kotlin
 */
public external abstract class MediaError {
    open val code: Short

    companion object {
        val MEDIA_ERR_ABORTED: Short
        val MEDIA_ERR_NETWORK: Short
        val MEDIA_ERR_DECODE: Short
        val MEDIA_ERR_SRC_NOT_SUPPORTED: Short
    }
}

/**
 * Exposes the JavaScript [AudioTrackList](https://developer.mozilla.org/en/docs/Web/API/AudioTrackList) to Kotlin
 */
public external abstract class AudioTrackList : EventTarget {
    open val length: Int
    open var onchange: (Event) -> dynamic
    open var onaddtrack: (TrackEvent) -> dynamic
    open var onremovetrack: (TrackEvent) -> dynamic
    fun getTrackById(id: String): AudioTrack?
}

@kotlin.internal.InlineOnly
public inline operator fun AudioTrackList.get(index: Int): AudioTrack? = asDynamic()[index]

/**
 * Exposes the JavaScript [AudioTrack](https://developer.mozilla.org/en/docs/Web/API/AudioTrack) to Kotlin
 */
public external abstract class AudioTrack : UnionAudioTrackOrTextTrackOrVideoTrack {
    open val id: String
    open val kind: String
    open val label: String
    open val language: String
    open var enabled: Boolean
    open var sourceBuffer: SourceBuffer?
}

/**
 * Exposes the JavaScript [VideoTrackList](https://developer.mozilla.org/en/docs/Web/API/VideoTrackList) to Kotlin
 */
public external abstract class VideoTrackList : EventTarget {
    open val length: Int
    open val selectedIndex: Int
    open var

val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAreaElement) to Kotlin

open var coords: String
open var var shape: String
open var target: String
open var download: String
open var ping: String
open var rel: String
open val relList: DOMTokenList
open var referrerPolicy: String
open var noHref: Boolean

val ELEMENT_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short

* Exposes the JavaScript [HTMLTableElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableElement) to Kotlin

open var caption: String
open var tHead: HTMLTableSectionElement
open var tBodies: HTMLCollection
open var rows: HTMLCollection
open var align: String
open var border: String
open var frame: String
open var rules: String
open var summary: String
open var width: String
open var bgColor: String
open var cellPadding: String
open var cellSpacing: String
fun createCaption(): HTMLTableCaptionElement
fun deleteCaption(): HTMLTableCaptionElement
fun createTHead(): HTMLTableSectionElement
fun deleteTHead(): HTMLTableSectionElement
fun createTFoot(): HTMLTableSectionElement
fun deleteTFoot(): HTMLTableSectionElement
fun insertRow(index: Int = definedExternally): HTMLTableRowElement
fun deleteRow(index: Int)

* Exposes the JavaScript [HTMLTableCaptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCaptionElement) to Kotlin

open var span: Int

align: String
open var ch: String
open var chOff: String
open var vAlign: String
open var width:

String

companion object {

val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [HTMLTableSectionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableSectionElement) to Kotlin
 */
public external abstract class HTMLTableSectionElement : HTMLElement {
    open val rows: HTMLCollection
    open var align: String
    open var ch: String
    open var chOff: String
    open var vAlign: String
    fun insertRow(index: Int = definedExternally): HTMLElement
    fun deleteRow(index: Int)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTableRowElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableRowElement) to Kotlin
 */
public external abstract class HTMLTableRowElement : HTMLElement {
    open val rowIndex: Int
    open val sectionRowIndex: Int
    open val cells: HTMLCollection
    open var align: String
    open var ch: String
    open var chOff: String
    open var vAlign: String
    open var bgColor: String
    fun insertCell(index: Int = definedExternally): HTMLElement
    fun deleteCell(index: Int)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTableCellElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCellElement) to Kotlin
 */
public external abstract class HTMLTableCellElement : HTMLElement {
    open var colSpan: Int
    open var rowSpan: Int
    open var headers: String
    open val cellIndex: Int
    open var scope: String
    open var abbr: String
    open var align: String
    open var axis: String
    open var height: String
    open var width: String
    open var ch: String
    open var chOff: String
    open var noWrap: Boolean
    open var vAlign: String
    open var bgColor: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```kotlin
/**
 * Exposes the JavaScript [HTMLFormElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFormElement) to Kotlin
 */
public external abstract class HTMLFormElement : HTMLElement {
    open var acceptCharset: String
    open var action: String
    open var autocomplete: String
    open var enctype: String
    open var encoding: String
    open var method: String
    open var name: String
    open var noValidate: Boolean
    open var target: String
    open val elements: HTMLFormControlsCollection
    open val length: Int
    fun submit()
    fun reset()
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

@kotlin.internal.InlineOnly
public inline operator fun HTMLFormElement.get(index: Int): Element? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun HTMLFormElement.get(name: String): UnionElementOrRadioNodeList? = asDynamic()[name]

/**
 * Exposes the JavaScript [HTMLLabelElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLabelElement) to Kotlin
 */
public external abstract class HTMLLabelElement : HTMLElement {
    open var htmlFor: String
    open val control: HTMLElement?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to Kotlin
 */
public external abstract class HTMLInputElement : HTMLElement {
    open var accept: String
    open var alt: String
    open var autocomplete: String
    open var autofocus: Boolean
    open var defaultChecked: Boolean
    open var checked: Boolean
    open var dirName: String
    open var disabled: Boolean
    open var form: HTMLFormElement?
    open var files: FileList?
    open var formAction: String
    open var formEnctype: String
    open var formMethod: String
    open var formNoValidate: Boolean
    open var formTarget: String
    open var height: Int
    open var indeterminate: Boolean
    open var inputMode: String
    open var list: HTMLElement?
    open var max: String
    open var maxLength: Int
    open var min: String
    open var minLength: Int
    open var multiple: Boolean
    open var name: String
    open var pattern: String
    open var placeholder: String
    open var readOnly: Boolean
    open var required: Boolean
    open var required: Boolean
    open var size: Int
    open var step: String
    open var type: String
    open var value: String
    open var valueAsDate: dynamic
    open var valueAsNumber: Double
    open var width: Int
    open var willValidate: Boolean
    open var validationMessage: String
    open var labels: NodeList
    open var selectionStart: Int
    open var selectionEnd: Int
    open var
```
selectionDirection: String
open var align: String
open var useMap: String
fun stepUp(n: Int = definedExternally)
fun stepDown(n: Int = definedExternally)
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)
fun select():
fun setRangeText(replacement: String)
fun setRangeText(replacement: String, start: Int, end: Int, selectionMode: SelectionMode = definedExternally)
fun setSelectionRange(start: Int, end: Int, direction: String = definedExternally)

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLButtonElement](https://developer.mozilla.org/en/docs/Web/API/HTMLButtonElement) to Kotlin
 */
public external abstract class HTMLButtonElement : HTMLElement {
open var autofocus: Boolean
open var disabled: Boolean
open val form: HTMLFormElement?
open var formAction: String
open var formEnctype: String
open var formMethod: String
open var formNoValidate: Boolean
open var formTarget: String
open var name: String
open var type: String
open var value: String
open var menu: HTMLMenuElement?
open var willValidate: Boolean
open var validity: ValidityState
open var validationMessage: String
open val labels: NodeList
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLSelectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSelectElement) to Kotlin
 */
public external abstract class HTMLSelectElement : HTMLElement, ItemArrayLike<Element> {
open var autocomplete: String
open var autofocus: Boolean
open var disabled: Boolean
open val form: HTMLFormElement?
open var multiple: Boolean
open var name: String
open var required: Boolean
open var size: Int
open var type: String
open val options: HTMLOptionsCollection
override var length: Int
open val selectedOptions: HTMLCollection
open var selectedIndex: Int
open var value: String
open val willValidate: Boolean
open val validity: ValidityState
open val validationMessage: String
open val labels: NodeList
fun namedItem(name: String): HTMLOptionElement?
fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)
fun remove(index: Int)
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)
override fun item(index: Int): Element?
override fun item(index: Int): Element?

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

Open Source Used In DNAC Platform Antelope 9597
/**
 * Exposes the JavaScript [HTMLProgressElement](https://developer.mozilla.org/en/docs/Web/API/HTMLProgressElement) to Kotlin
 *
public external abstract class HTMLProgressElement : HTMLElement {
    open var value: Double
    open var max: Double
    open val position: Double
    open val labels: NodeList

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
    }
}

/**
 * Exposes the JavaScript [HTMLMeterElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMeterElement) to Kotlin
 *
public external abstract class HTMLMeterElement : HTMLElement {
    open var value: Double
    open var min: Double
    open var max: Double
    open var low: Double
    open var high: Double
    open var optimum: Double
    open val labels: NodeList

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
    }
}

/**
 * Exposes the JavaScript [HTMLFieldSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFieldSetElement) to Kotlin
 *
public external abstract class HTMLFieldSetElement : HTMLElement {
    open var disabled: Boolean
    open val form: HTMLFormElement
    open val name: String
    open val type: String
    open val elements: HTMLCollection
    open val willValidate: Boolean
    open val validity: ValidityState
    open val validationMessage: String
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
    }
}

/**
 * Exposes the JavaScript [HTMLLegendElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLegendElement) to Kotlin
 *
public external abstract class HTMLLegendElement : HTMLElement {
    open val form: HTMLFormElement
    open var align: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
    }
}
DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [ValidityState](https://developer.mozilla.org/en/docs/Web/API/ValidityState) to Kotlin
 */
public external abstract class ValidityState {
    open val valueMissing: Boolean
    open val typeMismatch: Boolean
    open val patternMismatch: Boolean
    open val tooLong: Boolean
    open val tooShort: Boolean
    open val rangeUnderflow: Boolean
    open val rangeOverflow: Boolean
    open val stepMismatch: Boolean
    open val badInput: Boolean
    open val customError: Boolean
    open val valid: Boolean
}

/**
 * Exposes the JavaScript [HTMLDetailsElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDetailsElement) to Kotlin
 */
public external abstract class HTMLDetailsElement : HTMLElement {
    open var open: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class HTMLMenuElement : HTMLElement {
    open var type: String
    open var label: String
    open var compact: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class HTMLMenuItemElement : HTMLElement {
    open var type: String
    open var label: String
    open var icon: String
    open var disabled: Boolean
    open var checked: Boolean
    open var radiogroup: String
    open var default: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external open class RelatedEvent(type: String, eventInitDict: RelatedEventInit = definedExternally) : Event {
    open val relatedTarget: EventTarget?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface RelatedEventInit : EventInit {
    var relatedTarget: EventTarget? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun RelatedEventInit(relatedTarget: EventTarget?): RelatedEventInit? = RelatedEventInit(relatedTarget)
RelatedEventInit {
    new val o = js("()")
    o["relatedTarget"] = relatedTarget
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/* Exposes the JavaScript [HTMLDialogElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDialogElement) to Kotlin */
public external abstract class HTMLDialogElement : HTMLElement {
    open var open: Boolean
    open var returnValue: String
    fun show(anchor: UnionElementOrMouseEvent = definedExternally)
    fun showModal(anchor: UnionElementOrMouseEvent = definedExternally)
    fun close(returnValue: String = definedExternally)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/* Exposes the JavaScript [HTMLScriptElement](https://developer.mozilla.org/en/docs/Web/API/HTMLScriptElement) to Kotlin */
public external abstract class HTMLScriptElement : HTMLElement, HTMLOrSVGScriptElement {
    open var src: String
    open var type: String
    open var charset: String
    open var async: Boolean
    open var defer: Boolean
    open var crossOrigin: String?
    open var text: String
    open var nonce: String
    open var event: String
    open var htmlFor: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/* Exposes the JavaScript [HTMLTemplateElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTemplateElement) to Kotlin */
public external abstract class HTMLTemplateElement : HTMLElement {
    open val content: DocumentFragment

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/* Exposes the JavaScript [HTMLSlotElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSlotElement) to Kotlin */
public external abstract class HTMLSlotElement : HTMLElement {
    open var name: String
    fun assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION Node: Short
    }
}
null
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

public external interface AssignedNodesOptions {
    var flatten: Boolean? /* = false */ get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun AssignedNodesOptions(flatten: Boolean? = false): AssignedNodesOptions {
    val o = js("({})")
    o["flatten"] = flatten
    return o
}

/**
 * Exposes the JavaScript [HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement) to Kotlin
 */
public external abstract class HTMLCanvasElement : HTMLElement, CanvasImageSource, TexImageSource {
    open var width: Int
    open var height: Int
    fun getContext(contextId: String, vararg arguments: Any?): RenderingContext?
    fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally): String
    fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? = definedExternally): void

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface CanvasRenderingContext2DSettings {
    var alpha: Boolean? /* = true */ get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {
    val o = js("({})")
    o["alpha"] = alpha
    return o
}

/**
 * Exposes the JavaScript [CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D) to Kotlin
 */
public external abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform, CanvasCompositing, CanvasImageSmoothing, CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters, CanvasRect, CanvasDrawPath, CanvasUserInterface, CanvasText, CanvasDrawImage, CanvasHitRegion, CanvasImageData, CanvasPathDrawingStyles, CanvasTextDrawingStyles, CanvasPath, RenderingContext {
    open val canvas: HTMLCanvasElement
}

public external interface CanvasState {
    fun save()
    fun restore()
}

public external interface CanvasTransform {
    fun scale(x: Double, y: Double)
    fun rotate(angle: Double)
    fun translate(x: Double, y: Double)
    fun transform(a: Double, b: Double, c: Double, d: Double, e: Double, f: Double)
    fun resetTransform()
}

public external interface CanvasCompositing {
    var globalAlpha: Double
    var globalCompositeOperation: String
}

public external interface CanvasImageSmoothing {
    var imageSmoothingEnabled: Boolean
    var imageSmoothingQuality: Number
}

public external interface CanvasFillStrokeStyles {
    var strokeStyle: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var fillStyle: dynamic
        get() = definedExternally
        set(value) = definedExternally
    fun createLinearGradient(x0: Double, y0: Double, x1: Double, y1: Double): CanvasGradient
    fun createRadialGradient(x0: Double, y0: Double, r0: Double, x1: Double, y1: Double, r1: Double): CanvasGradient
    fun createPattern(image: CanvasImageSource, repetition: String): CanvasPattern
}

Open Source Used In DNAC Platform Antelope 9603
*\nget() = definedExternally
set(value) = definedExternally
var fillRule: CanvasFillRule? /* = CanvasFillRule.NONZERO */
*\nget() = definedExternally
set(value) = definedExternally
var id: String? /* = "" */
*\nget() = definedExternally
set(value) = definedExternally
var parentID: String? /* = null */
*\nget() = definedExternally
set(value) = definedExternally
var role: String? /* = null */

*\nget() = definedExternally
set(value) = definedExternally
var cursor: String? /* = "inherit" */

*\nget() = definedExternally
set(value) = definedExternally
var control: Element? /* = null */

*\nget() = definedExternally
set(value) = definedExternally
var label: String? /* = null */

    val o = js("{}").`function`
    o["path"] = path
    o["fillRule"] = fillRule
    o["id"] = id
    o["parentID"] = parentID
    o["cursor"] = cursor
    o["control"] = control
    o["label"] = label
    return o
}/**
 * Exposes the JavaScript [ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to Kotlin
 */
public external open class ImageData : ImageBitmapSource, TexImageSource {
    constructor(sw: Int, sh: Int)
    constructor(data: Uint8ClampedArray, sw: Int, sh: Int)
    open val width: Int
    open val height: Int
    open val data: Uint8ClampedArray
}

Path2D(): CanvasPath {

    constructor(path: Path2D)
    constructor(paths: Array<Path2D>, fillRule: CanvasFillRule = CanvasFillRule.NONZERO)
    constructor(d: String)
    fun addPath(path: Path2D, transform: dynamic = definedExternally)
    override fun closePath()
    override fun moveTo(x: Double, y: Double)
    override fun lineTo(x: Double, y: Double)
    override fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)
    override fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)
    override fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)
    override fun rect(x: Double, y: Double, w: Double, h: Double)
    override fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally */)
null

setDragImage(image: Element, x: Int, y: Int) { fun getData(format: String): String

fun clearData(format: String = definedExternally) { un/** */

* Exposes the JavaScript

[DataTransferItemList](https://developer.mozilla.org/en/docs/Web/API/DataTransferItemList) to Kotlin

* /unnecessary

external abstract class DataTransferItemList { un open val length: Int

fun add(data: String, type: String): DataTransferItem?

fun add(data: File): DataTransferItem?

fun remove(index: Int): Int

clear() { un

@kotlin.internal.InlineOnly

public

external abstract class DataTransferItemList { un number

fun add(data: String, type: String): DataTransferItem?

fun add(data: File): DataTransferItem?

fun remove(index: Int): Int

clear() { un

/**
 * Exposes the JavaScript

[DataTransferItem](https://developer.mozilla.org/en/docs/Web/API/DataTransferItem) to Kotlin

* /unnecessary

external abstract class DataTransferItem { un number

open val kind: String

open val type: String

getAsString(_callback: ((String) -> Unit)?)

getAsFile(): File?

/**
 * Exposes the JavaScript

[DragEvent](https://developer.mozilla.org/en/docs/Web/API/DragEvent) to Kotlin

* /unnecessary

open class DragEvent(type: String, eventInitDict: DragEventInit = definedExternally) : MouseEvent {

open val dataTransfer: DataTransfer?

companion object {

val NONE: Short

val CAPTURING_PHASE: Short

val AT_TARGET: Short

val BUBBLING_PHASE: Short

open interface DragEventInit : MouseEventInit {

var dataTransfer: DataTransfer? /* = null */

set(value) = definedExternally

get() = definedExternally

un */

un

un

un

un

un

un

un

un

un

un

un

un

un

un
fun alert(message: String)
fun confirm(message: String = definedExternally): Boolean
fun prompt(message: String?, targetOrigin: String, transfer: Array<dynamic> = definedExternally): String?
fun requestAnimationFrame(callback: (Double) -> Unit): Int
fun cancelAnimationFrame(handle: Int)
fun captureEvents(): Any?
fun releaseEvents(): Void
fun matchMedia(query: String): MediaQueryList
fun moveTo(x: Int, y: Int)
fun moveBy(x: Int, y: Int)
fun resizeTo(x: Int, y: Int)
fun resizeBy(x: Int, y: Int)
fun scroll(options: ScrollToOptions = definedExternally): Void
fun scroll(x: Double, y: Double)
fun scrollBy(options: ScrollToOptions = definedExternally)
fun scrollBy(x: Double, y: Double)
fun getComputedStyle(elt: Element, pseudoElt: String? = definedExternally): CSSStyleDeclaration

@kotlin.internal.InlineOnly
public inline operator fun Window.get(name: String): dynamic = asDynamic()[name]

public external abstract class BarProp {
    open val visible: Boolean
}

/**
 * Exposes the JavaScript [History](https://developer.mozilla.org/en/docs/Web/API/History) to Kotlin
 */
public external abstract class History {
    open val length: Int
    open var scrollRestoration: ScrollRestoration
    open val state: Any?
    fun go(delta: Int = definedExternally)
    fun back()
    fun forward()
    fun pushState(data: Any?, title: String, url: String? = definedExternally)
    fun replaceState(data: Any?, title: String, url: String? = definedExternally)
}

/**
 * Exposes the JavaScript [Location](https://developer.mozilla.org/en/docs/Web/API/Location) to Kotlin
 */
public external abstract class Location {
    open var href: String
    open val origin: String
    open var protocol: String
    open var host: String
    open var hostname: String
    open var port: String
    open var pathname: String
    open var search: String
    open var hash: String
    open val ancestorOrigins: Array<out String>
    fun assign(url: String)
    fun replace(url: String)
    fun reload()
}

/**
 * Exposes the JavaScript [PopStateEvent](https://developer.mozilla.org/en/docs/Web/API/PopStateEvent) to Kotlin
 */
public external open class PopStateEvent(type: String, eventInitDict: PopStateEventInit = definedExternally) : Event {
    open val state: Any? /* = null */

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface PopStateEventInit : EventInit {
    var state: Any? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["state"] = state
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [HashChangeEvent](https://developer.mozilla.org/en/docs/Web/API/HashChangeEvent) to Kotlin
 */
public external open class HashChangeEvent(type: String, eventInitDict: HashChangeEventInit = definedExternally) : Event {
    open val oldURL: String
    open val newURL: String

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface HashChangeEventInit : EventInit {
    var oldURL: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var newURL: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["oldURL"] = oldURL
    o["newURL"] = newURL
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [PageTransitionEvent](https://developer.mozilla.org/en/docs/Web/API/PageTransitionEvent) to Kotlin
 */
public external open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) : Event {
    open val persisted: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface PageTransitionEventInit : EventInit {
    var persisted: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun PageTransitionEventInit(persisted: Boolean? = false): PageTransitionEventInit {
    val o = js("({})")
    o["persisted"] = persisted
    return o
}
to Kotlin

```kotlin
fun btoa(data: String): String
fun atob(data: String): String
fun setTimeout(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int
fun clearTimeout(handle: Int = definedExternally)
fun setInterval(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int
fun createImageBitmap(image: ImageBitmapSource, options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>
fun fetch(input: dynamic, init: RequestInit = definedExternally): Promise<Response>
```

```kotlin
fun requestMediaKeySystemAccess(keySystem: String, supportedConfigurations: Array<MediaKeySystemConfiguration>): Promise<MediaKeySystemAccess>
fun getUserMedia(constraints: MediaStreamConstraints, successCallback: (MediaStream) -> Unit, errorCallback: (dynamic) -> Unit)
fun vibrate(pattern: dynamic): Boolean
```

```kotlin
fun registerProtocolHandler(scheme: String, url: String, title: String)
fun registerContentHandler(mimeType: String, url: String, title: String)
fun isProtocolHandlerRegistered(scheme: String, url: String): String
fun isContentHandlerRegistered(mimeType: String, url: String): String
fun unregisterProtocolHandler(scheme: String, url: String)
fun unregisterContentHandler(mimeType: String, url: String)
```

```kotlin
fun javaEnabled(): Boolean
```

```kotlin
fun refresh(reload: Boolean = definedExternally)
```

```kotlin
fun override fun item(index: Int): Plugin?
fun override fun item(name: String): Plugin?
```

```kotlin
fun PluginArray.refresh(reload: Boolean = definedExternally)
```

```kotlin
fun PluginArray.get(index: Int): Plugin?
fun PluginArray.get(name: String): Plugin?
```

```kotlin
fun PluginArray.get(index: Int): Plugin?
```
MimeType? = asDynamic()[name]

/**
 * Exposes the JavaScript
[MimeType](https://developer.mozilla.org/en/docs/Web/API/MimeType) to Kotlin
 */

abstract class MimeType {
    open val type: String
    open val description: String
    open val suffixes: String
    open val enabledPlugin: Plugin
}

/**
 * Exposes the JavaScript
[ImageBitmap](https://developer.mozilla.org/en/docs/Web/API/ImageBitmap) to Kotlin
 */

abstract class ImageBitmap : CanvasImageSource, TexImageSource {
    open val width: Int
    open val height: Int
    fun close()
}

interface ImageBitmapOptions {
    var imageOrientation: ImageOrientation? /* = ImageOrientation.NONE */
    var premultiplyAlpha: PremultiplyAlpha? /* = PremultiplyAlpha.DEFAULT */
    var colorSpaceConversion: ColorSpaceConversion? /* = ColorSpaceConversion.DEFAULT */
    var resizeWidth: Int
    var resizeHeight: Int
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["imageOrientation"] = imageOrientation
    o["premultiplyAlpha"] = premultiplyAlpha
    o["colorSpaceConversion"] = colorSpaceConversion
    o["resizeWidth"] = resizeWidth
    o["resizeHeight"] = resizeHeight
    o["resizeQuality"] = resizeQuality
    return o
}

/**
 * Exposes the JavaScript
[EventSource](https://developer.mozilla.org/en/docs/Web/API/EventSource) to Kotlin
 */

open class EventSource(url: String, eventSourceInitDict: EventSourceInit = definedExternally) : EventTarget {
    open val url: String
    open val withCredentials: Boolean
    open val readyState: Short
    var onopen: ((Event) -> dynamic)?
    var onmessage: ((MessageEvent) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    fun close()
}

interface EventSourceInit {
    var withCredentials: Boolean? /* = false */
}

@kotlin.internal.InlineOnly
internal inline fun EventSourceInit(withCredentials: Boolean? = false): EventSourceInit { return this }

val url = ""
EventSourceInit(withCredentials: Boolean? = false): EventSourceInit {
    val o = js("({})")
    o["withCredentials"] = withCredentials
    return o
}

EventSourceInit(withCredentials: Boolean? = false): EventSourceInit {
    val o = js("({})")
    o["withCredentials"] = withCredentials
    return o
}

WebSocket(url: String, protocols: dynamic = definedExternally): WebSocket {
    this.url = url
    this.protocols = protocols
    return this
}

/**
 * Exposes the JavaScript [WebSocket](https://developer.mozilla.org/en/docs/Web/API/WebSocket) to Kotlin
 */
public external open class WebSocket(url: String, protocols: dynamic = definedExternally) : EventTarget {
    open val url: String
    open val readyState: Short
    open val bufferedAmount: Number
    var onopen: ((Event) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    var onclose: ((Event) -> dynamic)?
    open val extensions: String
    open val protocol: String
    var onmessage: ((MessageEvent) -> dynamic)?
    var binaryType: BinaryType
    fun close(code: Short = definedExternally, reason: String = definedExternally)
    fun send(data: String)
    fun send(data: Blob)
    fun send(data: ArrayBuffer)
    fun send(data: ArrayBufferView)

    companion object {
        val CONNECTING: Short
        val OPEN: Short
        val CLOSING: Short
        val CLOSED: Short
    }
}

CloseEvent(type: String, eventInitDict: CloseEventInit = definedExternally) : CloseEvent {
    this.type = type
    this.eventInitDict = eventInitDict
    return this
}

CloseEvent(type: String, eventInitDict: CloseEventInit = definedExternally) : CloseEvent {
    this.type = type
    this.eventInitDict = eventInitDict
    return this
}

MessageChannel {
    this.port1 = port1
    this.port2 = port2
}

MessagePort {
    this.onmessage = onmessage
    this.postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
    this.start()
    this.close()
}

BroadcastChannel(name: String) : BroadcastChannel {
    this.name = name
    this.onmessage = onmessage
    return this
}

WorkerGlobalScope {
    this.name = name
    this.applicationCache = applicationCache
    this.onconnect = onconnect
    this.close()
}

DedicatedWorkerGlobalScope {
    this.onmessage = onmessage
    this.postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
    this.close()
}

SharedWorkerGlobalScope {
    this.name = name
    this.open val applicationCache: ApplicationCache
    this.open val onconnect: ((Event) -> dynamic)?
    this.open val close() = definedExternally
}

Open Source Used In DNAC Platform Antelope 9613
[AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to Kotlin

```kotlin
public external interface AbstractWorker {
    var onerror: ((Event) -> dynamic)?
        get() = definedExternally
        set(value)
    var onmessage: ((MessageEvent) -> dynamic)?
    fun terminate()
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}
```

```kotlin
/**
 * Exposes the JavaScript [Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin
 */
public external open class Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {
    var onmessage: ((MessageEvent) -> dynamic)?
    override var onerror: ((Event) -> dynamic)?
    fun terminate()
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}
```

```kotlin
public external interface WorkerOptions {
    var type: WorkerType? /* = WorkerType.CLASSIC */
        get() = definedExternally
        set(value) = definedExternally
    var credentials: RequestCredentials? /* = RequestCredentials.OMIT */
        get() = definedExternally
        set(value) = definedExternally
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun WorkerOptions(type: WorkerType? = WorkerType.CLASSIC, credentials: RequestCredentials? = RequestCredentials.OMIT): WorkerOptions {
    val o = js("({})")
    o["type"] = type
    o["credentials"] = credentials
    return o
}
```

```kotlin
/**
 * Exposes the JavaScript [SharedWorker](https://developer.mozilla.org/en/docs/Web/API/SharedWorker) to Kotlin
 */
public external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {
    open val port: MessagePort
    override var onerror: ((Event) -> dynamic)?
}
```

```kotlin
/**
 * Exposes the JavaScript [NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to Kotlin
 */
public external interface NavigatorConcurrentHardware {
    val hardwareConcurrency: Number
}
```

```kotlin
/**
 * Exposes the JavaScript [WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin
 */
public external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorConcurrentHardware {
    open val serviceWorker: ServiceWorkerContainer
}
```

```kotlin
/**
 * Exposes the JavaScript [WorkerLocation](https://developer.mozilla.org/en/docs/Web/API/WorkerLocation) to Kotlin
 */
public external abstract class WorkerLocation {
    open val href: String
    open val origin: String
    open val protocol: String
    open val host: String
    open val hostname: String
    open val port: String
    open val pathname: String
    open val search: String
    open val hash: String
}
```

```kotlin
/**
 * Exposes the JavaScript [Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to Kotlin
 */
public external abstract class Storage {
    open val length: Int
    fun key(index: Int): String?
    fun removeItem(key: String)
    fun clear()
    fun getItem(key: String): String?
    fun setItem(key: String, value: String)
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline operator fun Storage.get(key: String): String? = asDynamic()[key]
@kotlin.internal.InlineOnly
public inline operator fun Storage.set(key: String, value: String) { asDynamic()[key] = value }
```

```kotlin
/**
 * Exposes the JavaScript [WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin
 */
public external interface WindowSessionStorage {
    val sessionStorage: Storage
}
```

```kotlin
/**
 * Exposes the JavaScript [WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage) to Kotlin
 */
public external interface WindowLocalStorage {
    val localStorage: Storage
}
```

```kotlin
/**
 * Exposes the JavaScript [StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin
 */
public external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) : Event {
    open val key: String?
    open val oldValue: String?
    open val newValue: String?
    open val url: String
    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}
```

```kotlin
public external interface StorageEventInit : EventInit {
    var key: String? /* = null */
        get() = definedExternally
        set(value)
    var oldValue: String?
        get() = definedExternally
        set(value)
    var newValue: String?
        get() = definedExternally
        set(value)
    var url: String?
        get() = definedExternally
        set(value)
    var storageArea: Storage?
        get() = definedExternally
        set(value)
}
```
    val o = js("({})")
    o["key"] = key
    o["oldValue"] = oldValue
    o["newValue"] = newValue
    o["url"] = url
    o["storageArea"] = storageArea
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

public external abstract class HTMLAppletElement : HTMLElement {
    open var align: String
    open var alt: String
    open var archive: String
    open var code: String
    open var codeBase: String
    open var height: String
    open var hspace: Int
    open var name: String
    open var object: String
    open var vspace: Int
    open var width: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }

    /**
     * Exposes the JavaScript [HTMLMarqueeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMarqueeElement) to Kotlin
     */
    public external abstract class HTMLMarqueeElement : HTMLElement {
        open var behavior: String
        open var bgColor: String
        open var direction: String
        open var height: String
        open var hspace: Int
        open var loop: Int
        open var scrollAmount: Int
        open var scrollDelay: Int
        open var trueSpeed: Boolean
        open var vspace: Int
        open var width: String
        open var onbounce: ((Event) -> dynamic)?
        open var onfinish: ((Event) -> dynamic)?
        open var onstart: ((Event) -> dynamic)?
        fun start()
        fun stop()

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val NOTATION_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
        }
    }

    /**
     * Exposes the JavaScript [HTMLFrameSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFrameSetElement) to Kotlin
     */
    public external abstract class HTMLFrameSetElement : HTMLElement, WindowEventHandlers {
        open var cols: String
        open var rows: String

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val NOTATION_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
        }
    }

    public external abstract class HTMLFrameElement : HTMLElement {
        open var name: String
        open var scrolling: String
        open var src: String
        open var frameborder: String
        open var longDesc: String
        open var noResize: Boolean
        open var contentDocument: Document?
        open var contentWindow: Window?
        open var marginHeight: String
        open var marginWidth: String

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val NOTATION_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
        }
    }
}
Boolean? /* = false */

get() = definedExternally

set(value) = definedExternally

var parenthesis = "(\{\})")

o["capture"] = capture

return o


public external interface AddEventListenerOptions : EventListenerOptions {

    var passive: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally

    var once: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally

}

@kotlin.internal.InlineOnly
public inline fun EventListenerOptions(capture: Boolean? = false): EventListenerOptions {

    val o = js("({})")

    o["capture"] = capture

    return o

}

public external interface AddEventListenerOptions : EventListenerOptions {

    var passive: Boolean? = false, once: Boolean? = false, capture: Boolean? = false:

    AddEventListenerOptions {

        val o = js("({})")

        o["passive"] = passive

        o["once"] = once

        return o

    }

}

@kotlin.internal.InlineOnly
public inline func


definedExternally
set(value) = definedExternally\n

set(o = jsObject({}));
setter(\n
childList: childList
attributes: attributes
characterData: characterData
attributeOldValue: attributeOldValue
characterDataOldValue: characterDataOldValue
attributeFilter: attributeFilter
characterDataFilter: characterDataFilter
return o\n
**Exposes the JavaScript [MutationRecord](https://developer.mozilla.org/en/docs/Web/API/MutationRecord) to Kotlin\n**

public external abstract class MutationRecord {
    open val type: String
    open val target: Node
    open val addedNodes: NodeList
    open val removedNodes: NodeList
    open val previousSibling: Node?
    open val nextSibling: Node?
    open val attributeName: String?
    open val attributeNamespace: String?
    open val oldValue: String?
}

/**
 * Exposes the JavaScript [Node](https://developer.mozilla.org/en/docs/Web/API/Node) to Kotlin\n */
public external abstract class Node : EventTarget {
    open val nodeType: Short
    open val nodeName: String
    open val baseURI: String
    open val isConnected: Boolean
    open val ownerDocument: Document?
    open val parentNode: Node?
    open val parentElement: Element?
    open val childNodes: NodeList
    open val firstChild: Node?
    open val lastChild: Node?
    open val previousSibling: Node?
    open val nextSibling: Node?
    open var nodeValue: String?
    open var textContent: String?
    fun getRootNode(options: GetRootNodeOptions = definedExternally): Node
    fun hasChildNodes(): Boolean
    fun normalize()
    fun cloneNode(deep: Boolean = definedExternally): Node
    fun isEqualNode(otherNode: Node?): Boolean
    fun isSameNode(otherNode: Node?): Boolean
    fun compareDocumentPosition(other: Node): Short
    fun contains(other: Node?): Boolean
    fun lookupPrefix(namespace: String?): String?
    fun lookupNamespaceURI(prefix: String?): String?
    fun isDefaultNamespace(namespace: String?): Boolean
    fun insertBefore(node: Node, child: Node?): Node
    fun appendChild(node: Node): Node
    fun replaceChild(node: Node, child: Node): Node
    fun removeChild(child: Node): Node

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

public external interface GetRootNodeOptions {
    var composed: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun GetRootNodeOptions(composed: Boolean? = false): GetRootNodeOptions {
    val o = jsObject({});
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [Document](https://developer.mozilla.org/en/docs/Web/API/Document) to Kotlin\n */
    open val implementation: DOMImplementation
    open val URL: String
    open val documentURI: String
    open val origin: String
    open val compatMode: String
    open val characterSet: String
    open val inputEncoding: String
    open val contentType: String
    open val doctype: DocumentType?
    open val documentElement: Element?
    open val location: Location?
    var domain: String
    open val referrer: String
    var cookie: String
    open val lastModified: String
    open val readyState: DocumentReadyState
    var title: String
    var dir: String
    var body: HTMLElement?
    open val head: HTMLHeadElement?
    open val images: HTMLCollection
    open val embeds: HTMLCollection
    open val plugins: HTMLCollection
    open val links: HTMLCollection
    open val forms: HTMLCollection
    open val scripts: HTMLCollection
    open val documentElement: Element?
    open val location: Location?
    var domain: String
    open val referrer: String
    var cookie: String
    open val lastModified: String
    open val readyState: DocumentReadyState
    var title: String
    var dir: String
    var body: HTMLElement?
    open val head: HTMLHeadElement?
    open val images: HTMLCollection
    open val embeds: HTMLCollection
    open val plugins: HTMLCollection
    open val links: HTMLCollection
    open val forms: HTMLCollection
    open val scripts: HTMLCollection
    open val documentElement: Element?
    open val location: Location?
    var domain: String
    open val referrer: String
    var cookie: String
    open val lastModified: String
    open val readyState: DocumentReadyState
    var title: String
    var dir: String
    var body: HTMLElement?
    open val head: HTMLHeadElement?
    open val images: HTMLCollection
    open val embeds: HTMLCollection
    open val plugins: HTMLCollection
    open val links: HTMLCollection
    open val forms: HTMLCollection
    open val scripts: HTMLCollection
    open val documentElement: Element?
    open val location: Location?
    var domain: String
    open val referrer: String
    var cookie: String
    open val lastModified: String
    open val readyState: DocumentReadyState
    var title: String
    var dir: String
    var body: HTMLElement?
    open val head: HTMLHeadElement?
    open val images: HTMLCollection
    open val embeds: HTMLCollection
    open val plugins: HTMLCollection
    open val links: HTMLCollection
    open val forms: HTMLCollection
    open val scripts: HTMLCollection
    open val documentElement: Element?
    open val location: Location?
    var domain: String
    open val referrer: String
    var cookie: String
    open val
var designMode: String
var onreadystatechange: ((Event) -> dynamic)
override var onabort: ((Event) -> dynamic)
override var onblur: ((FocusEvent) -> dynamic)
override var oncancel: ((Event) -> dynamic)
override var oncanplay: ((Event) -> dynamic)
override var oncanplaythrough: ((Event) -> dynamic)
override var onchangeposition: ((Event) -> dynamic)
override var oncontextmenu: ((MouseEvent) -> dynamic)
override var oncuechange: ((Event) -> dynamic)
override var ondrag: ((DragEvent) -> dynamic)
override var ondragend: ((DragEvent) -> dynamic)
override var ondragenter: ((DragEvent) -> dynamic)
override var ondragleave: ((DragEvent) -> dynamic)
override var ondragover: ((DragEvent) -> dynamic)
override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)
override var onfocus: ((FocusEvent) -> dynamic)
override var onkeydown: ((KeyboardEvent) -> dynamic)
override var onkeypress: ((KeyboardEvent) -> dynamic)
override var onkeyup: ((KeyboardEvent) -> dynamic)
override var onload: ((Event) -> dynamic)
override var onloadend: ((ProgressEvent) -> dynamic)
override var onmessage: ((MessageEvent) -> dynamic)
override var onmousemove: ((MouseEvent) -> dynamic)
override var onmouseout: ((MouseEvent) -> dynamic)
override var onmouseover: ((MouseEvent) -> dynamic)
override var onmouseup: ((MouseEvent) -> dynamic)
override var onwheel: ((WheelEvent) -> dynamic)
override var onpause: ((Event) -> dynamic)
override var onplay: ((Event) -> dynamic)
override var onplaying: ((Event) -> dynamic)
override var onratechange: ((Event) -> dynamic)
override var onreset: ((Event) -> dynamic)
override var onscroll: ((Event) -> dynamic)
override var onseeking: ((Event) -> dynamic)
override var onselect: ((Event) -> dynamic)
override var onsubmit: ((Event) -> dynamic)
override var onstop: ((Event) -> dynamic)
override var onstorage: ((StorageEvent) -> dynamic)
override var onstoragechange: ((StorageEvent) -> dynamic)
override var onstorageend: ((StorageEvent) -> dynamic)
override var onstorageerror: ((StorageEvent) -> dynamic)
override var onstorageupdate: ((StorageEvent) -> dynamic)
override var onstorageveto: ((StorageEvent) -> dynamic)
override var onstoragevetochange: ((StorageEvent) -> dynamic)
override var onstoragevetoend: ((StorageEvent) -> dynamic)
override var onstoragevetoerror: ((StorageEvent) -> dynamic)
override var onstoragevetoupdate: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdate: ((StorageEvent) -> dynamic)
override var onstoragevetoenderror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdateerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerrorerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerrorerrorerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerrorerrorerrorerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerrorerrorerrorerrorerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerrorerrorerrorerrorerrorerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerrorerrorerrorerrorerrorerrorerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerrorerrorerrorerrorerrorerrorerrorerror: ((StorageEvent) -> dynamic)
override var onstoragevetoendupdaterrorerrorerrorerrorerrorerrorerrorerrorerrorerror: (}}

Open Source Used In DNAC Platform Antelope 9619
options: ElementCreationOptions = definedExternally): Element
    fun createElementNS(namespace: String?,
        qualifiedName: String, options: ElementCreationOptions = definedExternally): Element
    fun createDocumentFragment(): DocumentFragment
    fun createTextNode(data: String): Text
    fun createCDATASection(data: String): CDATASection
    fun createComment(data: String): Comment
    fun createProcessingInstruction(target: String, data: String): ProcessingInstruction
    fun importNode(node: Node, deep: Boolean = definedExternally): Node
    fun adoptNode(node: Node): Node
    fun createAttribute(localName: String): Attr
    fun createAttributeNS(namespace: String?,
        qualifiedName: String): Attr
    fun createEvent("interface": String): Event
    fun createRange(): Range
    fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally): TreeWalker
    fun getElementsByTagName(elementName: String): NodeList
    fun open(type: String = definedExternally, replace: String = definedExternally): Document
    fun close()
    fun write(vararg text: String)
    fun writeln(vararg text: String)
    fun hasFocus(): Boolean
    fun execCommand(commandId: String, showUI: Boolean = definedExternally, value: String = definedExternally): Boolean
    fun queryCommandEnabled(commandId: String): Boolean
    fun queryCommandIndeterm(commandId: String): Boolean
    fun queryCommandState(commandId: String): Boolean
    fun queryCommandSupported(commandId: String): Boolean
    fun queryCommandValue(commandId: String): String
    fun clear()
    fun captureEvents()
    fun releaseEvents()

    fun elementFromPoint(x: Double, y: Double): Element?
    fun elementsFromPoint(x: Double, y: Double): Array<Element>
    fun caretPositionFromPoint(x: Double, y: Double): CaretPosition?
    fun createTouchList(vararg touches: Touch): TouchList
    fun exitFullscreen(): Promise<Unit>
    override fun "getter": Element?
    override fun "getter": NodeList
    override fun querySelector(selectors: String): Element?
    override fun querySelectorAll(selectors: String): NodeList
    override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>
    override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
    override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
    override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }

    @kotlin.internal.InlineOnly
    public inline operator fun Document.get(name: String): dynamic =
        asDynamic("getter")[name]/* * Exposes the JavaScript
[XMLDocument] to Kotlin */

Open Source Used In DNAC Platform Antelope 9620
Exposes the JavaScript [NamedNodeMap](https://developer.mozilla.org/en/docs/Web/API/NamedNodeMap) to Kotlin via an external abstract class [NamedNodeMap](https://developer.mozilla.org/en/docs/Web/API/NamedNodeMap) to Kotlin:

```kotlin
*\npublic external abstract class NamedNodeMap : ItemArrayLike<Attr> {
    \n    fun getNamedItemNS(namespace: String?, localName: String): Attr?
    fun setNamedItem(attr: Attr): Attr?
    fun removeNamedItemNS(qualifiedName: String): Attr?
    fun removeNamedItemNS(namespace: String?, localName: String): Attr?
    override fun item(index: Int): Attr?
    fun getNamedItem(qualifiedName: String): Attr?
}

@kotlin.internal.InlineOnly
public inline operator fun NamedNodeMap.get(index: Int): Attr? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun NamedNodeMap.get(qualifiedName: String): Attr? = asDynamic()[qualifiedName]

/**
 * Exposes the JavaScript [Attr](https://developer.mozilla.org/en/docs/Web/API/Attr) to Kotlin
 */
public external abstract class Attr : Node {
    open val namespaceURI: String?
    open val prefix: String?
    open val localName: String
    open val name: String
    open var value: String
    open val ownerElement: Element?
    open var specified: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript [CharacterData](https://developer.mozilla.org/en/docs/Web/API/CharacterData) to Kotlin
 */
public external abstract class CharacterData : Node, NonDocumentTypeChildNode, ChildNode {
    open var data: String
    open val length: Int
    fun substringData(offset: Int, count: Int): String
    fun appendData(data: String)
    fun insertData(offset: Int, data: String)
    fun deleteData(offset: Int, count: Int)
    fun replaceData(offset: Int, count: Int, data: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript [Text](https://developer.mozilla.org/en/docs/Web/API/Text) to Kotlin
 */
public external open class Text(data: String = definedExternally) : CharacterData, Slotable, GeometryUtils {
    open val wholeText: String
    override val assignedSlot: HTMLSlotElement?
    override val previousElementSibling: Element?
    override val nextElementSibling: Element?
    fun splitText(offset: Int): Text
    override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): DOMQuad
    override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
    override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
    override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint
    override fun before(vararg nodes: dynamic)
    override fun after(vararg nodes: dynamic)
    override fun replaceWith(vararg nodes: dynamic)
    override fun remove()

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
```
DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val
DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val
DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val
DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val
DOCUMENT_POSITION_CONTAINED_BY: Short
val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

// Exposes the JavaScript [CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin

```kotlin
public external open class CDATASection : Text {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

```kotlin
/**
 * Exposes the JavaScript [ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin
 */
public external abstract class ProcessingInstruction : CharacterData, LinkStyle,
UnionElementOrProcessingInstruction {
    open val target: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

```kotlin
/**
 * Exposes the JavaScript [Comment](https://developer.mozilla.org/en/docs/Web/API/Comment) to Kotlin
 */
public external open class Comment(data: String = definedExternally) : CharacterData {
    override val previousElementSibling: Element?
    override val nextElementSibling: Element?
    override fun before(vararg nodes: dynamic)
    override fun after(vararg nodes: dynamic)
    override fun replaceWith(vararg nodes: dynamic)
    override fun remove()

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

```kotlin
/**
 * Exposes the JavaScript [Range](https://developer.mozilla.org/en/docs/Web/API/Range) to Kotlin
 */
public external open class Range {
    open val startContainer: Node
    open val startOffset: Int
    open val endContainer: Node
    open val endOffset: Int
    open val collapsed: Boolean
    open val commonAncestorContainer: Node
    fun setStart(node: Node, offset: Int)
    fun setEnd(node: Node, offset: Int)
    fun setStartBefore(node: Node)
    fun setStartAfter(node: Node)
    fun setEndBefore(node: Node)
    fun setEndAfter(node: Node)
    fun collapse(toStart: Boolean = definedExternally)
    fun selectNode(node: Node)
    fun selectNodeContents(node: Node)
}
fun compareBoundaryPoints(how: Short, sourceRange: Range): Short
fun deleteContents(): DocumentFragment
fun cloneContents(): DocumentFragment
fun insertNode(node: Node): Boolean
fun getClientRects(): Array<DOMRect>
fun getBoundingClientRect(): DOMRect
fun createContextualFragment(fragment: String): DocumentFragment

companion object {
    val START_TO_START: Short
    val START_TO_END: Short
    val END_TO_END: Short
    val END_TO_START: Short
}

/**
 * Exposes the JavaScript [NodeIterator](https://developer.mozilla.org/en/docs/Web/API/NodeIterator) to Kotlin
 */
public external abstract class NodeIterator {
    open val root: Node
    open val referenceNode: Node
    open val pointerBeforeReferenceNode: Boolean
    open val whatToShow: Int
    open val filter: NodeFilter?
    fun nextNode(): Node?
    fun previousNode(): Node?
    fun detach()
}

/**
 * Exposes the JavaScript [TreeWalker](https://developer.mozilla.org/en/docs/Web/API/TreeWalker) to Kotlin
 */
public external abstract class TreeWalker {
    open val root: Node
    open val whatToShow: Int
    open val filter: NodeFilter?
    open var currentNode: Node
    fun parentNode(): Node?
    fun firstChild(): Node?
    fun lastChild(): Node?
    fun previousSibling(): Node?
    fun nextSibling(): Node?
    fun previousNode(): Node?
    fun nextNode(): Node?
}

/**
 * Exposes the JavaScript [NodeFilter](https://developer.mozilla.org/en/docs/Web/API/NodeFilter) to Kotlin
 */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NodeFilter {
    fun acceptNode(node: Node): Short

    companion object {
        val FILTER_ACCEPT: Short
        val FILTER_REJECT: Short
        val FILTER_SKIP: Short
        val SHOW_ALL: Int
        val SHOW_ELEMENT: Int
        val SHOW_ATTRIBUTE: Int
        val SHOW_TEXT: Int
        val SHOW_CDATA_SECTION: Int
        val SHOW_ENTITY_REFERENCE: Int
        val SHOW_ENTITY: Int
        val SHOW_PROCESSING_INSTRUCTION: Int
        val SHOW_COMMENT: Int
        val SHOW_DOCUMENT: Int
        val SHOW_DOCUMENT_TYPE: Int
        val SHOW_DOCUMENT_FRAGMENT: Int
        val SHOW_NOTATION: Int
    }
}

/**
 * Exposes the JavaScript [DOMTokenList](https://developer.mozilla.org/en/docs/Web/API/DOMTokenList) to Kotlin
 */
public external abstract class DOMTokenList : ItemArrayLike<String> {
    open var value: String
    fun contains(token: String): Boolean
    fun add(vararg tokens: String)
    fun remove(vararg tokens: String)
    fun toggle(token: String, force: Boolean = definedExternally): Boolean
    fun replace(token: String, newToken: String)
    fun supports(token: String): Boolean
    override fun item(index: Int): String?
}

@kotlin.internal.InlineOnly
public inline operator fun DOMTokenList.get(index: Int): String? = asDynamic()[index]

/**
 * Exposes the JavaScript [DOMPointReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMPointReadOnly) to Kotlin
 */
public external open class DOMPointReadOnly(x: Double, y: Double, z: Double, w: Double) {
    open val x: Double
    open val y: Double
    open val z: Double
    open val w: Double
    fun matrixTransform(matrix: DOMMatrixReadOnly): DOMPoint
}

/**
 * Exposes the JavaScript [DOMPoint](https://developer.mozilla.org/en/docs/Web/API/DOMPoint) to Kotlin
 */
public external open class DOMPoint : DOMPointReadOnly {
    constructor(point: DOMPointInit)
    constructor(x: Double = definedExternally, y: Double = definedExternally, z: Double = definedExternally, w: Double = definedExternally)
    override var x: Double
    override var y: Double
    override var z: Double
    override var w: Double
}

/**
 * Exposes the JavaScript [DOMPointInit](https://developer.mozilla.org/en/docs/Web/API/DOMPointInit) to Kotlin
 */
public external interface DOMPointInit {
    var x: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var y: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var z: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var w: Double? /* = 1.0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun DOMPointInit(x: Double? = 0.0, y: Double? = 0.0, z: Double? = 0.0, w: Double? = 0.0): DOMPoint = definedExternally
= 0.0, z: Double? = 0.0, w: Double? = 1.0): DOMPointInit {
    \n    val o = js('({})').o
    o['x'] = x
    o['y'] = y
    o['z'] = z
    o['w'] = w
    return o
}

/**
 * Exposes the JavaScript
 * [DOMRect](https://developer.mozilla.org/en/docs/Web/API/DOMRect) to Kotlin
 */
public external open class DOMRect(x: Double = definedExternally, y: Double = definedExternally, width: Double = definedExternally, height: Double = definedExternally) : DOMRectReadOnly {
    override var x: Double
    override var y: Double
    override var width: Double
    override var height: Double
}

/**
 * Exposes the JavaScript
 * [DOMRectReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMRectReadOnly) to Kotlin
 */
public external open class DOMRectReadOnly(x: Double, y: Double, width: Double, height: Double) {
    open val x: Double
    open val y: Double
    open val width: Double
    open val height: Double
}

public external interface DOMRectInit {
    var x: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var y: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var width: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var height: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun DOMRectInit(x: Double? = 0.0, y: Double? = 0.0, width: Double? = 0.0, height: Double? = 0.0): DOMRectInit {
    val o = js('({})').o
    o['x'] = x
    o['y'] = y
    o['width'] = width
    o['height'] = height
    return o
}

public external interface DOMRectList : ItemArrayLike<DOMRect> {
    override fun item(index: Int): DOMRect?
}

@kotlin.internal.InlineOnly
public inline operator fun DOMRectList.get(index: Int): DOMRect? = asDynamic()[index]

/**
 * Exposes the JavaScript
 * [DOMQuad](https://developer.mozilla.org/en/docs/Web/API/DOMQuad) to Kotlin
 */
public external open class DOMQuad {
    constructor(p1: DOMPointInit = definedExternally, p2: DOMPointInit = definedExternally, p3: DOMPointInit = definedExternally, p4: DOMPointInit = definedExternally)
    constructor(rect: DOMRectInit)
    open val p1: DOMPoint
    open val p2: DOMPoint
    open val p3: DOMPoint
    open val p4: DOMPoint
    open val bounds: DOMRectReadOnly
}

/**
 * Exposes the JavaScript
 * [DOMMatrixReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMMatrixReadOnly) to Kotlin
 */
public external open class DOMMatrixReadOnly(numberSequence: Array<Double>) {
    open val a: Double
    open val b: Double
    open val c: Double
    open val d: Double
    open val e: Double
    open val f: Double
    open val m11: Double
    open val m12: Double
    open val m13: Double
    open val m14: Double
    open val m21: Double
    open val m22: Double
    open val m23: Double
    open val m24: Double
    open val m31: Double
    open val m32: Double
    open val m33: Double
    open val m34: Double
    open val m41: Double
    open val m42: Double
    open val m43: Double
    open val m44: Double
    open val is2D: Boolean
    open val isIdentity: Boolean
    fun translate(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix
    fun scale(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
    fun scale3d(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix
    fun scaleNonUniform(scaleX: Double, scaleY: Double, scaleZ: Double): DOMMatrix
    fun rotate(angle: Double, originX: Double, originY: Double): DOMMatrix
    fun rotateFromVector(x: Double, y: Double): DOMMatrix
    fun rotateAxisAngle(x: Double, y: Double, z: Double, angle: Double): DOMMatrix
    fun skewX(sx: Double): DOMMatrix
    fun skewY(sy: Double): DOMMatrix
    fun multiply(other: DOMMatrix): DOMMatrix
    fun flipX(): DOMMatrix
    fun flipY(): DOMMatrix
    fun inverse(): DOMMatrix
    fun transformPoint(point: DOMPointInit = definedExternally): DOMPoint
    fun toFloat32Array(): Float32Array
    fun toFloat64Array(): Float64Array
}

/**
 * Exposes the JavaScript
 * [DOMMatrix](https://developer.mozilla.org/en/docs/Web/API/DOMMatrix) to Kotlin
 */
public external open class DOMMatrix() : DOMMatrixReadOnly {
    constructor(transformList: String)
    constructor(other: DOMMatrixReadOnly)
    constructor(array32: Float32Array)
    constructor(array64: Float64Array)
    override var a: Double
    override var b: Double
    override
null
Any?) -> dynamic)?
override var onfocus: ((FocusEvent) -> dynamic)?
override var oninput: ((InputEvent) -> dynamic)?
override var oninvalid: ((Event) -> dynamic)?
override var onkeydown: ((KeyboardEvent) -> dynamic)?
override var onkeypress: ((KeyboardEvent) -> dynamic)?
override var onkeyup: ((KeyboardEvent) -> dynamic)?
override var oncontextmenu: ((MouseEvent) -> dynamic)?
override var oncontextmenu: ((MouseEvent) -> dynamic)?
override var oncontextmenu: ((MouseEvent) -> dynamic)?
override var onmousedown: ((MouseEvent) -> dynamic)?
override var onmouseenter: ((MouseEvent) -> dynamic)?
override var onmouseleave: ((MouseEvent) -> dynamic)?
override var onmousemove: ((MouseEvent) -> dynamic)?
override var onmouseout: ((MouseEvent) -> dynamic)?
override var onmouseover: ((MouseEvent) -> dynamic)?
override var onmouseup: ((MouseEvent) -> dynamic)?
override var onwheel: ((WheelEvent) -> dynamic)?
override var onpause: ((Event) -> dynamic)?
override var onplay: ((Event) -> dynamic)?
override var onplaying: ((Event) -> dynamic)?
override var onratechange: ((Event) -> dynamic)?
override var onreadystatechange: ((Event) -> dynamic)?
override var onresize: ((Event) -> dynamic)?
override var onscroll: ((Event) -> dynamic)?
override var onseeked: ((Event) -> dynamic)?
override var onseeking: ((Event) -> dynamic)?
override var onselect: ((Event) -> dynamic)?
override var onshow: ((Event) -> dynamic)?
override var onstalled: ((Event) -> dynamic)?
override var onsubmit: ((Event) -> dynamic)?
override var onsuspend: ((Event) -> dynamic)?
override var ontimeupdate: ((Event) -> dynamic)?
override var ontoggle: ((Event) -> dynamic)?
override var onvolumechange: ((Event) -> dynamic)?
override var onwaiting: ((Event) -> dynamic)?
override var ongотpointercapture: ((PointerEvent) -> dynamic)?
override var onpointercancel: ((PointerEvent) -> dynamic)?
override var onpointerdown: ((PointerEvent) -> dynamic)?
override var onpointercancel: ((PointerEvent) -> dynamic)?
override var onpointermove: ((PointerEvent) -> dynamic)?
override var onpointerleave: ((PointerEvent) -> dynamic)?
override var oncopy: ((ClipboardEvent) -> dynamic)?
override var oncut: ((ClipboardEvent) -> dynamic)?
override var onpaste: ((ClipboardEvent) -> dynamic)?

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
}
Audio(src: String = definedExternally) : HTMLAudioElement

    override var onabort: ((Event) -> dynamic)?
    override var oncanplay: ((Event) -> dynamic)?
    override var oncanplaythrough: ((Event) -> dynamic)?
    override var onchange: ((Event) -> dynamic)?
    override var oncontextmenu: ((MouseEventArgs) -> dynamic)?
    override var oncercuechange: ((Event) -> dynamic)?
    override var ondurationchange: ((Event) -> dynamic)?
    override var onemptied: ((Event) -> dynamic)?
    override var onended: ((Event) -> dynamic)?
    override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?
    override var onfocus: ((FocusEvent) -> dynamic)?
    override var oninput: ((InputEvent) -> dynamic)?
    override var oninvalid: ((Event) -> dynamic)?
    override var onkeydown: ((KeyboardEvent) -> dynamic)?
    override var onkeypress: ((KeyboardEvent) -> dynamic)?
    override var onkeyup: ((KeyboardEvent) -> dynamic)?
    override var onload: ((Event) -> dynamic)?
    override var onloadeddata: ((Event) -> dynamic)?
    override var onloadedmetadata: ((Event) -> dynamic)?
    override var onloadend: ((Event) -> dynamic)?
    override var onloadstart: ((ProgressEvent) -> dynamic)?
    override var onpause: ((Event) -> dynamic)?
    override var onplay: ((Event) -> dynamic)?
    override var onplaying: ((Event) -> dynamic)?
    override var onprogress: ((ProgressEvent) -> dynamic)?
    override var onratechange: ((Event) -> dynamic)?
    override var onreset: ((Event) -> dynamic)?
    override var onresize: ((Event) -> dynamic)?
    override var onscroll: ((Event) -> dynamic)?
    override var onseeked: ((Event) -> dynamic)?
    override var onseeking: ((Event) -> dynamic)?
    override var onselect: ((Event) -> dynamic)?
    override var onshow: ((Event) -> dynamic)?
    override var onstalled: ((Event) -> dynamic)?
    override var onsubmit: ((Event) -> dynamic)?
    override var onsuspend: ((Event) -> dynamic)?
    override var ontimeupdate: ((Event) -> dynamic)?
    override var ontoggle: ((Event) -> dynamic)?
    override var onvolumechange: ((Event) -> dynamic)?
    override var onwaiting: ((Event) -> dynamic)?
    override var ongotpointercapture: ((EventHandler) -> dynamic)?
    override var onlostpointercapture: ((EventHandler) -> dynamic)?
    override var onpointercancel: ((EventHandler) -> dynamic)?
    override var onpointermove: ((EventHandler) -> dynamic)?
    override var onpointerdown: ((EventHandler) -> dynamic)?
    override var onpointerenter: ((EventHandler) -> dynamic)?
    override var onpointermove: ((EventHandler) -> dynamic)?
    override var onpointerover: ((EventHandler) -> dynamic)?
    override var onpointerout: ((EventHandler) -> dynamic)?
    override var onpointercancel: ((EventHandler) -> dynamic)?
    override var onqueryselector: ((EventHandler) -> dynamic)?
    override var onqueryselectorall: ((EventHandler) -> dynamic)?
    override var oncopy: ((ClipboardEvent) -> dynamic)?
    override var oncut: ((ClipboardEvent) -> dynamic)?
    override var onpaste: ((ClipboardEvent) -> dynamic)?
    override var contenteditable: String
    override val iscontenteditable: Boolean
    override val style: CSSStyleDeclaration
    override val children: HTMLCollection
    override val firstElementChild: Element?
    override val lastElementChild: Element?
    override val childElementCount: Int
    override val previousElementSibling: Element?
    override val nextElementSibling: Element?
    override val assignedSlot: HTMILSlotElement
    override fun prepend(vararg nodes: dynamic): dynamic
    override fun append(vararg nodes: dynamic): dynamic
    override fun querySelector(selectors: String): Element?
    override fun querySelectorAll(selectors: String): NodeList
    override fun before(vararg nodes: dynamic): dynamic
    override fun after(vararg nodes: dynamic): dynamic
    override fun replaceWith(vararg nodes: dynamic): dynamic
    override fun remove(): Unit
    override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>
Open Source Used In DNAC Platform Antelope
onvolumechange: ((Event) -> dynamic)?
override var onwaiting: ((Event) -> dynamic)?
override var ongopointercapture: ((PointerEvent) -> dynamic)?
override var onlostpointercapture: ((PointerEvent) -> dynamic)?
override var onpointerdown: ((PointerEvent) -> dynamic)?
override var onpointerup: ((PointerEvent) -> dynamic)?
override var onpointercancel: ((PointerEvent) -> dynamic)?
override var onpointermove: ((PointerEvent) -> dynamic)?
override var onpointerover: ((PointerEvent) -> dynamic)?
override var onpointerout: ((PointerEvent) -> dynamic)?
override var oncopy: ((ClipboardEvent) -> dynamic)?
override var oncut: ((ClipboardEvent) -> dynamic)?
override var onpaste: ((ClipboardEvent) -> dynamic)?
override var contentEditable: String
override val isContentEditable: Boolean
override var style: CSSStyleDeclaration
override val children: HTMLCollection
override val firstElementChild: Element?
override val lastElementChild: Element?
override val childElementCount: Int
override val previousElementSibling: Element?
override val nextElementSibling: Element?
override val assignedSlot: HTMLSlotElement?
override fun prepend(vararg nodes: dynamic): HTMLCollection
override fun append(vararg nodes: dynamic): HTMLCollection
override fun querySelector(selectors: String): Element?
override fun querySelectorAll(selectors: String): NodeList
override fun before(vararg nodes: dynamic): HTMLCollection
override fun after(vararg nodes: dynamic): HTMLCollection
override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>
override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

public external interface UnionElementOrHTMLCollection

public external interface UnionElementOrRadioNodeList

public external interface UnionHTMLOptGroupElementOrHTMLOptionElement

public external interface MediaProvider

public external interface UnionAudioTrackOrTextTrackOrVideoTrack

public external interface UnionElementOrMouseEvent

public external interface RenderingContext

public external interface HTMLOrSVGImageElement : CanvasImageSource

public external interface CanvasImageSource : ImageBitmapSource

public external interface ImageBitmapSource

public external interface UnionMessagePortOrWindowProxy

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface DocumentReadyState {
companion object
}


/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanPlayTypeResult {
companion object
}

public inline val CanPlayTypeResult.Companion.EMPTY: CanPlayTypeResult get() =
CanPlayTypeResult get() = \"\".asDynamic().unsafeCast<CanPlayTypeResult>().\n\npublic inline val
CanPlayTypeResult.Companion.MAYBE: CanPlayTypeResult get() =
\"maybe\".asDynamic().unsafeCast<CanPlayTypeResult>().\n\npublic inline val
CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() =
\"probably\".asDynamic().unsafeCast<CanPlayTypeResult>().\n/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface TextTrackMode
{
    companion object
}

public inline val TextTrackMode.Companion.DISABLED: TextTrackMode get() =
\"disabled\".asDynamic().unsafeCast<TextTrackMode>().\n\npublic inline val
TextTrackMode.Companion.HIDDEN: TextTrackMode get() =
\"hidden\".asDynamic().unsafeCast<TextTrackMode>().\nTextTrackMode.Companion.SHOWING: TextTrackMode get() =
\"showing\".asDynamic().unsafeCast<TextTrackMode>().\n/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface TextTrackKind
{
    companion object
}

public inline val TextTrackKind.Companion.SUBTITLES: TextTrackKind get() =
\"subtitles\".asDynamic().unsafeCast<TextTrackKind>().\n\npublic inline val
TextTrackKind.Companion.CAPTIONS: TextTrackKind get() =
\"captions\".asDynamic().unsafeCast<TextTrackKind>().\n\npublic inline val
TextTrackKind.Companion.DESCRIPTIONS: TextTrackKind get() =
\"descriptions\".asDynamic().unsafeCast<TextTrackKind>().\n\npublic inline val
TextTrackKind.Companion.CHAPTERS: TextTrackKind get() =
\"chapters\".asDynamic().unsafeCast<TextTrackKind>().\n\npublic inline val
TextTrackKind.Companion.METADATA: TextTrackKind get() =
\"metadata\".asDynamic().unsafeCast<TextTrackKind>().\n/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface SelectionMode
{
    companion object
}

public inline val SelectionMode.Companion.SELECT: SelectionMode get() =
\"select\".asDynamic().unsafeCast<SelectionMode>().\npublic inline val
SelectionMode.Companion.START: SelectionMode get() = \"start\".asDynamic().unsafeCast<SelectionMode>().\npublic inline val
SelectionMode.Companion.END: SelectionMode get() = \"end\".asDynamic().unsafeCast<SelectionMode>().\npublic inline val
SelectionMode.Companion.Preserve: SelectionMode get() = \"preserve\".asDynamic().unsafeCast<SelectionMode>().\n/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasFillRule
{
    companion object
}

public inline val CanvasFillRule.Companion.NONZERO: CanvasFillRule get() = \"nonzero\".asDynamic().unsafeCast<CanvasFillRule>().\npublic inline val
CanvasFillRule.Companion.EVENODD: CanvasFillRule get() = \"evenodd\".asDynamic().unsafeCast<CanvasFillRule>().\n/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ImageSmoothingQuality
{
    companion object
}

ImageSmoothingQuality.Companion.HIGH: ImageSmoothingQuality get() = \"high\".asDynamic().unsafeCast<ImageSmoothingQuality>().\n/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasLineCap
{
    companion object
}

public inline val CanvasLineCap.Companion.BUTT: CanvasLineCap get() = \"butt\".asDynamic().unsafeCast<CanvasLineCap>().\npublic inline val
CanvasLineCap.Companion.BUTT: CanvasLineCap get() = \"butt\".asDynamic().unsafeCast<CanvasLineCap>().\npublic inline val
CanvasLineCap.Companion.ROUNDED: CanvasLineCap get() = \"round\".asDynamic().unsafeCast<CanvasLineCap>().\npublic inline val
CanvasLineCap.Companion.SQUARE: CanvasLineCap get() = \"square\".asDynamic().unsafeCast<CanvasLineCap>().\n/* please, don't implement this interface! */
Open Source Used In DNAC Platform Antelope 9633
PremultiplyAlpha.Companion.DEFAULT: PremultiplyAlpha get() =
"default".asDynamic().unsafeCast<PremultiplyAlpha>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ColorSpaceConversion {
    companion object
}

public inline val ColorSpaceConversion.Companion.DEFAULT:
ColorSpaceConversion get() = "default".asDynamic().unsafeCast<ColorSpaceConversion>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ResizeQuality {
    companion object
}

public inline val ResizeQuality.Companion.LOW:
ResizeQuality get() = "low".asDynamic().unsafeCast<ResizeQuality>()

public inline val ResizeQuality.Companion.MEDIUM:
ResizeQuality get() = "medium".asDynamic().unsafeCast<ResizeQuality>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface BinaryType {
    companion object
}

public inline val BinaryType.Companion.ARRAYBUFFER:
BinaryType get() = "arraybuffer".asDynamic().unsafeCast<BinaryType>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface WorkerType {
    companion object
}

public inline val WorkerType.Companion.MODULE:
WorkerType get() = "module".asDynamic().unsafeCast<WorkerType>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ShadowRootMode {
    companion object
}

public inline val ShadowRootMode.Companion.CLOSED:
ShadowRootMode get() = "closed".asDynamic().unsafeCast<ShadowRootMode>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollBehavior {
    companion object
}

public inline val ScrollBehavior.Companion.AUTO:
ScrollBehavior get() = "auto".asDynamic().unsafeCast<ScrollBehavior>()

public inline val ScrollBehavior.Companion.INSTANT:
ScrollBehavior get() = "instant".asDynamic().unsafeCast<ScrollBehavior>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollLogicalPosition {
    companion object
}

public inline val ScrollLogicalPosition.Companion.END:
ScrollLogicalPosition get() = "end".asDynamic().unsafeCast<ScrollLogicalPosition>()

public inline val ScrollLogicalPosition.Companion.NEAREST:
ScrollLogicalPosition get() = "nearest".asDynamic().unsafeCast<ScrollLogicalPosition>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CSSBoxType {
    companion object
}

public inline val CSSBoxType.Companion.MARGIN:
CSSBoxType get() = "margin".asDynamic().unsafeCast<CSSBoxType>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ColorSpaceConversion {
    companion object
}

public inline val ColorSpaceConversion.Companion.NONE:
ColorSpaceConversion get() = "none".asDynamic().unsafeCast<ColorSpaceConversion>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ResizeQuality {
    companion object
}

public inline val ResizeQuality.Companion.HIGH:
ResizeQuality get() = "high".asDynamic().unsafeCast<ResizeQuality>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface BinaryType {
    companion object
}

public inline val BinaryType.Companion.BLOB:
BinaryType get() = "blob".asDynamic().unsafeCast<BinaryType>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface WorkerType {
    companion object
}

public inline val WorkerType.Companion.CLASSIC:
WorkerType get() = "classic".asDynamic().unsafeCast<WorkerType>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ShadowRootMode {
    companion object
}

public inline val ShadowRootMode.Companion.OPEN:
ShadowRootMode get() = "open".asDynamic().unsafeCast<ShadowRootMode>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollBehavior {
    companion object
}

public inline val ScrollBehavior.Companion.SMOOTH:
ScrollBehavior get() = "smooth".asDynamic().unsafeCast<ScrollBehavior>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollLogicalPosition {
    companion object
}

public inline val ScrollLogicalPosition.Companion.START:
ScrollLogicalPosition get() = "start".asDynamic().unsafeCast<ScrollLogicalPosition>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CSSBoxType {
    companion object
}

public inline val CSSBoxType.Companion.CONTENT:
CSSBoxType get() = \"content\".unsafeCast\t<CSSBoxType>()*/\n/*n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n*/

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED,
// DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n
package org.w3c.fetch

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript

[Headers](https://developer.mozilla.org/en/docs/Web/API/Headers) to Kotlin\n
*\n
public external open class Headers(init: dynamic = definedExternally) \n{\n    fun append(name: String, value: String)\n    fun delete(name: String): Boolean\n    fun get(name: String): String?\n    fun has(name: String): Boolean\n    fun set(name: String, value: String)\n}

/**
 * Exposes the JavaScript

[Body](https://developer.mozilla.org/en/docs/Web/API/Body) to Kotlin\n
*\n
public external interface Body \n{\n    val bodyUsed: Boolean\n    fun arrayBuffer(): Promise<ArrayBuffer>\n    fun blob(): Promise<Blob>\n    fun formData(): Promise<FormData>\n    fun json(): Promise<Any?>\n    fun text(): Promise<String>\n}

/**
 * Exposes the JavaScript

[Request](https://developer.mozilla.org/en/docs/Web/API/Request) to Kotlin\n
*\n
public external open class Request(input: dynamic, init: RequestInit = definedExternally) : Body \n{\n    open val method: String\n    open val url: String\n    open val headers: Headers\n    open val type: RequestType\n    open val destination: RequestDestination?\n    open val referrer: String\n    open val referrerPolicy: dynamic\n    open val mode: RequestMode\n    open val credentials: RequestCredentials\n    open val cache: RequestCache\n    open val redirect: RequestRedirect?\n    open val integrity: String\n    open val keepalive: Boolean\n    override val bodyUsed: Boolean\n    fun clone(): Request\n    override fun arrayBuffer(): Promise<ArrayBuffer>\n    override fun blob(): Promise<Blob>\n    override fun formData(): Promise<FormData>\n    override fun json(): Promise<Any?>\n    override fun text(): Promise<String>\n}

public external interface RequestInit \n{\n    var method: String? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n    var headers: dynamic\n        get() = definedExternally
        set(value) = definedExternally\n    var body: dynamic\n        get() = definedExternally
        set(value) = definedExternally\n    var referrer: String? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n    var referrerPolicy: dynamic\n        get() = definedExternally
        set(value) = definedExternally\n    var mode: RequestMode? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n    var credentials: RequestCredentials? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n    var cache: RequestCache? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n    var redirect: RequestRedirect? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n    var integrity: String? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n    var keepalive: Boolean? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n    var window: Any? = undefined\n        get() = definedExternally
        set(value) = definedExternally\n}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["method"] = method
    o["headers"] = headers
    o["body"] = body
    o["referrer"] = referrer
    o["referrerPolicy"] = referrerPolicy
    o["mode"] = mode
    o["credentials"] = credentials
    o["cache"] = cache
    o["redirect"] = redirect
    o["integrity"] = integrity
    o["keepalive"] = keepalive
    o["window"] = window
    return o
}
Boolean
open val status: Short
open val ok: Boolean
open val statusText: String
open val bodyUsed: Boolean
fun clone(): Response
override fun arrayBuffer(): Promise<ArrayBuffer>
override fun blob(): Promise<Blob>
override fun formData(): Promise<FormData>
override fun json(): Promise<Any?>
override fun text(): Promise<String>

companion object {
    fun error(): Response
    fun redirect(url: String, status: Short = definedExternally): Response
}

public external interface ResponseInit {
    var status: Short? /* = 200 */
        get() = definedExternally
        set(value) = definedExternally
    var statusText: String? /* = "OK" */
        get() = definedExternally
        set(value) = definedExternally
    var headers: dynamic
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ResponseInit(status: Short? = 200,
    statusText: String? = "OK", headers: dynamic = undefined): ResponseInit {
    val o = js("({})")
    o["status"] = status
    o["statusText"] = statusText
    o["headers"] = headers
    return o
}

/* please, don't implement
this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestType {
    companion object
}

public inline val RequestType.Companion.EMPTY: RequestType get() = "".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.AUDIO: RequestType get() = "audio".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.FONT: RequestType get() = "font".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.IMAGE: RequestType get() = "image".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.SCRIPT: RequestType get() = "script".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.STYLE: RequestType get() = "style".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.TRACK: RequestType get() = "track".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.VIDEO: RequestType get() = "video".asDynamic().unsafeCast<RequestType>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestDestination {
    companion object
}

public inline val RequestDestination.Companion.EMPTY: RequestDestination get() = "".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.DOCUMENT: RequestDestination get() = "document".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.EMBED: RequestDestination get() = "embed".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.FONT: RequestDestination get() = "font".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.IMAGE: RequestDestination get() = "image".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.MANIFEST: RequestDestination get() = "manifest".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.MEDIA: RequestDestination get() = "media".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.OBJECT: RequestDestination get() = "object".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.REPORT: RequestDestination get() = "report".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.SCRIPT: RequestDestination get() = "script".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.SERVICEWORKER: RequestDestination get() = "serviceworker".asDynamic().unsafeCast<RequestDestination>()
open source used in DNAC platform Antelope

```
RequestDestination.Companion.SHAREDWORKER: RequestDestination get() = "sharedworker".asDynamic().unsafeCast<RequestDestination>()

RequestDestination.Companion стиль: RequestDestination get() = "style".asDynamic().unsafeCast<RequestDestination>()

RequestDestination.Companion.WORKER: RequestDestination get() = "worker".asDynamic().unsafeCast<RequestDestination>()

RequestDestination.Companion.XSLT: RequestDestination get() = "xslt".asDynamic().unsafeCast<RequestDestination>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestMode {
    companion object
}

RequestMode.Companion.NAVIGATE: RequestMode get() = "navigate".asDynamic().unsafeCast<RequestMode>()

RequestMode.Companion.SAME_ORIGIN: RequestMode get() = "same-origin".asDynamic().unsafeCast<RequestMode>()

RequestMode.Companion.NO_CORS: RequestMode get() = "no-cors".asDynamic().unsafeCast<RequestMode>()

RequestMode.Companion.CORS: RequestMode get() = "cors".asDynamic().unsafeCast<RequestMode>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestCredentials {
    companion object
}

RequestCredentials.Companion.Omit: RequestCredentials get() = "omit".asDynamic().unsafeCast<RequestCredentials>()

RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = "same-origin".asDynamic().unsafeCast<RequestCredentials>()

RequestCredentials.Companion.INCLUDE: RequestCredentials get() = "include".asDynamic().unsafeCast<RequestCredentials>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestCache {
    companion object
}

RequestCache.Companion.DEFAULT: RequestCache get() = "default".asDynamic().unsafeCast<RequestCache>()

RequestCache.Companion.NO_STORE: RequestCache get() = "no-store".asDynamic().unsafeCast<RequestCache>()

RequestCache.Companion.RELOAD: RequestCache get() = "reload".asDynamic().unsafeCast<RequestCache>()

RequestCache.Companion.NO_CACHE: RequestCache get() = "no-cache".asDynamic().unsafeCast<RequestCache>()

RequestCache.Companion.FORCE_CACHE: RequestCache get() = "force-cache".asDynamic().unsafeCast<RequestCache>()

RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = "only-if-cached".asDynamic().unsafeCast<RequestCache>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestRedirect {
    companion object
}

RequestRedirect.Companion.FOLLOW: RequestRedirect get() = "follow".asDynamic().unsafeCast<RequestRedirect>()

RequestRedirect.Companion.ERROR: RequestRedirect get() = "error".asDynamic().unsafeCast<RequestRedirect>()

RequestRedirect.Companion.MANUAL: RequestRedirect get() = "manual".asDynamic().unsafeCast<RequestRedirect>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ResponseType {
    companion object
}

ResponseType.Companion.BASIC: ResponseType get() = "basic".asDynamic().unsafeCast<ResponseType>()

ResponseType.Companion.CORS: ResponseType get() = "cors".asDynamic().unsafeCast<ResponseType>()

ResponseType.Companion.DEFAULT: ResponseType get() = "default".asDynamic().unsafeCast<ResponseType>()

ResponseType.Companion.ERROR: ResponseType get() = "error".asDynamic().unsafeCast<ResponseType>()
```

Open Source Used In DNAC Platform Antelope 9638
"opaque\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val
$$opaqueredirect\".asDynamic().unsafeCast<ResponseType>\n\"opaqueredirect\".asDynamic().unsafeCast<ResponseType>\n
public inline val

/**
 * Exposes the JavaScript
 * [MediaStream](https://developer.mozilla.org/en/docs/Web/API/MediaStream) to Kotlin
 */
public external open class MediaStream() : EventTarget, MediaProvider {
    constructor(stream: MediaStream)
    constructor(tracks: Array<MediaStreamTrack>)
    open val id: String
    open val active: Boolean
    var onaddtrack:
            (MediaStreamTrackEvent) -> dynamic?
    var onremovetrack: (MediaStreamTrackEvent) -> dynamic?
    fun
            getAudioTracks(): Array<MediaStreamTrack>
    fun
            getVideoTracks(): Array<MediaStreamTrack>
    fun
            getTracks(): Array<MediaStreamTrack>
    fun
            getTrackById(trackId: String): MediaStreamTrack?
    fun
            addTrack(track: MediaStreamTrack)
    fun
            removeTrack(track: MediaStreamTrack)
    fun
            clone(): MediaStream
}

/**
 * Exposes the JavaScript
 * [MediaStreamTrack](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrack) to Kotlin
 */
public external abstract class MediaStreamTrack : EventTarget {
    open val kind: String
    open val id: String
    open val label: String
    open var enabled: Boolean
    open var muted: Boolean
    open var onmute: (Event) -> dynamic?
    open var onunmute: (Event) -> dynamic?
    open val readyState: MediaStreamTrackState
    open var onended: (Event) -> dynamic?
    open var onoverconstrained: (Event) -> dynamic?
    fun
            clone(): MediaStreamTrack
    fun
            stop()
    fun
            getCapabilities(): MediaTrackCapabilities
    fun
            getConstraints(): MediaTrackConstraints
    fun
            applyConstraints(constraints: MediaTrackCapabilities = definedExternally): Promise<Unit>
}

/**
 * Exposes the JavaScript
 * [MediaTrackSupportedConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSupportedConstraints) to Kotlin
 */
public external interface MediaTrackSupportedConstraints {
    var width: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var height: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var facingMode: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var resizeMode: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var volume: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var sampleSize: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var echoCancellation: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var autoGainControl: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var latency: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var groupId: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    fun
            getSupportedConstraints(): MediaTrackSupportedConstraints
}

@kotlin.internal.InlineOnly
public inline fun
            MediaTrackSupportedConstraints(width: Boolean? /* = true */
            height: Boolean? /* = true */
            aspectRatio: Boolean? /* = true */
            frameRate: Boolean? /* = true */
            facingMode: Boolean? /* = true */
            resizeMode: Boolean? /* = true */
            volume: Boolean? /* = true */
            sampleRate: Boolean? /* = true */
            sampleSize: Boolean? /* = true */
            echoCancellation: Boolean? /* = true */
            autoGainControl: Boolean? /* = true */
            latency: Boolean? /* = true */
            channelCount: Boolean? /* = true */
            deviceId: Boolean? /* = true */
            groupId: Boolean? /* = true */
            @null


    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceID
    o["groupId"] = groupId
    return o
}

public external interface MediaTrackCapabilities {
    var width: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var height: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var facingMode: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
    var resizeMode: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
    var volume: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var sampleSize: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var echoCancellation: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var autoGainControl: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var noiseSuppression: Array<Boolean>?
        get() = definedExternally
        set(value) = definedExternally
    var latency: DoubleRange?
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: ULongRange?
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: String?
        get() = definedExternally
        set(value) = definedExternally
    var groupId: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceID
    o["groupId"] = groupId
    return o
}

/**
 * Exposes the JavaScript
 [MediaTrackConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackConstraints) to Kotlin
 */
public external interface MediaTrackConstraints : MediaTrackConstraintSet {
    var advanced: Array<MediaTrackConstraintSet>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["advanced"] = advanced
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
}
aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}

public external interface MediaTrackConstraintSet {
    var width: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var height: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var facingMode: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var resizeMode: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var volume: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var sampleSize: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var echoCancellation: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var autoGainControl: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var noiseSuppression: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var latency: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var groupId: dynamic
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
}

/**
 * Exposes the JavaScript [MediaTrackSettings](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSettings) to Kotlin
 */
public external interface MediaTrackSettings {
    var width: Int?
        get() = definedExternally
        set(value) = definedExternally
    var height: Int?
        get() = definedExternally
        set(value) = definedExternally
    var aspectRatio: Double?
        get() = definedExternally
        set(value) = definedExternally
    var frameRate: Double?
        get() = definedExternally
        set(value) = definedExternally
    var facingMode: String?
        get() = definedExternally
        set(value) = definedExternally
    var resizeMode: String?
        get() = definedExternally
        set(value) = definedExternally
    var volume: Double?
        get() = definedExternally
        set(value) = definedExternally
    var sampleRate: Int?
        get() = definedExternally
        set(value) = definedExternally
    var sampleSize: Int?
        get() = definedExternally
        set(value) = definedExternally
    var echoCancellation: Boolean?
        get() = definedExternally
        set(value) = definedExternally
    var autoGainControl: Boolean?
        get() = definedExternally
        set(value) = definedExternally
    var noiseSuppression: Boolean?
        get() = definedExternally
        set(value) = definedExternally
    var latency: Double?
        get() = definedExternally
        set(value) = definedExternally
    var channelCount: Int?
        get() = definedExternally
        set(value) = definedExternally
    var deviceId: String?
        get() = definedExternally
        set(value) = definedExternally
    var groupId: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun MediaTrackSettings(width:
    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
 o["frameRate"] = frameRate
 o["facingMode"] = facingMode
 o["resizeMode"] = resizeMode
 o["volume"] = volume
 o["sampleRate"] = sampleRate
 o["sampleSize"] = sampleSize
 o["echoCancellation"] = echoCancellation
 o["autoGainControl"] = autoGainControl
 o["noiseSuppression"] = noiseSuppression
 o["latency"] = latency
 o["channelCount"] = channelCount
 o["deviceId"] = deviceId
 o["groupId"] = groupId
    return o
}

/**
 * Exposes the JavaScript [MediaStreamTrackEvent](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrackEvent) to Kotlin
 */
public external open class MediaStreamTrackEvent(type: String, eventInitDict: MediaStreamTrackEventInit) : Event {
    open val track: MediaStreamTrack

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface MediaStreamTrackEventInit : EventInit {
    var track: MediaStreamTrack?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["track"] = track
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [MediaDevices](https://developer.mozilla.org/en/docs/Web/API/MediaDevices) to Kotlin
 */
public external abstract class MediaDevices : EventTarget {
    open var ondevicechange: ((Event) -> dynamic)?
    fun enumerateDevices(): Promise<Array<MediaDeviceInfo>>
    fun getSupportedConstraints(): MediaTrackSupportedConstraints
    fun getUserMedia(constraints: MediaStreamConstraints = definedExternally): Promise<MediaStream>
}

/**
 * Exposes the JavaScript [MediaDeviceInfo](https://developer.mozilla.org/en/docs/Web/API/MediaDeviceInfo) to Kotlin
 */
public external abstract class MediaDeviceInfo {
    open val deviceId: String
    open val kind: MediaDeviceKind
    open val label: String
    open val groupId: String
    fun toJSON(): dynamic
}

public external abstract class InputDeviceInfo : MediaDeviceInfo {
    fun getCapabilities(): MediaTrackCapabilities
}

/**
 * Exposes the JavaScript [MediaStreamConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaStreamConstraints) to Kotlin
 */
public external interface MediaStreamConstraints {
    var video: dynamic /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var audio: dynamic /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun MediaStreamConstraints(video: dynamic = false, audio: dynamic = false): MediaStreamConstraints {
    val o = js("({})")
    o["video"] = video
    o["audio"] = audio
    return o
}

public external interface ConstrainablePattern {
    var onoverconstrained: ((Event) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    fun getCapabilities(): Capabilities
    fun getConstraints(): Constraints
    fun getSettings(): Settings
    fun applyConstraints(constraints: Constraints = definedExternally): Promise<Unit>
}
**Antelope**

The document contains code snippets related to JavaScript and Kotlin, specifically focusing on ranges and constraints. Here is a natural representation of the exported interfaces and functions:

```kotlin
interface DoubleRange {
    var max: Double?
        get() = definedExternally
        set(value) = definedExternally
    var min: Double?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun DoubleRange(max: Double? = undefined,
min: Double? = undefined): DoubleRange {
    val o = js("({})")
    o["max"] = max
    o["min"] = min
    return o
}

public external interface ConstrainDoubleRange : DoubleRange {
    var exact: Double?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Double?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ConstrainDoubleRange(exact: Double? = undefined,
    val o = js("({})")
    o["exact"] = exact
    o["ideal"] = ideal
    o["max"] = max
    o["min"] = min
    return o
}

public external interface ULongRange {
    var max: Int?
        get() = definedExternally
        set(value) = definedExternally
    var min: Int?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ULongRange(max: Int? = undefined,
min: Int? = undefined): ULongRange {
    val o = js("({})")
    o["max"] = max
    o["min"] = min
    return o
}

public external interface ConstrainULongRange : ULongRange {
    var exact: Int?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Int?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ConstrainULongRange(exact: Int? = undefined,
    val o = js("({})")
    o["exact"] = exact
    o["ideal"] = ideal
    o["max"] = max
    o["min"] = min
    return o
}

public external interface ConstrainBooleanParameters {
    var exact: Boolean?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Boolean?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ConstrainBooleanParameters(exact: Boolean? = undefined,
ideal: Boolean? = undefined): ConstrainBooleanParameters {
    val o = js("({})")
    o["exact"] = exact
    o["ideal"] = ideal
    return o
}

public external interface ConstrainDOMStringParameters {
    var exact: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var ideal: dynamic
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ConstrainDOMStringParameters(exact: dynamic = undefined,
ideal: dynamic = undefined): ConstrainDOMStringParameters {
    val o = js("({})")
    o["exact"] = exact
    o["ideal"] = ideal
    return o
}

public external interface ConstraintSet

@kotlin.internal.InlineOnly
public inline fun ConstraintSet(): ConstraintSet {
    val o = js("({})")
    return o
}

public external interface Constraints : ConstraintSet {
    var advanced: Array<ConstraintSet>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun Constraints(advanced: Array<ConstraintSet>? = undefined): Constraints {
    val o = js("({})")
    o["advanced"] = advanced
    return o
}

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface MediaStreamTrackState {
    companion object
}

public inline val MediaStreamTrackState.Companion.LIVE: MediaStreamTrackState get() = "live".asDynamic().unsafeCast<MediaStreamTrackState>()

public inline val MediaStreamTrackState.Companion.ENDED: MediaStreamTrackState get() = "ended".asDynamic().unsafeCast<MediaStreamTrackState>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ConstraintSet

@kotlin.internal.InlineOnly
public inline fun ConstraintSet(): ConstraintSet {
    val o = js("({})")
    return o
}

public external interface Constraints : ConstraintSet {
    var advanced: Array<ConstraintSet>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun Constraints(advanced: Array<ConstraintSet>? = undefined): Constraints {
    val o = js("({})")
    o["advanced"] = advanced
    return o
}

// Exposes the JavaScript

[doubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin

```
open inline val VideoFacingModeEnum.Companion.USER: VideoFacingModeEnum get() = "user".asDynamic().unsafeCast<VideoFacingModeEnum>()


open inline val VideoFacingModeEnum.Companion.LEFT: VideoFacingModeEnum get() = "left".asDynamic().unsafeCast<VideoFacingModeEnum>()

open inline val VideoFacingModeEnum.Companion.RIGHT: VideoFacingModeEnum get() = "right".asDynamic().unsafeCast<VideoFacingModeEnum>()

open inline val VideoResizeModeEnum.Companion.NONE: VideoResizeModeEnum get() = "none".asDynamic().unsafeCast<VideoResizeModeEnum>()


open inline val MediaDeviceKind.Companion.AUDIOINPUT: MediaDeviceKind get() = "audioinput".asDynamic().unsafeCast<MediaDeviceKind>()

open inline val MediaDeviceKind.Companion.AUDIOOUTPUT: MediaDeviceKind get() = "audiooutput".asDynamic().unsafeCast<MediaDeviceKind>()

open inline val MediaDeviceKind.Companion.VIDEOINPUT: MediaDeviceKind get() = "videoinput".asDynamic().unsafeCast<MediaDeviceKind>()

public external class MediaSource : EventTarget, MediaProvider
{   open val sourceBuffers: SourceBufferList
    open val activeSourceBuffers: SourceBufferList
    open val readyState: ReadyState
    var duration: Double
    var onsourceopen: ((Event) -> dynamic)?
    var onsourceended: ((Event) -> dynamic)?
    var onsourceclose: ((Event) -> dynamic)?
    fun addSourceBuffer(type: String): SourceBuffer
    fun removeSourceBuffer(sourceBuffer: SourceBuffer)
    fun endOfStream(error: EndOfStreamError = definedExternally)
    fun setLiveSeekableRange(start: Double, end: Double)
    fun clearLiveSeekableRange()

    companion object {
        fun isTypeSupported(type: String): Boolean
    }
}

public abstract class SourceBuffer : EventTarget
{   open var mode: AppendMode
    open val updating: Boolean
    open val buffered: TimeRanges
    open var timestampOffset: Double
    open val audioTracks: AudioTrackList
    open val videoTracks: VideoTrackList
    open val textTracks: TextTrackList
    open var appendWindowStart: Double
    open var appendWindowEnd: Double
    open var onupdate: ((Event) -> dynamic)?
    open var onsourceopen: ((Event) -> dynamic)?
    open var onsourceended: ((Event) -> dynamic)?
    open var onabort: ((Event) -> dynamic)?
    fun appendBuffer(data: dynamic)
    fun abort()
    fun remove(start: Double, end: Double)
}

public abstract class SourceBufferList : EventTarget
{   open val length: Int
    open var onaddsourcebuffer: ((Event) -> dynamic)?
    open var onremovesourcebuffer: ((Event) -> dynamic)?
}

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface VideoResizeModeEnum {
    companion object
}

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface MediaDeviceKind {
    companion object
}

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.dom.mediasource

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*
interface ReadyState {
    companion object
}

public inline val ReadyState.Companion.CLOSED: ReadyState
    get() = "closed".asDynamic().unsafeCast<ReadyState>()

public inline val ReadyState.Companion.OPEN: ReadyState
    get() = "open".asDynamic().unsafeCast<ReadyState>()

public inline val ReadyState.Companion.ENDED: ReadyState
    get() = "ended".asDynamic().unsafeCast<ReadyState>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface EndOfStreamError {
    companion object
}

public inline val EndOfStreamError.Companion.NETWORK: EndOfStreamError
    get() = "network".asDynamic().unsafeCast<EndOfStreamError>()

public inline val EndOfStreamError.Companion.DECODE: EndOfStreamError
    get() = "decode".asDynamic().unsafeCast<EndOfStreamError>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface AppendMode {
    companion object
}

public inline val AppendMode.Companion.SEGMENTS: AppendMode
    get() = "segments".asDynamic().unsafeCast<AppendMode>()

public inline val AppendMode.Companion.SEQUENCE: AppendMode
    get() = "sequence".asDynamic().unsafeCast<AppendMode>()


o["shiftKey"] = shiftKey
  o["altKey"] = altKey
  o["metaKey"] = metaKey
  o["modifierAltGraph"] = modifierAltGraph
  o["modifierCapsLock"] = modifierCapsLock
  o["modifierFn"] = modifierFn
  o["modifierFnLock"] = modifierFnLock
  o["modifierHyper"] = modifierHyper
  o["modifierNumLock"] = modifierNumLock
  o["modifierScrollLock"] = modifierScrollLock
  o["modifierSuper"] = modifierSuper
  o["modifierSymbol"] = modifierSymbol
  o["modifierSymbolLock"] = modifierSymbolLock
  o["view"] = view
  o["detail"] = detail
  o["bubbles"] = bubbles
  o["cancelable"] = cancelable
  o["composed"] = composed
  return o
}

/**
 * Exposes the JavaScript
 [PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent) to Kotlin
 */
public external open class PointerEvent(type: String, eventInitDict: PointerEventInit = definedExternally) : MouseEvent {
  open val pointerId: Int
  open val width: Double
  open val height: Double
  open val pressure: Float
  open val tangentialPressure: Float
  open val tiltX: Int
  open val tiltY: Int
  open val twist: Int
  open val pointerType: String
  open val isPrimary: Boolean

  companion object {
    val NONE: Short
    val CAPTURING_PHASE: Short
    val AT_TARGET: Short
    val BUBBLING_PHASE: Short
  }
}

/*
 Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.dom.svg

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript
 [SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin
 */
public external abstract class SVGElement : Element, ElementCSSInlineStyle, GlobalEventHandlers, SVGElementInstance {

  open val dataset: DOMStringMap
  open val ownerSVGElement: SVGSVGElement?
  open val viewportElement: SVGElement?
  open var tabIndex: Int
      fun focus()
      fun blur()

  companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short

    val SVGBoundingBoxOptions {
      var fill: Boolean? /* = true */
          get() = definedExternally
          set(value) = definedExternally
      var stroke: Boolean? /* = false */
          get() = definedExternally
          set(value) = definedExternally
      var markers: Boolean? /* = false */
          get() = definedExternally
          set(value) = definedExternally
      var clipped: Boolean? /* = false */
          get() = definedExternally
          set(value) = definedExternally
  }

  @kotlin.internal.InlineOnly
    val o = js("({})")
    o["fill"] = fill
    o["stroke"] = stroke
    o["markers"] = markers
    o["clipped"] = clipped
    return o
  }

  /**
   * Exposes the JavaScript
   [SVGGraphicsElement](https://developer.mozilla.org/en/docs/Web/API/SVGGraphicsElement) to Kotlin
   */
  public external abstract class SVGGraphicsElement : SVGElement, SVGTests {
  }
fun clear()
fun initialize(newItem: SVGNumber): SVGNumber
fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
fun removeItem(index: Int): SVGNumber
fun appendItem(newItem: SVGNumber): SVGNumber
fun getItem(index: Int): SVGNumber

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript [SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGLengthList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGLength): SVGLength
    fun insertItemBefore(newItem: SVGLength, index: Int): SVGLength
    fun replaceItem(newItem: SVGLength, index: Int): SVGLength
    fun removeItem(index: Int): SVGLength
    fun appendItem(newItem: SVGLength): SVGLength
    fun getItem(index: Int): SVGLength
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGLength? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGLength) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGAnimatedAlgorithm](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAlgorithm) to Kotlin
 */
public external abstract class SVGAnimatedAlgorithm {
    open var baseVal: Number
    open val animVal: Number
}

/**
 * Exposes the JavaScript
 [SVGAnimatedList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedList) to Kotlin
 */
public external abstract class SVGAnimatedList {
    open val baseVal: Array<SVGNumber>
    open val animVal: Array<SVGNumber>
}

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript
 [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber) { asDynamic()[index] = newItem }
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}
Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}
}

public external abstract class SVGUnknownElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGDefsElement](https://developer.mozilla.org/en/docs/Web/API/SVGDefsElement) to Kotlin
 */
public external abstract class SVGDefsElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGDescElement](https://developer.mozilla.org/en/docs/Web/API/SVGDescElement) to Kotlin
 */
public external abstract class SVGDescElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGMetadataElement](https://developer.mozilla.org/en/docs/Web/API/SVGMetadataElement) to Kotlin
 */
public external abstract class SVGMetadataElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
**Exposes the JavaScript [SVGTitleElement](https://developer.mozilla.org/en/docs/Web/API/SVGTitleElement) to Kotlin**

```kotlin
public external abstract class SVGTitleElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

**Exposes the JavaScript [SVGSymbolElement](https://developer.mozilla.org/en/docs/Web/API/SVGSymbolElement) to Kotlin**

```kotlin
public external abstract class SVGSymbolElement : SVGGraphicsElement, SVGFitToViewBox {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

**Exposes the JavaScript [SVGUseElement](https://developer.mozilla.org/en/docs/Web/API/SVGUseElement) to Kotlin**

```kotlin
public external abstract class SVGUseElement : SVGGraphicsElement, SVGURIReference {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open val instanceRoot: SVGElement?
    open val animatedInstanceRoot: SVGElement?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

**Exposes the JavaScript [SVGSymbolElement](https://developer.mozilla.org/en/docs/Web/API/SVGSymbolElement) to Kotlin**

```kotlin
public external abstract class SVGSymbolElement : SVGSymbolElement, SVGSymbolElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

**Exposes the JavaScript [SVGUseElement](https://developer.mozilla.org/en/docs/Web/API/SVGUseElement) to Kotlin**

```kotlin
public external open class SVGUseElementShadowRoot : ShadowRoot {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

```
public external interface SVGElementInstance {
    val correspondingElement: SVGElement?
        get() = definedExternally
    val correspondingUseElement: SVGUseElement?
        get() = definedExternally
}

public external open class ShadowAnimation(source: dynamic, newTarget: dynamic) {
    open val sourceAnimation: dynamic
}

/**
 * Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement) to Kotlin */
public external abstract class SVGSwitchElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

public external interface GetSVGDocument {
    fun getSVGDocument(): Document
}

/**
 * Exposes the JavaScript [SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin */
public external abstract class SVGStyleElement : SVGElement, LinkStyle {
    open var type: String
    open var media: String
    open var title: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript [SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin */
public external abstract class SVGTransform {
    open val type: Short
    open val matrix: DOMMatrix
    open val angle: Float
    fun setMatrix(matrix: DOMMatrixReadOnly)
    fun setTranslate(tx: Float, ty: Float)
    fun setScale(sx: Float, sy: Float)
    fun setRotate(angle: Float, cx: Float, cy: Float)
    fun setSkewX(angle: Float)
    fun setSkewY(angle: Float)

    companion object {
        val SVG_TRANSFORM_UNKNOWN: Short
        val SVG_TRANSFORM_MATRIX: Short
        val SVG_TRANSFORM_TRANSLATE: Short
        val SVG_TRANSFORM_SCALE: Short
        val SVG_TRANSFORM_ROTATE: Short
        val SVG_TRANSFORM_SKEWX: Short
        val SVG_TRANSFORM_SKEWY: Short
    }
}

/**
 * Exposes the JavaScript [SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to Kotlin */
public external abstract class SVGAnimatedTransformList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGTransform): SVGTransform
    fun insertItemBefore(newItem: SVGTransform, index: Int): SVGTransform
    fun replaceItem(newItem: SVGTransform, index: Int): SVGTransform
    fun removeItem(index: Int): SVGTransform
    fun appendItem(newItem: SVGTransform): SVGTransform
    fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform
    fun consolidate(): SVGTransform?
    fun getItem(index: Int): SVGTransform
}

@kotlin.internal.InlineOnly
public inline operator fun SVGAnimatedTransformList.get(index: Int): SVGTransform = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGAnimatedTransformList.set(index: Int, newItem: SVGTransform) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript [SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to Kotlin */

Open Source Used In DNAC Platform Antelope 9652
Kotlin

```kotlin
public external abstract class SVGAnimatedTransformList {
    open val baseVal: SVGTransformList
    open val animVal: SVGTransformList
}
```

```kotlin
/**
 * Exposes the JavaScript [SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin
 */
public external abstract class SVGPreserveAspectRatio {
    open var align: Short
    open var meetOrSlice: Short

    companion object {
        val SVG_PRESERVEASPECTRATIO_UNKNOWN: Short
        val SVG_PRESERVEASPECTRATIO_NONE: Short
        val SVG_PRESERVEASPECTRATIO_XMINYMIN: Short
        val SVG_PRESERVEASPECTRATIO_XMAXYMAX: Short
        val SVG_PRESERVEASPECTRATIO_XMINYMID: Short
        val SVG_PRESERVEASPECTRATIO_XMIDYMID: Short
        val SVG_PRESERVEASPECTRATIO_XMAXYMID: Short
        val SVG_PRESERVEASPECTRATIO_XMINYMAX: Short
        val SVG_PRESERVEASPECTRATIO_XMIDYMAX: Short
        val SVG_PRESERVEASPECTRATIO_XMIDYMIN: Short
        val SVG_MEETORSLICE_UNKNOWN: Short
        val SVG_MEETORSLICE_MEET: Short
        val SVG_MEETORSLICE_SLICE: Short
    }
}
```

```kotlin
/**
 * Exposes the JavaScript [SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspectRatio) to Kotlin
 */
public external abstract class SVGAnimatedPreserveAspectRatio {
    open val baseVal: SVGPreserveAspectRatio
    open val animVal: SVGPreserveAspectRatio
}
```

```kotlin
/**
 * Exposes the JavaScript [SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin
 */
public external abstract class SVGPathElement : SVGGeometryElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

```kotlin
/**
 * Exposes the JavaScript [SVGCircleElement](https://developer.mozilla.org/en/docs/Web/API/SVGCircleElement) to Kotlin
 */
public external abstract class SVGCircleElement : SVGGeometryElement {
    open val cx: SVGAnimatedLength
    open val cy: SVGAnimatedLength
    open val r: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```
/**
 * Exposes the JavaScript [SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin
 */
public external abstract class SVGEllipseElement : SVGGeometryElement {
    open val cx: SVGAnimatedLength
    open val cy: SVGAnimatedLength
    open val rx: SVGAnimatedLength
    open val ry: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin
 */
public external abstract class SVGLineElement : SVGGeometryElement {
    open val x1: SVGAnimatedLength
    open val y1: SVGAnimatedLength
    open val x2: SVGAnimatedLength
    open val y2: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin
 */
public external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGAnimatedPoints](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPoints) to Kotlin
 */
public external interface SVGAnimatedPoints {
    val points: SVGPointList
    val animatedPoints: SVGPointList
}

public external abstract class SVGPointList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: DOMPoint): DOMPoint
    fun insertItemBefore(newItem: DOMPoint, index: Int): DOMPoint
    fun replaceItem(newItem: DOMPoint, index: Int): DOMPoint
    fun removeItem(index: Int): DOMPoint
    fun appendItem(newItem: DOMPoint):
DOMPoint

fun getItem(index: Int): DOMPoint

fun getComputedTextLength(): Float

fun getSubStringLength(charnum: Int, nchars: Int): Float

fun getStartPositionOfChar(charnum: Int): DOMPoint

fun getEndPositionOfChar(charnum: Int): DOMPoint

fun getExtentOfChar(charnum: Int): DOMRect

fun getRotationOfChar(charnum: Int): Float

fun getCharNumAtPosition(point: DOMPoint): Int

fun selectSubString(charnum: Int, nchars: Int)

// Exposes the JavaScript

[SVGPolylineElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolylineElement) to Kotlin

*An external abstract class SVGPolylineElement : SVGGeometryElement, SVGAnimatedPoints

**Exposes the JavaScript [SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin

*An external abstract class SVGPathElement : SVGGeometryElement, SVGAnimatedPath

**Exposes the JavaScript [SVGTextContentElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextContentElement) to Kotlin

*An external abstract class SVGTextContentElement : SVGGraphicsElement

**Exposes the JavaScript [SVGTextPositioningElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPositioningElement) to Kotlin

*An external abstract class SVGTextPositioningElement : SVGElement

### Companion Objects

<table>
<thead>
<tr>
<th>Class</th>
<th>NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEMENT_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>ATTRIBUTE_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>TEXT_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>CDATA_SECTION_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>ENTITY_REFERENCE_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENT_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENT_TYPE_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENT_FRAGMENT_NODE</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENTPOSITION_DISCONNECTED</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENTPOSITION_PRECEDING</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENTPOSITION_FOLLOWING</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENTPOSITION_CONTAINS</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENTPOSITION_CONTAINED_BY</td>
<td>Short</td>
</tr>
<tr>
<td>DOCUMENTPOSITION_IMPLEMENTATIONSPECIFIC</td>
<td>Short</td>
</tr>
</tbody>
</table>
val LENGTHADJUST_UNKNOWN: Short
val LENGTHADJUST_SPACING: Short
val LENGTHADJUST_SPACINGANDGLYPHS: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGTextElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextElement) to Kotlin
 */
public external abstract class SVGTextElement : SVGTextPositioningElement {
    companion object {
        val LENGTHADJUST_UNKNOWN: Short
        val LENGTHADJUST_SPACING: Short
        val LENGTHADJUST_SPACINGANDGLYPHS: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGTSpanElement](https://developer.mozilla.org/en/docs/Web/API/SVGTSpanElement) to Kotlin
 */
public external abstract class SVGTSpanElement : SVGTextPositioningElement {
    companion object {
        val LENGTHADJUST_UNKNOWN: Short
        val LENGTHADJUST_SPACING: Short
        val LENGTHADJUST_SPACINGANDGLYPHS: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGTextPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPathElement) to Kotlin
 */
public external abstract class SVGTextPathElement : SVGTextContentElement, SVGURIReference {
    open val startOffset: SVGAnimatedLength
    open val method: SVGAnimatedEnumeration
    open val spacing: SVGAnimatedEnumeration

    companion object {
        val TEXTPATH_METHODTYPE_UNKNOWN: Short
        val TEXTPATH_METHODTYPE_ALIGN: Short
        val TEXTPATH_METHODTYPE_STRETCH: Short
        val TEXTPATH_SPACINGTYPE_UNKNOWN: Short
        val TEXTPATH_SPACINGTYPE_AUTO: Short
        val TEXTPATH_SPACINGTYPE_EXACT: Short
        val LENGTHADJUST_UNKNOWN: Short
        val LENGTHADJUST_SPACING: Short
        val LENGTHADJUST_SPACINGANDGLYPHS: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
/**
 * Exposes the JavaScript [SVGImageElement](https://developer.mozilla.org/en/docs/Web/API/SVGImageElement) to Kotlin
 */
public external abstract class SVGImageElement : SVGGraphicsElement, SVGURIReference,
HTMLOrSVGImageElement {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open
    val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open val preserveAspectRatio:
SVGAnimatedPreserveAspectRatio
open var crossOrigin: String?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGForeignObjectElement](https://developer.mozilla.org/en/docs/Web/API/SVGForeignObjectElement) to Kotlin
 */
public external abstract class SVGForeignObjectElement : SVGGraphicsElement {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGMarkerElement : SVGElement, SVGFitToViewBox {
    open val refX: SVGAnimatedLength
    open val refY: SVGAnimatedLength
    open val markerUnits: SVGAnimatedEnumeration
    open val markerWidth: SVGAnimatedLength
    open val markerHeight: SVGAnimatedLength
    open val orientType: SVGAnimatedEnumeration
    open val orientAngle: SVGAnimatedAngle
    open var orient: String
    fun setOrientToAuto()
    fun setOrientToAngle(angle: SVGAngle)

    companion object {
        val SVG_MARKERUNITS_UNKNOWN: Short
        val SVG_MARKERUNITS_USERSPACEONUSE: Short
        val SVG_MARKERUNITS_STROKEWIDTH: Short
        val SVG_MARKER_ORIENT_UNKNOWN: Short
        val SVG_MARKER_ORIENT_AUTO: Short
        val SVG_MARKER_ORIENT_ANGLE: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGStopElement](https://developer.mozilla.org/en/docs/Web/API/SVGStopElement) to Kotlin
 */
public external abstract class SVGStopElement : SVGElement {
    open val offset: SVGAnimatedNumber

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGPatternElement](https://developer.mozilla.org/en/docs/Web/API/SVGPatternElement) to Kotlin
 */
public external abstract class SVGPatternElement : SVGElement, SVGFitToViewBox, SVGURIReference, SVGUnitTypes {
    open val patternUnits: SVGAnimatedEnumeration
    open val patternContentUnits: SVGAnimatedEnumeration
    open val patternTransform: SVGAnimatedTransformList
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength

    companion object {
        val SVG_UNIT_TYPE_UNKNOWN: Short
        val SVG_UNIT_TYPE_USERSPACEONUSE: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGHatchElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class SVGHatchpathElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin
 */
public external abstract class SVGElement : SVGElement, SVGURIReference {
    open val viewBox: SVGAnimatedString
    open val originX: SVGAnimatedLength
    open val originY: SVGAnimatedLength
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGCursorElement](https://developer.mozilla.org/en/docs/Web/API/SVGCursorElement) to Kotlin
 */
public external abstract class SVGCursorElement : SVGElement, SVGURIReference {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGScriptElement](https://developer.mozilla.org/en/docs/Web/API/SVGScriptElement) to Kotlin
 */
public external abstract class SVGScriptElement : SVGElement, SVGURIReference, HTMLOrSVGScriptElement {
    open var type: String
    open var crossOrigin: String?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGAElement](https://developer.mozilla.org/en/docs/Web/API/SVGAElement) to Kotlin
 */
public external abstract class SVGAElement : SVGGraphicsElement, SVGURIReference {
    open val target: SVGAnimatedString
    open val download: SVGAnimatedString
    open val rel: SVGAnimatedString
    open val relList: SVGAnimatedString
    open val hreflang: SVGAnimatedString
    open val type: SVGAnimatedString

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGViewElement](https://developer.mozilla.org/en/docs/Web/API/SVGViewElement) to Kotlin
 */
public external abstract class SVGViewElement : SVGElement, SVGFitToViewBox, SVGZoomAndPan {
    companion object {
        val SVG_ZOOMANDPAN_UNKNOWN: Short
        val SVG_ZOOMANDPAN_DISABLE: Short
        val SVG_ZOOMANDPAN_MAGNIFY: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.performance.*
import org.w3c.workers.*

/**
 * Exposes the JavaScript [Notification](https://developer.mozilla.org/en/docs/Web/API/Notification) to Kotlin
 */
public external open class Notification(title: String, options: NotificationOptions = definedExternally) : EventTarget {
    var onclick: ((MouseEvent) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    open val title: String
    open val dir: NotificationDirection
    open val lang: String
    open val body: String
    open val tag: String
    open val image: String
    open val icon: String
    open val badge: String
    open val sound: String
    open val vibrate: dynamic
    open val timestamp: Number
    open val renotify: Boolean
    open val silent: Boolean
    open val noscreen: Boolean
    open val requireInteraction: Boolean
    open val open: Boolean
    open val data: Any?
    open val actions: Array<out NotificationAction>
    fun close()

    companion object {
        val permission: NotificationPermission
        val maxActions: Int
        fun requestPermission(deprecatedCallback: (NotificationPermission) -> Unit = definedExternally): Promise<NotificationPermission>
    }
}

public external interface NotificationOptions {
    var dir: NotificationDirection? /* = NotificationDirection.AUTO */
    var lang: String? /* = "" */
    var body: String? /* = "" */
    var tag: String? /* = "" */
    var image: String?
    var icon: String?
    var badge: String?
    var sound: String?
    var vibrate: dynamic
    var timestamp: Number?
    var renotify: Boolean? /* = false */
    var silent: Boolean? /* = false */
    var noscreen: Boolean? /* = false */
    var requireInteraction: Boolean? /* = false */
    var sticky: Boolean? /* = false */
    var data: Any? /* = null */
    var actions: Array<NotificationAction>? /* = arrayOf() */
}

public external interface NotificationAction {
    var action: String?
    var title: String?
    var icon: String?
}

public external interface GetNotificationOptions {
    var tag: String? /* = "" */
}

@kotlin.internal.InlineOnly
public inline fun get() = definedExternally
set(value) = definedExternally
var lang: String? */ = \"\" */
get() = definedExternally
set(value) = definedExternally
var body: String? */ = \"\" */
get() = definedExternally
set(value) = definedExternally
var tag: String? */ = \"\" */
get() = definedExternally
set(value) = definedExternally
var image: String? */ = \"\" */
get() = definedExternally
set(value) = definedExternally
var icon: String? */ = \"\" */
get() = definedExternally
set(value) = definedExternally
var badge: String? */ = \"\" */
get() = definedExternally
set(value) = definedExternally
var sound: String? */ = \"\" */
get() = definedExternally
set(value) = definedExternally
var vibrate: dynamic
get() = definedExternally
set(value) = definedExternally
var timestamp: Number? */ = \"\" */
get() = definedExternally
set(value) = definedExternally
var renotify: Boolean? */ = false */
get() = definedExternally
set(value) = definedExternally
var silent: Boolean? */ = false */
get() = definedExternally
set(value) = definedExternally
var noscreen: Boolean? */ = false */
get() = definedExternally
set(value) = definedExternally
var requireInteraction: Boolean? */ = false */
get() = definedExternally
set(value) = definedExternally
var data: Any? */ = null */
get() = definedExternally
set(value) = definedExternally
var actions: Array<NotificationAction>? */ = arrayOf() */
get() = definedExternally
set(value) = definedExternally
var actions: Array<NotificationAction>? */ = arrayOf() */
get() = definedExternally
set(value) = definedExternally

GetNotificationOptions {
    val o = js("({})")
    o["tag"] = tag
    return o
}

/**
 * Exposes the
 * JavaScript [NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin
 */
public external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) :

ExtendableEvent {
    open val notification: Notification
    open val action: String

    companion object {
        val NONE: Short = 0
        val CAPTURING_PHASE: Short = 1
        val AT_TARGET: Short = 2
        val BUBBLING_PHASE: Short = 3
    }
}

public external interface NotificationEventInit : ExtendableEventInit {
    var notification: Notification?
        get() = definedExternally
        set(value) = definedExternally
    var action: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["notification"] = notification
    o["action"] = action
    o["bubbles"] = bubbles
    o["Cancelable"] = cancelable
    o["Composed"] = composed
    return o
}

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NotificationPermission {
    companion object
}

public inline val NotificationPermission.Companion.DEFAULT: NotificationPermission get() = "default".asDynamic().unsafeCast<NotificationPermission>()

public inline val NotificationPermission.Companion.DENIED: NotificationPermission get() = "denied".asDynamic().unsafeCast<NotificationPermission>()

public inline val NotificationPermission.Companion.GRANTED: NotificationPermission get() = "granted".asDynamic().unsafeCast<NotificationPermission>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NotificationDirection {
    companion object
}

public inline val NotificationDirection.Companion.AUTO: NotificationDirection get() = "auto".asDynamic().unsafeCast<NotificationDirection>()

public inline val NotificationDirection.Companion.LTR: NotificationDirection get() = "ltr".asDynamic().unsafeCast<NotificationDirection>()

public inline val NotificationDirection.Companion.RTL: NotificationDirection get() = "rtl".asDynamic().unsafeCast<NotificationDirection>()

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
 * found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.workers

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.performance.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript [ServiceWorker](https://developer.mozilla.org/en/docs/Web/API/ServiceWorker) to Kotlin
 */
public external abstract class ServiceWorker : EventTarget, AbstractWorker, UnionMessagePortOrServiceWorker, UnionClientOrMessagePortOrServiceWorker {
    open val scriptURL: String
    open val state: ServiceWorkerState
    open var onstatechange: ((Event) -> dynamic)?
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}

/**
 * Exposes the JavaScript [ServiceWorkerRegistration](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerRegistration) to Kotlin
 */
public external abstract class ServiceWorkerRegistration : EventTarget {
    open val installing: ServiceWorker?
    open val waiting: ServiceWorker?
    open val active: ServiceWorker?
    open val scope: String
    open var onupdatefound: ((Event) -> dynamic)?
    fun update(): Promise<Unit>
    fun unregister(): Promise<Boolean>
    fun showNotification(title: String, options: NotificationOptions = definedExternally): Promise<Unit>
    fun getNotifications(filter: GetNotificationOptions = definedExternally):
Clients {
    fun get(id: String): Promise<Any?>
    fun matchAll(options: ClientQueryOptions = definedExternally): Promise<Array<Client>>
    fun openWindow(url: String): Promise<ClientWindow>?
    fun claim(): Promise<Unit>
}

public external interface ClientQueryOptions {
    var includeUncontrolled: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var type: ClientType? /* = ClientType.WINDOW */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ClientQueryOptions(includeUncontrolled: Boolean? = false, type: ClientType? = ClientType.WINDOW): ClientQueryOptions {
    val o = js("({})")
    o["includeUncontrolled"] = includeUncontrolled
    o["type"] = type
    return o
}

/**
 * Exposes the JavaScript [ExtendableEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableEvent) to Kotlin
 */
public external open class ExtendableEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : Event {
    fun waitUntil(f: Promise<Any?>)

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface ExtendableEventInit : EventInit

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [InstallEvent](https://developer.mozilla.org/en/docs/Web/API/InstallEvent) to Kotlin
 */
public external open class InstallEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : ExtendableEvent {
    fun registerForeignFetch(options: ForeignFetchOptions)

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface ForeignFetchOptions {
    var scopes: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
    var origins: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ForeignFetchOptions(scopes: Array<String>?, origins: Array<String>?){
    val o = js("({})")
    o["scopes"] = scopes
    o["origin"] = origins
    return o
}

/**
 * Exposes the JavaScript [FetchEvent](https://developer.mozilla.org/en/docs/Web/API/FetchEvent) to Kotlin
 */
public external open class FetchEvent(type: String, eventInitDict: FetchEventInit) : ExtendableEvent {
    open val request: Request
    open val clientId: String?
    open val isReload: Boolean
    fun respondWith(r: Promise<Response>)

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface FetchEventInit : ExtendableEventInit {
    var request: Request?
        get() = definedExternally
        set(value) = definedExternally
    var clientId: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var isReload: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["request"] = request
    o["clientId"] = clientId
    o["isReload"] = isReload
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

public external interface ForeignFetchResponse {
    var response: Response?
        get() = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ForeignFetchResponse(response: Response?): ForeignFetchResponse {
    val o = js("({})")
    o["response"] = response
    return o
}
abstract class CacheStorage {
    fun match(request: dynamic, options: CacheQueryOptions = definedExternally):
    Promise<Any?>
    fun has(cacheName: String): Promise<Cache>
    fun open(cacheName: String): Promise<Cache>
    fun delete(cacheName: String): Promise<Cache>
    fun keys(): Array<String>
}

public external open class FunctionalEvent : ExtendableEvent {
    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface UnionMessagePortOrServiceWorker

public external interface UnionClientOrMessagePortOrServiceWorker

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ServiceWorkerState {
    companion object
}

inline val ServiceWorkerState.Companion.INSTALLING: ServiceWorkerState get() = "installing".asDynamic().unsafeCast<ServiceWorkerState>()

inline val ServiceWorkerState.Companion.INSTALLED: ServiceWorkerState get() = "installed".asDynamic().unsafeCast<ServiceWorkerState>()

inline val ServiceWorkerState.Companion.ACTIVATING: ServiceWorkerState get() = "activating".asDynamic().unsafeCast<ServiceWorkerState>()

inline val ServiceWorkerState.Companion.ACTIVATED: ServiceWorkerState get() = "activated".asDynamic().unsafeCast<ServiceWorkerState>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface FrameType {
    companion object
}

inline val FrameType.Companion.AUXILIARY: FrameType get() = "auxiliary".asDynamic().unsafeCast<FrameType>()

inline val FrameType.Companion.TOP_LEVEL: FrameType get() = "top-level".asDynamic().unsafeCast<FrameType>()

inline val FrameType.Companion.NESTED: FrameType get() = "nested".asDynamic().unsafeCast<FrameType>()

inline val FrameType.Companion.NONE: FrameType get() = "none".asDynamic().unsafeCast<FrameType>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ClientType {
    companion object
}

inline val ClientType.Companion.WINDOW: ClientType get() = "window".asDynamic().unsafeCast<ClientType>()

inline val ClientType.Companion.WORKER: ClientType get() = "worker".asDynamic().unsafeCast<ClientType>()

inline val ClientType.Companion.SHAREDWORKER: ClientType get() = "sharedworker".asDynamic().unsafeCast<ClientType>()

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.xhr

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*

/**
 * Exposes the JavaScript [XMLHttpRequestEventTarget](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequestEventTarget) to Kotlin
 */
public external abstract class XMLHttpRequestEventTarget : EventTarget {
    open var onloadstart: ((ProgressEvent) -> dynamic)?
    open var onprogress: ((ProgressEvent) -> dynamic)?
    open var onabort: ((Event) -> dynamic)?
    open var onerror: ((Event) -> dynamic)?
    open var onload: ((Event) -> dynamic)?
    open var ontimeout: ((Event) -> dynamic)?
    open var onloadend: ((Event) -> dynamic)?
}

public external abstract class XMLHttpRequestUpload : XMLHttpRequestEventTarget

/**
 * Exposes the JavaScript [XMLHttpRequest](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequest) to Kotlin
 */
public
external open class XMLHttpRequest : XMLHttpRequestEventTarget {
    var onreadystatechange: ((Event) -> dynamic)?
    open val readyState: Short
    var timeout: Int
    var withCredentials: Boolean
    open val upload: XMLHttpRequestUpload
    open val responseURL: String
    open val status: Short
    open val statusText: String
    var responseType: XMLHttpRequestResponseType
    open val response: Any?
    open val responseText: String
    open val responseXML: Document?
    fun open(method: String, url: String)
    fun setRequestHeader(name: String, value: String)
    fun send(body: dynamic = definedExternally)
    fun abort() = definedExternally
    fun getResponseHeader(name: String): String?
    fun getAllResponseHeaders(): Array<dynamic>
    fun overrideMimeType(mime: String)

    companion object {
        val UNSENT: Short
        val OPENED: Short
        val HEADERS_RECEIVED: Short
        val LOADING: Short
        val DONE: Short
    }
}

/**
 * Exposes the JavaScript
 [FormData](https://developer.mozilla.org/en/docs/Web/API/FormData) to Kotlin
 */
public external open class FormData(form: HTMLFormElement = definedExternally) {
    fun append(name: String, value: String)
    fun append(name: String, value: Blob, filename: String = definedExternally)
    fun delete(name: String)
    fun get(name: String): dynamic
    fun getAll(name: String): Array<dynamic>
    fun has(name: String): Boolean
    fun set(name: String, value: String)
    fun set(name: String, value: Blob, filename: String = definedExternally)

    companion object {
        // ...
    }
}

/**
 * Exposes the JavaScript
 [ProgressEvent](https://developer.mozilla.org/en/docs/Web/API/ProgressEvent) to Kotlin
 */
public external open class ProgressEvent(type: String, eventInitDict: ProgressEventInit = definedExternally) : Event {
    open val lengthComputable: Boolean
    open val loaded: Number
    open val total: Number

    companion object {
        // ...
    }

    public external interface ProgressEventInit : EventInit {
        var lengthComputable: Boolean? /* = false */
        var loaded: Number? /* = 0 */
        var total: Number? /* = 0 */
        var bubbles: Boolean? /* = false */
        var cancelable: Boolean? /* = false */
        var composed: Boolean? /* = false */
    }

    // ...
}

public external interface XMLHttpRequestResponseType {
    companion object
}

public inline val XMLHttpRequestResponseType.Companion.EMPTY: XMLHttpRequestResponseType get() = "".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

public inline val XMLHttpRequestResponseType.Companion.ARRAYBUFFER: XMLHttpRequestResponseType get() = "arraybuffer".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

public inline val XMLHttpRequestResponseType.Companion.BLOB: XMLHttpRequestResponseType get() = "blob".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

public inline val XMLHttpRequestResponseType.Companion.DOCUMENT: XMLHttpRequestResponseType get() = "document".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

public inline val XMLHttpRequestResponseType.Companion.JSON: XMLHttpRequestResponseType get() = "json".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

public inline val XMLHttpRequestResponseType.Companion.TEXT: XMLHttpRequestResponseType get() = "text".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface XMLHttpRequestResponse {
    companion object
}

/* *
 * Copyright 2010-2018 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file.
 * *
 * Open Source Used In DNAC Platform Antelope 9669
 */
open source used in DNAC Platform Antelope

kotlin.internal.RequireKotlinVersionKind

```kotlin
package kotlin

import kotlin.reflect.KClass

/**
 * Signals that the annotated annotation class is a marker of an experimental API. Any declaration annotated with that marker is considered an experimental declaration and its call sites should accept the experimental aspect of it either by using [UseExperimental] or by being annotated with that marker themselves, effectively causing further propagation of that experimental aspect. This class is deprecated in favor of a more general approach provided by [RequiresOptIn]/[OptIn].

@Target(ANNOTATION_CLASS)
@Retention(BINARY)
@SinceKotlin("1.2.50")
@RequiresOptIn("Please use RequiresOptIn instead.","")
@public annotation class Experimental(val level: Level = Level.ERROR) {
    /**
     * Severity of the diagnostic that should be reported on usages of experimental API which did not explicitly accept the experimental aspect of that API either by using [UseExperimental] or by being annotated with the corresponding marker annotation.
     */
    public enum class Level {
        WARNING,
        ERROR,
    }
}

/**
 * Allows to use experimental API denoted by the given markers in the annotated file, declaration, or expression. This class is deprecated in favor of a more general approach provided by [RequiresOptIn]/[OptIn].

@Target(
    CLASS,
    PROPERTY, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION,
    PROPERTY_GETTER, PROPERTY_SETTER, EXPRESSION, FILE,
    TYPEALIAS)
@Retention(SOURCE)
@SinceKotlin("1.2.50");
@RequiresOptIn("Please use OptIn instead.","")
@public annotation class UseExperimental(vararg val markerClass: KClass<out Annotation>)

@Target(CLASS, PROPERTY, CONSTRUCTOR, FUNCTION,
    TYPEALIAS)
@Retention(BINARY)
@MustBeDocumented
@SinceKotlin("1.3")
@public annotation class ExperimentalStdlibApi
```

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin

import kotlin.annotation.AnnotationTarget.*
import kotlin.experimental.ExperimentalTypeInference

/**
 * Allows to infer generic type arguments of a function from the calls in the annotated function parameter of that function.
 *
 * When this annotation is placed on a generic function parameter of a function, it enables to infer the type arguments of that generic function from the lambda body passed to that parameter.
 *
 * The calls that affect inference are either members of the receiver type or extensions for that type. The extensions must be themselves annotated with `@BuilderInference`.
 * Example: we declare:
 ```
 fun `<T>` sequence(@BuilderInference `block: suspend SequenceScope<`T``>.(`T`) -> Unit): Sequence<`T``
 * ` and use it like:
 ```
 val result = sequence {
```
yield("result")

* @Target(VALUE_PARAMETER, FUNCTION, PROPERTY)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.3")
@ExperimentalTypeInference
@public annotation class BuilderInference

/**
 * Enables overload selection based on the type of the value returned from lambda argument.
 * When two or more function overloads have otherwise the same parameter lists that differ only in the return type of a functional parameter, this annotation enables overload selection by the type of the value returned from the lambda function passed to this functional parameter.
 */
@Target(VALUE_PARAMETER, FUNCTION, PROPERTY)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.3")
@ExperimentalTypeInference
@public annotation class OverloadResolutionByLambdaReturnType

// Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
// Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

package kotlin

import kotlin.annotation.AnnotationTarget.*
import kotlin.reflect.KClass

/**
 * Marks an expected annotation class that it isn’t required to have actual counterparts in all platforms.
 * This annotation is only applicable to ‘expect’ annotation classes in multi-platform projects and marks that class as
 * ‘optional’. *
 * Optional expected class is allowed to have no corresponding actual class on the platform. Optional annotations can only be used to annotate something, not as types in signatures. If an optional annotation has no corresponding actual class on a platform, the annotation entries where it’s used are simply erased when compiling code on that platform.
 * Note: this annotation is experimental, see [ExperimentalMultiplatform] on how to opt-in for it.
 */
@Target(ANNOTATION_CLASS)
@Retention(AnnotationRetention.BINARY)
@RequireKotlin("1.2.50")
@versionKind = RequireKotlinVersionKind.COMPLIER_VERSION
@public annotation class OptionalExpectation

@Suppress("DEPRECATION")
@Experimental
@OptIn
@Target(CLASS, ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, TYPEALIAS)
@Retention(AnnotationRetention.BINARY)
@RequireKotlin("1.4")
@ExperimentalTypeInference
@public annotation class OverloadResolutionByLambdaReturnType

// Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
// Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

package kotlin

import kotlin.annotation.AnnotationRetention.BINARY
import kotlin.annotation.AnnotationRetention.SOURCE
import kotlin.reflect.KClass

/**
 * Signals that the annotated class is a marker of an API that requires an explicit opt-in.
 * Call sites of any declaration annotated with that marker should opt-in to the API either by using [OptIn], or by being annotated with that marker
themselves, effectively causing further propagation of the opt-in requirement.

* @property message message to be reported on usages of API without an explicit opt-in, or empty string for the default message.

* @property level specifies how usages of API without an explicit opt-in are reported in code.

* @Target(ANNOTATION_CLASS) @Retention(BINARY) @SinceKotlin("1.3") @RequireKotlin("1.3.70
"
Edition, versionKind = RequireKotlinVersionKind.COMPLIER_VERSION) @public annotation class RequiresOptIn(
    val message: String = "",
    val level: Level = Level.ERROR
) {
    /**
     * Severity of the diagnostic that should be reported on usages which did not explicitly opted into
     * the API either by using [OptIn] or by being annotated with the corresponding marker annotation.
     * @property message message to be reported on usages of API without an explicit opt-in, or empty string for the default message.
     */
    public enum class Level {
        /** Specifies that a warning should be reported on incorrect usages of this API. */
        WARNING,

        /** Specifies that an error should be reported on incorrect usages of this API. */
        ERROR,
    }
}

/**
 * Allows to use the API denoted by the given markers in the annotated file, declaration, or expression.
 * If a declaration is annotated with [OptIn], its usages are not required to opt in to that API.
 * This class requires opt-in itself and can only be used with the compiler argument `-Xopt-in=kotlin.RequiresOptIn`.
 */
@Target(
    CLASS, PROPERTY,
    LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER,
    PROPERTY_SETTER, EXPRESSION, FILE,
    TYPEALIAS
) @Retention(SOURCE) @SinceKotlin("1.1") @public annotation class OptIn(
    vararg val markerClass: KClass<out Annotation>
)


this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

*package kotlin.collections

/**
 * Provides a skeletal implementation of the read-only [Set] interface.
 *
 * This class is intended to help implementing read-only sets so it doesn't support concurrent modification tracking.
 *
 * @param E the type of elements contained in the set. The set is covariant in its element type.
 */
@SinceKotlin("1.1.1")
public abstract class AbstractSet<out E> protected constructor() : 
AbstractCollection<E>(), Set<E> { /* Compares this set with other set instance with the unordered structural equality.
 * @param E override fun equals(other: Any?): Boolean { if (other !== this) return false
 * return setEquals(this, other)
 * } */

    /**
     * Returns the hash code value for this set.
     */
    override fun hashCode(): Int = unorderedHashCode(this)

    internal companion object {
        internal fun unorderedHashCode(c: Collection<*>) = 

        internal fun setEquals(c: Set<*>, other: Set<*>) =

    }

}
first element, or throws [NoSuchElementException] if this deque is empty. 
    */
    public fun firstOrNull(): E? = if (isEmpty()) null else internalGet(internalIndex(firstIndex))

    /**
     * Returns the first element, or throws [NoSuchElementException] if this deque is empty. 
    */
    public fun first(): E = if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.") else internalGet(head)

    /**
     * Prepends the specified [element] to this deque. 
    */
    public fun addFirst(element: E) {
        val element = internalGet(head)
        elementData[head] = null
        head = incremented(head)
        size -= 1
        return element
    }

    /**
     * Removes the first element from this deque and returns that removed element, or throws [NoSuchElementException] if this deque is empty. 
    */
    public fun removeFirst(): E {
        if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")
        val element = internalGet(head)
        elementData[head] = null
        head = incremented(head)
        size -= 1
        return element
    }

    /**
     * Removes the last element from this deque and returns that removed element, or throws [NoSuchElementException] if this deque is empty. 
    */
    public fun removeLast(): E {
        if (isEmpty()) throw NoSuchElementException("ArrayDeque is empty.")
        val element = internalGet(last)
        elementData[last] = null
        last = decremented(last)
        size -= 1
        return element
    }

    /**
     * Removes the last element from this deque and returns that removed element, or throws [NoSuchElementException] if this deque is empty. 
    */
    public fun removeLastOrNull(): E? = if (isEmpty()) null else removeLast()
closer to the last element -> shift succeeding elements
val tail = internalIndex(size)
if (internalIndex < tail) { // head > tail
    elementData.copyInto(elementData, 1, 0, tail)
} else { // head < tail
    elementData.copyInto(elementData, 1, 0, tail)
} else if (index == size) {
    return false
} else if (index < (size + 1) shr 1) {
    // closer to the first element -> shift preceding
    if (shiftedHead >= head + elementsSize) {
        elementData.copyInto(elementData, shiftedHead, head, internalIndex)
    } else {
        elementData.copyInto(elementData, shiftedHead, head + shiftToBack, internalIndex)
    }
} else if (shiftedHead >= 0) {
    elementData.copyInto(elementData, shiftedHead, head, internalIndex)
} else { // head < tail, insertion leads to head >= tail
    if (shiftedHead >= elementData.size) {
        val elementsToShift = internalIndex - head
        val shiftToBack = elementData.size - shiftedHead - elementsSize
        if (shiftToBack >= elementsToShift) {
            shiftToBack -= elementsToShift
            if (shiftToBack >= 0) {
                elementData.copyInto(elementData, shiftedHead, head, shiftToBack)
            } else {
                val shiftToFront = tail + elementsSize - elementData.size
                elementData.copyInto(elementData, shiftedHead, head, shiftToBack)
            }
        } else { // head > tail, internalIndex < tail
            elementData.copyInto(elementData, shiftedHead, head, shiftToBack)
        }
        if (elementsSize >= internalIndex) {
            val head = shiftedHead
            copyCollectionElements(negativeMod(internalIndex - elementsSize), elements)
        } else if (internalIndex < tail) {
            val shiftedHead = internalIndex + elementsSize
            if (shiftedHead >= 0) {
                elementData.copyInto(elementData, shiftedHead, head, internalIndex)
            } else { // head < tail, insertion leads to head >= tail
                if (shiftedInternalIndex >= elementData.size) {
                    elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)
                } else if (internalIndex >= head) {
                    elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail)
                } else { // head > tail, internalIndex > head
                    elementData.copyInto(elementData, elementsSize, 0, tail)
                }
            } else if (shiftedInternalIndex >= elementData.size) {
                elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)
            } else {
                val shiftToFront = tail + elementsSize - elementData.size
                elementData.copyInto(elementData, 0, tail - shiftToFront, tail)
                elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail - shiftToFront)
                elementData.copyInto(elementData, 0, elementsSize, internalIndex)
            } else {
                copyCollectionElements(negativeMod(internalIndex - elementsSize), elements)
            }
        } else { // closer to the last element -> shift succeeding elements
            val shiftedInternalIndex = internalIndex + elementsSize
            if (shiftedInternalIndex < tail) {
                val shiftedHead = internalIndex
                if (shiftedHead >= 0) {
                    elementData.copyInto(elementData, shiftedHead, head, internalIndex)
                } else { // head < tail, internalIndex < tail
                    elementData.copyInto(elementData, shiftedHead, head, internalIndex)
                }
            } else if (shiftedInternalIndex >= elementData.size) {
                elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)
            } else if (internalIndex >= head) {
                elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail)
            } else { // head > tail, internalIndex > head
                elementData.copyInto(elementData, elementsSize, 0, tail)
            }
        }
    }
    size += elements.size
}

// closer to the last element -> shift succeeding elements
val shiftedInternalIndex = internalIndex + elementsSize
if (shiftedInternalIndex < tail) {
    val shiftedHead = internalIndex
    if (shiftedHead >= 0) {
        elementData.copyInto(elementData, shiftedHead, head, internalIndex)
    } else { // head < tail, insertion leads to head >= tail
        if (shiftedInternalIndex >= elementData.size) {
            elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)
        } else if (internalIndex >= head) {
            elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail)
        } else { // head > tail, internalIndex > head
            elementData.copyInto(elementData, elementsSize, 0, tail)
        }
    }
} else if (shiftedInternalIndex >= elementData.size) {
    elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)
} else {
    if (shiftedInternalIndex >= elementData.size) {
        elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)
    } else {
        val shiftToFront = tail + elementsSize - elementData.size
        elementData.copyInto(elementData, 0, tail - shiftToFront, tail)
        elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail - shiftToFront)
        elementData.copyInto(elementData, 0, elementsSize, internalIndex)
    }
} else { // closer to the last element -> shift succeeding elements
    val shiftedInternalIndex = internalIndex + elementsSize
    if (shiftedInternalIndex < tail) {
        val shiftedHead = internalIndex
        if (shiftedHead >= 0) {
            elementData.copyInto(elementData, shiftedHead, head, internalIndex)
        } else { // head < tail, internalIndex < tail
            elementData.copyInto(elementData, shiftedHead, head, internalIndex)
        }
    } else if (shiftedInternalIndex >= elementData.size) {
        elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex, tail)
    } else if (internalIndex >= head) {
        elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail)
    } else { // head > tail, internalIndex > head
        elementData.copyInto(elementData, elementsSize, 0, tail)
    }
}
indexOf(element: E): Int { 
    val tail = internalIndex(size)
    if (head < tail) { 
        for (index in tail - 1 downTo head) { 
            if (element == elementData[index]) return index - head
        }
    } else if (head >= tail) { 
        for (index in head until elementData.size) { 
            if (element == elementData[index]) return index - head
        }
    } else if (element == elementData[index]) return index - head
}

lastIndexOf(element: E): Int { 
    val tail = internalIndex(size)
    if (head < tail) { 
        for (index in tail - 1 downTo head) { 
            if (element == elementData[index]) return index - head
        }
    } else if (index == -1) return -1
}

removeAll(elements: Collection<E>): Boolean { 
    if (elementData.isEmpty()) return true
    removeAt(index)
    return true
}

removeAt(index: Int): E { 
    val element = elementData[index]
    elementData[index] = null
    return element
}

remove(element: E): Boolean { 
    val index = indexOf(element)
    if (index == -1) return false
    removeAt(index)
    return true
}

retainAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}

retainAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}

filterInPlace(predicate: (E) -> Boolean): Boolean { 
    if (predicate(element as E)) return true
    modified = false
    for (index in head until tail) { 
        if (element == elementData[index]) return false
    }
    return true
}

retainAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}

removeAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}

retainAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}

filterInPlace(predicate: (E) -> Boolean): Boolean { 
    if (predicate(element as E)) return true
    modified = false
    for (index in head until tail) { 
        if (element == elementData[index]) return false
    }
    return true
}

retainAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}

removeAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}

retainAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}

filterInPlace(predicate: (E) -> Boolean): Boolean { 
    if (predicate(element as E)) return true
    modified = false
    for (index in head until tail) { 
        if (element == elementData[index]) return false
    }
    return true
}

retainAll(elements: Collection<E>): Boolean { 
    if (elementData.containsAll(elements)) return true
    filterInPlace { 
        if (predicate(element as E)) 
            elementData[index++]
        else 
            modified = true
    }
    return true
}
minCapacity < 0) newCapacity = minCapacity 
if (newCapacity - maxArraySize > 0) 
newCapacity = if (minCapacity > maxArraySize) Int.MAX_VALUE else maxArraySize

newCapacity


 // For testing only

 internal fun internalStructure(structure: (head: Int, elements: Array<out Any?>) -> Unit) {

 val tail = internalIndex(size)

 if (isEmpty()) return 

 val elements = arrayOfNulls<Any?>(size)

 if (head < tail) {

 elementData.copyInto(elements, startIndex = head, endIndex = tail)

 structure(head, elements)
 }
 else {

 elementData.copyInto(elements, startIndex = head)

 elementData.copyInto(elements, 0, size - tail, startIndex = 0, endIndex = tail)

 structure(head - elementData.size, elements)
 }

 }
}

 internal fun internalStructure(structure: (head: Int, elements: Array<Any?>) -> Unit) {

 val tail = internalIndex(size)

 if (isEmpty()) {

 structure(head, emptyArray())

 return
 }

 val elements = arrayOfNulls<Any?>(size)

 if (head < tail) {

 elementData.copyInto(elements, startIndex = head, endIndex = tail)

 structure(head, elements)
 }
 else {

 elementData.copyInto(elements, startIndex = head)

 elementData.copyInto(elements, elementData.size - head, startIndex = 0, endIndex = tail)

 structure(head - elementData.size, elements)
 }

 return 
 }

 // For testing only

 internal fun internalStructure(structure: (head: Int, elements: Array<Any?>) -> Unit) {

 val tail = internalIndex(size)

 if (isEmpty()) {

 structure(head,
 emptyArray())

 return
 }

 val elements = arrayOfNulls<Any?>(size)

 if (head < tail) {

 elementData.copyInto(elements, startIndex = head, endIndex = tail)

 structure(head, elements)
 }
 else {

 elementData.copyInto(elements, startIndex = head)

 elementData.copyInto(elements, elementData.size - head, startIndex = 0, endIndex = tail)

 structure(head - elementData.size, elements)
 }

 return 
 }
}
to overflow Int.MAX_VALUE

    return buildString(length) {
        contentDeepToStringInternal(this,
            mutableListOf())
    }

@OptIn(ExperimentalUnsignedTypes::class)
private fun <T> Array<out T>.contentDeepToStringInternal(result: StringBuilder, processed: MutableList<Array<*>>) {
    if (this in
        processed) {
        result.append("[...]\n"
    } else {
        result.append('[')

        for (i
            in indices) {
            if (i != 0) {
                result.append(', '
            )
            val element = this[i]
            when
                (element) {
                null            -> result.append("null")
                is Array<*>     -> element.contentDeepToStringInternal(result, processed)
                is ByteArray    -> result.append(element.contentToString())
                is ShortArray   -> result.append(element.contentToString())
                is IntArray     -> result.append(element.contentToString())
                is LongArray    -> result.append(element.contentToString())
                is FloatArray   -> result.append(element.contentToString())
                is DoubleArray  -> result.append(element.contentToString())
                is CharArray    -> result.append(element.contentToString())
                is BooleanArray -> result.append(element.contentToString())
                is UByteArray   -> result.append(element.contentToString())
                is UShortArray  -> result.append(element.contentToString())
                is UIntArray    -> result.append(element.contentToString())
                is ULongArray   -> result.append(element.contentToString())
                else            -> result.append(element.toString())
            }
        }

        result.append(']')

        processed.removeAt(processed.lastIndex)
    }
}

package kotlin.collections

/**
 * Data class representing a value from a
collection or sequence, along with its index in that
collection or sequence.
 *
 * @property value the underlying
 * value.
 * @property index the index of the value in the
collection or sequence.
 */
public data class IndexedValue<out T>(public val index: Int, public val value: T)

package kotlin.collections

import kotlin.reflect.KProperty
import kotlin.internal.Exact

/**
 * Returns the value of
 * the property for the given object from this read-only map.
 * @param thisRef the object for which the value is
 * requested (not used).
 * @param property the metadata for the property, used to get
 * the name of property and store the value associated with that name in the map.
 * @return the property value.
 *
 * @throws
 * NoSuchElementException when the map doesn't contain value
 * for the property name and doesn't provide an implicit
default (see [withDefault]).
 */
@kotlin.internal.InlineOnly
public inline operator fun <V, V1 : V> Map<in String, @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =
    @Suppress("UNCHECKED_CAST") (getOrImplicitDefault(property.name) as V1)

/**
 * Returns the value of the property for the given object from this mutable map.
 * @param thisRef the object for which the value is
 * requested (not used).
 * @param property the metadata for the property, used to get
 * the name of property and store the value associated with that name in the map.
 * @return the property value.
 *
 * @throws
 * NoSuchElementException when the map doesn't contain value
 * for the property name and doesn't provide an implicit
default (see [withDefault]).
 */
@kotlin.jvm.JvmName("getVar")
@kotlin.internal.InlineOnly
public inline operator fun <V, V1 : V> MutableMap<in String, out @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =
    @Suppress("UNCHECKED_CAST") (getOrImplicitDefault(property.name) as V1)

/**
 * Stores the value of the property for the given object in this mutable map.
 * @param thisRef the object for which the value is
 * requested (not used).
 * @param property the metadata for the property, used to get
 * the value corresponding to this name in the map.
 * @param value the value to store.
 */
@kotlin.internal.InlineOnly
public inline operator fun <V> MutableMap<in String, in V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {
    this.put(property.name, value)
}
default no implicit value is provided for maps and a [NoSuchElementException] is thrown.
* @n @n To create a map with implicit default value use [withDefault] method.
* @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n * @n *
not present in the collection.

*\n */
@kotlin.internal.InlineOnly
public inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.remove(element: T): Boolean =
    @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).remove(element)

*\n */
@kotlin.internal.InlineOnly
public inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.removeAll(elements: Collection<T>): Boolean =
    @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).removeAll(elements)

*\n */
@kotlin.internal.InlineOnly
public inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.retainAll(elements: Collection<T>): Boolean =
    @Suppress("UNCHECKED_CAST") (this as MutableCollection<T>).retainAll(elements)

*\n */
@kotlin.internal.InlineOnly
public inline operator fun <T> MutableCollection<in T>.plusAssign(element: T) {
    this.add(element)
}

*\n */
@kotlin.internal.InlineOnly
public inline operator fun <T> MutableCollection<in T>.plusAssign(elements: Iterable<T>) {
    this.addAll(elements)
}

*\n */
@kotlin.internal.InlineOnly
public inline operator fun <T> MutableCollection<in T>.plusAssign(elements: Array<T>) {
    this.addAll(elements)
}

*\n */
@kotlin.internal.InlineOnly
public inline operator fun <T> MutableCollection<in T>.plusAssign(elements: Sequence<T>) {
    this.addAll(elements)
}

*\n */
@kotlin.internal.InlineOnly
public inline operator fun <T> MutableCollection<in T>.minusAssign(element: T) {
    this.remove(element)
}

*\n */
@kotlin.internal.InlineOnly
public inline operator fun <T> MutableCollection<in T>.minusAssign(elements: Iterable<T>) {
    this.removeAll(elements)
}

*\n */
@kotlin.internal.InlineOnly
public inline operator fun <T> MutableCollection<in T>.minusAssign(elements: Array<T>) {
    this.removeAll(elements)
}

*\n */
@kotlin.internal.InlineOnly
public inline operator fun <T> MutableCollection<in T>.minusAssign(elements: Sequence<T>) {
    this.removeAll(elements)
}

*\n */
public fun <T> MutableCollection<in T>.addAll(elements: Iterable<T>): Boolean {
    when (elements) {
        is Collection -> return addAll(elements)
        else -> {
            var result: Boolean = false
            for (item in elements)
                if (add(item)) result = true
            return result
        }
    }
}

*\n */
public fun <T> MutableCollection<in T>.addAll(elements: Sequence<T>): Boolean {
    var result: Boolean = false
    for (item in elements) {
        if (add(item)) result = true
    }
    return result
}

*\n */
public fun <T> MutableCollection<in T>.addAll(elements: Array<out T>): Boolean {
    return addAll(elements.asList())
}

*\n */
public fun <T> MutableCollection<in T>.removeAll(elements: Iterable<T>): Boolean {
    return removeAll(elements)
}

*\n */
public fun <T> MutableCollection<in T>.removeAll(elements: Array<out T>): Boolean {
    return elements.isNotEmpty() &&

removeAll(elements.toHashSet())}

/**
 * Retains only elements of this [MutableCollection] that are contained in the given [elements] collection.
 */
public fun <T> MutableCollection<T>.retainAll(elements: Iterable<T>): Boolean {
    return retainAll(elements.convertToSetForSetOperationWith(this))}

/**
 * Retains only elements of this [MutableCollection] that are contained in the given [elements] array.
 */
public fun <T> MutableCollection<T>.retainAll(elements: Array<out T>): Boolean {
    if (elements.isNotEmpty())
        return retainAll(elements.toHashSet())
    else
        return retainNothing()
}

/**
 * Retains only elements of this [MutableCollection] that are contained in the given [elements] sequence.
 */
public fun <T> MutableCollection<T>.retainAll(elements: Sequence<T>): Boolean {
    if (elements.isNotEmpty())
        return retainAll(elements.toHashSet())
    else
        return retainNothing()
}

private fun <T> MutableCollection<T>.retainNothing(): Boolean {
    val result = isNotEmpty()
    clear()
    return result
}

/**
 * Removes all elements from this [MutableIterable] that match the given [predicate].
 * @return `true` if any element was removed from this collection, or `false` when no elements were removed.
 * @throws [NoSuchElementException] if the collection is empty.
 */
public fun <T> MutableIterable<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, true)

/**
 * Retains only elements of this [MutableIterable] that match the given [predicate].
 * @return `true` if any element was removed from this collection, or `false` when all elements were retained.
 */
public fun <T> MutableIterable<T>.retainAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, false)

private fun <T> MutableIterable<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {
    var result = false
    with(iterator()) {
        while (hasNext())
            if (predicate(next()) == predicateResultToRemove) {
                remove()
                result = true
            }
    }
    return result
}

/**
 * Removes the element at the specified [index] from this list.
 * In Kotlin one should use the [MutableList.removeAt] function instead.
 * @throws [NoSuchElementException] if the list is empty.
 */
@Deprecated("Use removeAt(index) instead.", ReplaceWith("removeAt(index)"), level = DeprecationLevel.ERROR)
public inline fun <T> MutableList<T>.removeAt(index: Int): T = removeAt(index)

/**
 * Removes the first element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> MutableList<T>.removeFirst(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else removeAt(0)

/**
 * Removes the first element from this mutable list and returns that removed element, or returns `null` if this list is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> MutableList<T>.removeFirstOrNull(): T? = if (isEmpty()) null else removeAt(0)

/**
 * Removes the last element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> MutableList<T>.removeLast(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else removeAt(lastIndex)

/**
 * Removes the last element from this mutable list and returns that removed element, or returns `null` if this list is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> MutableList<T>.removeLastOrNull(): T? = if (isEmpty()) null else removeAt(lastIndex)
predicateResultToRemove
) continue

if (writeIndex != readIndex)
    this[writeIndex] = element

writeIndex++

if (writeIndex < size) {
    for (removeIndex in lastIndex downTo
        override fun get(index: Int): T =
            delegate[reverseElementIndex(index)]
    }
    if (writeIndex < size) {
        for (removeIndex in lastIndex downTo
            override fun clear() =
        }
    }
    else {
        return false
    }
}

return true

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CollectionsKt")

package kotlin.collections

private open class ReversedListReadOnly<out T>(private val delegate: List<T>) : AbstractList<T>() {
    override val size: Int get() = delegate.size
    override fun get(index: Int): T = delegate[reverseElementIndex(index)]
}

private class ReversedList<T>(private val delegate: MutableList<T>) : AbstractMutableList<T>() {
    override val size: Int get() = delegate.size
    override fun get(index: Int): T = delegate[reverseElementIndex(index)]

    override fun clear() = delegate.clear()
    override fun removeAt(index: Int): T = delegate.removeAt(reverseElementIndex(index))
    override fun set(index: Int, element: T): T =
        delegate.set(reverseElementIndex(index), element)
    override fun add(index: Int, element: T) {
        delegate.add(reversePositionIndex(index), element)
    }
}

private fun List<*>.reverseElementIndex(index: Int) =
    if (index in 0..lastIndex) lastIndex - index else throw IndexOutOfBoundsException("Element index $index must be in range [0..lastIndex].")

private fun List<*>.reversePositionIndex(index: Int) =
    if (index in 0..size) size - index else throw IndexOutOfBoundsException("Position index $index must be in range [0..size].")

/**
 * Returns a reversed read-only view of the original List.
 * All changes made in the original list will be reflected in the reversed one.
 * @sample samples.collections.ReversedViews.asReversed
 */
public fun <T> List<T>.asReversed(): List<T> = ReversedListReadOnly(this)

/**
 * Returns a reversed mutable view of the original mutable List.
 * All changes made in the original list will be reflected in the reversed
 * one and vice versa.
 * @sample samples.collections.ReversedViews.asReversedMutable
 */
@kotlin.jvm.JvmName("asReversedMutable")
public fun <T> MutableList<T>.asReversed(): MutableList<T> = ReversedList(this)

/**
 * Builds a [Sequence] lazily yielding values one by one.
 */
@SinceKotlin("1.3")
public fun <T> sequence(@BuilderInference block: suspend SequenceScope<T>.() -> Unit): Sequence<T> = Sequence {
    iterator(block) }

@SinceKotlin("1.3")
@Deprecated("Use 'sequence { }' function instead.",
    ReplaceWith("sequence(builderAction)"), level = DeprecationLevel.ERROR)
public inline fun <T> buildSequence(@BuilderInference noinline builderAction: suspend SequenceScope<T>.() -> Unit): Sequence<T> =
    iterator(builderAction)

/**
 * The scope for yielding values of a [Sequence] or an [Iterator],
 * provides [yield] and [yieldAll] suspension functions.
 */
Sequences.buildSequenceYieldAll
samples.collections.Sequences.Building.buildFibonacciSequence

public abstract class SequenceScope<in T> internal constructor() {
    /**
     * Yields a value to the [Iterator] being built and suspends
     * until the next value is requested.
     */
    public abstract suspend fun yield(value: T)

    /**
     * Yields all values from the `iterator` to the [Iterator] being built
     * and suspends until all these values are iterated and the next one is requested.
     */
    public abstract suspend fun yieldAll(iterator: Iterator<T>)

    /**
     * Yields a collections of values to the [Iterator] being built
     * and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(elements: Iterable<T>)

    /**
     * Yields potentially infinite sequence of values to the [Iterator]
     * being built * and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(sequence: Sequence<T>)

    @Deprecated("Use SequenceScope class instead.",
        ReplaceWith("SequenceScope<T>"), level = DeprecationLevel.ERROR)
    public typealias SequenceBuilder<T> = SequenceScope<T>

    private typealias State = Int

    private const val State_NotReady: State = 0
    private const val State_ManyNotReady: State = 1
    private const val State_ManyReady: State = 2
    private const val State_Ready: State = 3
    private const val State_Done: State = 4
    private const val State_Failed: State = 5

    private class SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {
        private var state = State_NotReady
        private var nextValue: T? = null
        private var nextIterator: Iterator<T>? = null
        var nextStep: Continuation<Unit>? = null

        override fun hasNext(): Boolean {
            while (true) {
                when (state) {
                    State_NotReady -> {}
                    State_ManyNotReady ->
                        if (nextIterator!!.hasNext()) {
                            state = State_ManyReady
                            return true
                        }
                    else {
                        nextIterator = null
                        state = State_Done
                        return false
                    }
                    State_Done -> return false
                    State_Ready, State_ManyReady -> return true
                    else -> throw exceptionalState()
                }
            }
        }

        override fun next(): T {
            when (state) {
                State_NotReady, State_ManyNotReady -> return nextNotReady()
                State_ManyReady -> {
                    state = State_ManyNotReady
                    return nextIterator!!.next()
                }
                State_Ready -> {
                    state = State_ManyNotReady
                    return nextNotReady()
                }
                else -> throw exceptionalState()
            }
        }

        private fun nextNotReady(): T {
            if (!hasNext()) throw NoSuchElementException() else return next()
        }

        private fun exceptionalState(): Throwable = when (state) {
            State_Done -> NoSuchElementException()
            State_Failed -> IllegalStateException("Iterator has failed.")
            else -> IllegalStateException("Unexpected state of the iterator: $state")
        }

        override suspend fun yield(value: T) {
            state = State_Ready
            return suspendCoroutineUninterceptedOrReturn { c ->
                nextStep = c
                // Completion continuation implementation
                // override fun resumeWith(result: Result<Unit>)
                // result.getOrThrow() // just rethrow exception if it is there
                // state = State_Done
                // override val context:
                // CompletionContext
                // get() = EmptyCoroutineContext()
                // }*/
            }
        }

        override suspend fun yieldAll(iterator: Iterator<T>) {
            if (!iterator.hasNext()) return
            nextIterator = iterator
            return suspendCoroutineUninterceptedOrReturn { c ->
                nextStep = c
                // Completion continuation implementation
                // override fun resumeWith(result: Result<Unit>)
                // result.getOrThrow() // just rethrow exception if it is there
                // state = State_Done
                // override val context:
                // CompletionContext
                // get() = EmptyCoroutineContext()
                // }*/
            }
        }

        // Completion continuation implementation
        override fun resumeWith(result: Result<Unit>)
        // result.getOrThrow() // just rethrow exception if it is there
        // state = State_Done
        // override val context:
        // CompletionContext
        // get() = EmptyCoroutineContext()
        // }
        //
"Both size $size and step $step must be greater than zero.\""
first elements from the buffer or fails with [IllegalArgumentException] if not enough elements in the buffer to remove

*/
    fun removeFirst(n: Int) {
        require(n >= 0) { "n shouldn't be negative but it is $n" }

        require(n <= size) { "n shouldn't be greater than the buffer size: n = $n, size = $size" }

        if (n > 0) {
            val start = startIndex
            val end = start.forward(n)

            if (start > end) {
                buffer.fill(null, start, end)
            } else {
                buffer.fill(null, 0, end)
            }

            startIndex = end
            size -= n
        }
    }

    @Suppress("NOTHING_TO_INLINE")
    private inline fun Int.forward(n: Int): Int = (this + n) % capacity
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

// UByteArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(array: UByteArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(array: UByteArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// UShortArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(array: UShortArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(array: UShortArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// UIntArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(array: UIntArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(array: UIntArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// ULongArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(array: ULongArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(array: ULongArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// Interfaces
=============================================================================
/**
 * Sorts the given array using qsort algorithm.
 */
@ExperimentalUnsignedTypes
internal fun sortArray(array: UByteArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)

@ExperimentalUnsignedTypes
internal fun sortArray(array: UShortArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)

@ExperimentalUnsignedTypes
internal fun sortArray(array: UIntArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)

@ExperimentalUnsignedTypes
internal fun sortArray(array: ULongArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)
fun quickSort(array, fromIndex, toIndex - 1) { /*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
 * Language contributors. * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file. */

package kotlin.contracts

import kotlin.internal.ContractsDsl
import kotlin.internal.InlineOnly

/**
 * This marker distinguishes the experimental contract declaration API and is
 * used to opt-in for that feature
 * when declaring contracts of user functions. * Any usage of a declaration
 * annotated with `@ExperimentalContracts` must be accepted either by
 * annotating that usage with the [OptIn]
 * annotation, e.g.: `@OptIn(ExperimentalContracts::class)` or by using the compiler argument `-Xopt-
 * in=kotlin.contracts.ExperimentalContracts`.
 */
@Suppress("DEPRECATION")
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.3")
@Experimental
@RequiresOptIn
@MustBeDocumented
public annotation class ExperimentalContracts

/**
 * Provides a scope, where the functions of the contract DSL, such as [returns], [callsInPlace], etc., can be used to
 * describe the contract of a function. * This type is used as a receiver type of the lambda function passed to the
 * [contract] function.
 */
@ContractsDsl
@ExperimentalContracts
@SinceKotlin("1.3")
public interface ContractBuilder {
    /**
     * Describes a situation when a function returns normally, without any exceptions thrown.
     * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
     */
    @sample samples.contracts.returnsContract
    @ContractsDsl public fun returns(): Returns

    /**
     * Describes a situation when a function returns normally with the specified return value.
     * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
     */
    @sample samples.contracts.returnsTrueContract
    @sample samples.contracts.returnsFalseContract
    @sample samples.contracts.returnsNullContract
    @ContractsDsl public fun returns(value: Any?): Returns

    /**
     * Describes a situation when a function returns normally with any value that is not `null`.
     * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.
     */
    @sample samples.contracts.returnsNotNullContract
    @ContractsDsl public fun returnsNotNull(): ReturnsNotNull

    /**
     * Specifies that the function parameter [lambda] is invoked in place.
     * This contract specifies that:
     * 1. the function [lambda] can only be invoked during the call of the owner function;
     * and it won’t be invoked after that owner function call is completed;
     * 2. (optionally) the function [lambda] is invoked the amount of times specified by the [kind] parameter.
     * see the [InvocationKind] enum for possible values.
     * A function declaring the `callsInPlace` effect must be _inline_.
     */
    @ContractsDsl public fun callsInPlace(lambda: Function<R>, kind: InvocationKind = InvocationKind.UNKNOWN): CallsInPlace

    /**
     * Specifies how many times a function invokes its function parameter in place.
     * See [ContractBuilder.callsInPlace] for the details of the call-in-place function contract.
     */
    @ContractsDsl public enum class InvocationKind {
    AT_MOST_ONCE, AT_LEAST_ONCE, EXACTLY_ONCE, UNKNOWN
    }

    /**
     * Specifies the contract of a function.
     * The contract description must be at the beginning of a function and have at least one effect.
     * Only the top-level functions can have a contract for now.
     */
    @param builder the lambda where
the contract of a function is described with the help of the [ContractBuilder] members.

```kotlin
@ContractsDsl
@ExperimentalContracts
@InlineOnly
@SinceKotlin("1.3")
@Suppress("UNUSED_PARAMETER")
public inline fun contract(builder: ContractBuilder.() -> Unit) {
}
```

* The coroutines framework uses [ContinuationInterceptor.Key] to retrieve the
  interceptor and \(n\) intercepts all coroutine continuations with [interceptContinuation] invocations.

```kotlin
@SinceKotlin("1.3")
public interface ContinuationInterceptor : CoroutineContext.Element {
    companion object Key : CoroutineContext.Key<ContinuationInterceptor>

    fun <T> interceptContinuation(continuation: Continuation<T>): Continuation<T>

    fun releaseInterceptedContinuation(continuation: Continuation<*>)

        return if (key is AbstractCoroutineContextKey<*, *>) getPolymorphicElement(key) else
            if (ContinuationInterceptor === key) this as E else null
    }

    override fun minusKey(key: CoroutineContext.Key<*>): CoroutineContext {
        return if (ContinuationInterceptor === key) EmptyCoroutineContext else this
    }
}
```

- Every element in this set has a unique [Key].
- Persistent context for the coroutine. It is an indexed set of [Element] instances.
- An indexed set is a mix between a set and a map.
public operator fun <E : Element> get(key: Key<E>): E?/* * * */ /* Accumulates entries of this context starting with [initial] value and applying [operation] from left to right to current accumulator value and each element of this context.*/ /* */ public fun <R> fold(initial: R, operation: (R, Element) -> R): R/* * */ /* Returns a context containing elements from this context and elements from other context.*/ /* */ Elements from this context with the same key as in the other one are dropped. */ /* */ public operator fun plus(context: CoroutineContext): CoroutineContext = if (context === EmptyCoroutineContext) this else { /* make sure interceptor is always last in the context (and thus is fast to get when present) */ val interceptor = removed[ContinuationInterceptor] if (interceptor == null) CombinedContext(removed, element) else { val left = removed.minusKey(ContinuationInterceptor) if (left !== EmptyCoroutineContext) CombinedContext(left, element, interceptor) else { CombinedContext(CoroutineContext(), element, interceptor) /* Returns a context containing elements from this context, but without an element with the specified key. */ /* */ public fun minusKey(key: Key<*>): CoroutineContext = /* */ /* */ public interface Element : CoroutineContext { /* */ public override operator fun <E : Element> get(key: Key<E>): E? = if (this.key == key) this as E else null /* */ public override fun <R> fold(initial: R, operation: (R, Element) -> R): R = operation(initial, this) /* */ public override fun minusKey(key: Key<*>): CoroutineContext = if (this.key == key) EmptyCoroutineContext else this /* */ }

/* */ @SinceKotlin("1.3") public abstract class AbstractCoroutineContextElement(public override val key: Key<*>) : Element

/* */ @SinceKotlin("1.3") @ExperimentalStdlibApi public abstract class AbstractCoroutineContextKey<B : Element, E : B>(baseKey: Key<B>, private val safeCast: (element: Element) -> E?) /* Base class for [CoroutineContext.Element] implementations. */ /* */ package kotlin.coroutines import kotlin.coroutines.CoroutineContext.Element

import kotlin.coroutines.CoroutineContext.Key

/* */ package kotlin.coroutines

import kotlin.coroutines.CoroutineContext.Element
import kotlin.coroutines.CoroutineContext.Key

/* */ package kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.coroutines

import kotlin.corpu...
internal fun isSubKey(key: Key<*>) = this !== key && key.key === this || super.key === key

/**
 * Returns the current element if it is associated with the given [key] in a polymorphic manner
 * or `null` otherwise.
 * This method returns non-null value if either [Element.key] is equal to the given [key] or if
 * the [key] is associated with [Element.key] via [AbstractCoroutineContextKey].
 */
public fun <E : Element, K> Element.getPolymorphicElement(key: Key<K>): E? {
    if (key is AbstractCoroutineContextKey) {
        return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null
    }
    return if (this.key === key) this as E else null
}

/**
 * Returns empty coroutine context if the element is associated with the given [key] in a polymorphic manner
 * or `null` otherwise.
 * This method returns empty context if either [Element.key] is equal to the given [key] or if
 * the [key] is associated with [Element.key] via [AbstractCoroutineContextKey].
 */
public fun Element.minusPolymorphicKey(key: Key<*>): CoroutineContext {
    if (key is AbstractCoroutineContextKey) {
        return if (key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this
    }
    return if (this.key === key) EmptyCoroutineContext else this
}

/**
 * An empty coroutine context.
 */
public object EmptyCoroutineContext : CoroutineContext, Serializable {
    private const val serialVersionUID: Long = 0
    private fun readResolve(): Any = EmptyCoroutineContext

    public override fun <E : Element> get(key: Key<E>): E? = null
    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R = initial
    public override fun plus(context: CoroutineContext): CoroutineContext = context
    public override fun minusKey(key: Key<*>): CoroutineContext {
        return when {
            context === EmptyCoroutineContext -> this
            context === EmptyCoroutineContext -> super.minusKey(key)
            else -> CombinedContext(context, this)
        }
    }
    public override fun hashCode(): Int = 0
    public override fun toString(): String = "EmptyCoroutineContext"
}

//--------------------- internal impl ---------------------

// this class is not
// exposed, but is hidden inside implementations
// this is a left-biased list, so that `plus` works
// naturally
private class CombinedContext(
    private val left: CoroutineContext,
    private val element: Element
) : CoroutineContext, Serializable {

    override fun <E : Element> get(key: Key<E>): E? {
        var cur = this
        while (true) {
            cur = cur.element? [key] ?: return null
            val next = cur.left
            if (next is CombinedContext) {
                cur = next
            } else {
                return next[key]
            }
        }
    }

    override fun <R> fold(initial: R, operation: (R, Element) -> R): R = operation(left.fold(initial, operation), element)

    override fun minusKey(key: Key<*>): CoroutineContext {
        return when {
            left === this -> this
            left === EmptyCoroutineContext -> super.minusKey(key)
            else -> CombinedContext(left, element)
        }
    }

    override fun equals(other: Any?): Boolean = this === other || other is CombinedContext && other.size == size && other.containsAll(this)

    override fun hashCode(): Int = left.hashCode() + element.hashCode()

    override fun toString(): String = "[" + fold { acc, element -> "$acc, $element" } + "]

    private fun writeReplace(): Any { return Serializable(elements.fold(EmptyCoroutineContext, CoroutineContext::plus))

    private fun readResolve(): Any = elements.fold(EmptyCoroutineContext, CoroutineContext::plus)
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmName("IntrinsicsKt")
@file:kotlin.jvm.JvmMultifileClass

package kotlin.coroutines.intrinsics

import kotlin.contracts.*
import kotlin.coroutines.

package kotlin.internal.InlineOnly

suspend CoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }
    throw NotImplementedError("Implementation of suspendCoroutineUninterceptedOrReturn is intrinsic")
}

@SinceKotlin("1.3")
@PublishedApi // This class is Published API via serialized representation of SafeContinuation, don't rename/move\ninternal enum class CoroutineSingletons {
    COROUTINE_SUSPENDED, UNDECIDED, RESUMED }\n
package kotlin.experimental

/** Performs a bitwise AND operation between the two values. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.and(other: Byte): Byte = (this.toInt() and other.toInt()).toByte()

/** Performs a bitwise OR operation between the two values. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.or(other: Byte): Byte = (this.toInt() or other.toInt()).toByte()

/** Performs a bitwise XOR operation between the two values. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline infix fun Byte.xor(other: Byte): Byte = (this.toInt() xor other.toInt()).toByte()

/** Inverts the bits in this value. */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun Byte.inv(): Byte = (this.toInt().inv()).toByte()
*\@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.or(other: Short): Short = (this.toInt() or other.toInt()).toShort()

*\@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.xor(other: Short): Short = (this.toInt() xor other.toInt()).toShort()

*\@SinceKotlin("1.1")\n@kotlin.internal.InlineOnly\npublic inline fun Short.inv(): Short = (this.toInt().inv()).toShort()


"/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.experimental

/**
 * The experimental marker for type inference augmenting annotations.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.internal

/**
 * Specifies that the corresponding type should be ignored during type inference.
 */
@Target(AnnotationTarget.TYPE)
@Retention(AnnotationRetention.BINARY)
internal annotation class NoInfer

/**
 * Specifies that the constraint built for the type during type inference should be an equality one.
 */
@Target(AnnotationTarget.TYPE)
@Retention(AnnotationRetention.BINARY)
internal annotation class Exact

/**
 * Specifies that a corresponding member has the lowest priority in overload resolution.
 */
@Target(AnnotationTarget.FUNCTION,
  AnnotationTarget.PROPERTY)
@Retention(AnnotationRetention.BINARY)
internal annotation class LowPriorityInOverloadResolution

/**
 * Specifies that this function should not be called directly without inlining.
 */
@Target(AnnotationTarget.FUNCTION,
  AnnotationTarget.PROPERTY_GETTER,
  AnnotationTarget.PROPERTY_SETTER)
@Retention(AnnotationRetention.BINARY)
internal annotation class InlineOnly

/**
 * Specifies that the declaration can have dynamic receiver type.
 */
@Target(AnnotationTarget.FUNCTION,
  AnnotationTarget.PROPERTY_GETTER,
  AnnotationTarget.PROPERTY_SETTER)
@Retention(AnnotationRetention.BINARY)
internal annotation class DynamicExtension

/**
 * Specifies that an argument is a `lateinit` property, the backing field of which is accessible at the point where the corresponding argument is passed.
 */
@Target(AnnotationTarget.VALUE_PARAMETER)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
internal annotation class AccessibleLateinitPropertyLiteral

/**
 * Specifies that this declaration is only completely supported since the specified version.
 */
@Target(AnnotationTarget.FUNCTION)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
internal annotation class AccessibleLateinitProperty

/**
 * The value of this parameter should be mentioned in input types (argument types, receiver type or expected type).
 */
@Target(AnnotationTarget.VALUE_PARAMETER)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
internal annotation class OnlyInputTypes

/**
 * Specifies that this function should not be called directly without inlining.
 */
@Target(AnnotationTarget.FUNCTION,
  AnnotationTarget.PROPERTY_GETTER,
  AnnotationTarget.PROPERTY_SETTER)
@Retention(AnnotationRetention.BINARY)
internal annotation class InlineOnly

/**
 * Specifies that this declaration can have dynamic receiver type.
 */
@Target(AnnotationTarget.FUNCTION,
  AnnotationTarget.PROPERTY_GETTER,
  AnnotationTarget.PROPERTY_SETTER)
@Retention(AnnotationRetention.BINARY)
internal annotation class DynamicExtension

/**
 * The value of this parameter should be a property reference expression (`this::foo`), referencing a `lateinit` property.
 */
@Target(AnnotationTarget.VALUE_PARAMETER)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
internal annotation class AccessibleLateinitProperty

/**
 * Specifies that this declaration is only completely supported since the specified version.
 */
@Target(AnnotationTarget.FUNCTION)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
internal annotation class AccessibleLateinitProperty

/**
 * The diagnostic message can be specified with [message], or via [errorCode] (takes less space, but might not be immediately clear to the user). The diagnostic severity can be specified with [level]: WARNING/ERROR mean that either a warning or an error is going to be reported,
 */
@Target(AnnotationTarget.FUNCTION)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
internal annotation class AccessibleLateinitProperty

/**
 * The diagnostic message can be specified with [message], or via [errorCode] (takes less space, but might not be immediately clear to the user). The diagnostic severity can be specified with [level]: WARNING/ERROR mean that either a warning or an error is going to be reported,
HIDDEN means that the declaration is going to be removed from resolution completely.

* * [versionKind]

specifies which version should be compared with the [version] value, when compiling the usage of the annotated declaration.

* Note that prior to 1.2, only [RequireKotlinVersionKind.LANGUAGE_VERSION] was supported, so the Kotlin compiler before 1.2 is going to treat any [RequireKotlin] as if it requires the language version.

Since 1.2, the Kotlin compiler supports

* [RequireKotlinVersionKind.LANGUAGE_VERSION],

[RequireKotlinVersionKind.COMPILER_VERSION] and [RequireKotlinVersionKind.API_VERSION].

* If the actual value of [versionKind] is something different (e.g. a new version kind, added in future versions of Kotlin),

* Kotlin 1.2 is going to ignore this [RequireKotlin] altogether, where as Kotlin before 1.2 is going to treat this as a requirement

* on the language version.

* This annotation is erased at compile time; its arguments are stored in a more compact form in the Kotlin metadata.

*/
@Target(AnnotationTarget.CLASS,
        AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.CONSTRUCTOR,
        AnnotationTarget.TYPEALIAS)
@Retention(AnnotationRetention.SOURCE)
@Repeatable
@SinceKotlin("1.2")
internal annotation class RequireKotlin(
    val version: String,
    val message: String = "",
    val level: DeprecationLevel = DeprecationLevel.ERROR,
    val versionKind: RequireKotlinVersionKind = RequireKotlinVersionKind.LANGUAGE_VERSION,
    val errorCode: Int = -1
)

/**
 * The kind of the version that is required by [RequireKotlin].
 */
@SinceKotlin("1.2")
internal enum class RequireKotlinVersionKind {
    LANGUAGE_VERSION,
    COMPILER_VERSION,
    API_VERSION,
}

/**
 * Specifies that this declaration is a part of special DSL, used for constructing
 * function's contract.
 */
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
internal annotation class ContractsDsl
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.*/

package kotlin.properties

import kotlin.reflect.KProperty

/**
 * Standard property delegates.
 */
public object Delegates {
    /**
     * Returns a property delegate for a read/write
     * property with a non-`null` value that is initialized not during
     * object construction time but at a later time.
     *
     * @sample samples.properties.Delegates.notNullDelegate
     */
    public fun <T : Any> notNull():
        ReadWriteProperty<Any?, T> = NotNullVar()

    /**
     * Returns a property delegate for a read/write property
     * that calls a specified callback function when changed.
     * @param initialValue the initial value of the property.
     * @param onChange the callback which is called before a change to the property value is attempted.
     *  The value of the property hasn't been changed yet, when this callback is invoked.
     * @sample samples.properties.Delegates.observableDelegate
     */
    public inline fun <T> observable(initialValue: T, crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Boolean):
        ReadWriteProperty<Any?, T> = ObservableVar(initialValue, onChange)

    /**
     * Returns a property delegate for a read/write property that calls a specified callback function when changed.
     * @param initialValue the initial value of the property.
     * @param onChange the callback which is called after the change of the property is made.
     *  The value of the property has been changed, when this callback is invoked.
     * @param onReturn the callback which is called when the property value is not set.
     * @sample samples.properties.Delegates.vetoableDelegate
     */
    public inline fun <T> vetoable(initialValue: T, crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Boolean, crossinline onReturn: (property: KProperty<*>, oldValue: T, newValue: T) -> Boolean):
        ReadWriteProperty<Any?, T> = VetoableVar(initialValue, onChange, onReturn)

    private class NotNullVar<T : Any>() : ReadWriteProperty<Any?, T> {
        private var value: T? = null

        public override fun getValue(thisRef: Any?, property: KProperty<*>): T {
            return value
        }

        public override fun setValue(thisRef: Any?, property: KProperty<*>?, value: T?) {
            this.value = value
        }
    }

    private class ObservableVar<T : Any>() : ReadWriteProperty<Any?, T> {
        private var value: T? = null
        private val changes by observedChanges

        public override fun getValue(thisRef: Any?, property: KProperty<*>): T {
            return value
        }

        public override fun setValue(thisRef: Any?, property: KProperty<*>?, value: T?) {
            this.value = value
            changes.add(Change(changes, property, value!!))
        }
    }

    private class VetoableVar<T : Any>() : ReadWriteProperty<Any?, T> {
        private var value: T? = null
        private val changes by observedChanges
        private val veto by volatile observedVeto

        public override fun getValue(thisRef: Any?, property: KProperty<*>): T {
            return value
        }

        public override fun setValue(thisRef: Any?, property: KProperty<*>?, value: T?) {
            if (veto) return
            this.value = value
            changes.add(Change(changes, property, value!!))
        }
    }
}
override fun setValue(thisRef: Any?, property: KProperty<*>, value: T) {
    this.value = value
}

public override operator fun getValue(thisRef: T, property: KProperty<*>): V

public override operator fun setValue(thisRef: T, property: KProperty<*>, value: V) {
    // Implementation...
}

public interface ReadOnlyProperty<in T, out V> {
    public override fun getValue(thisRef: T, property: KProperty<*>): V
}

public interface ReadWriteProperty<in T, V> : ReadOnlyProperty<T, V> {
    public override fun getValue(thisRef: T, property: KProperty<*>): V

    public override fun setValue(thisRef: T, property: KProperty<*>, value: V)
}

public interface PropertyDelegateProvider<in T, out D> {
    public override fun provideDelegate(thisRef: T, property: KProperty<*>): D
}

abstract class ObservableProperty<V>(initialValue: V) : ReadWriteProperty<Any?, V> {
    private var value = initialValue

    protected open fun beforeChange(property: KProperty<*>, oldValue: V, newValue: V): Boolean = true

    protected open fun afterChange(property: KProperty<*>, oldValue: V, newValue: V): Unit {}

    public override fun getValue(thisRef: Any?, property: KProperty<*>): V
    
    public override fun setValue(thisRef: Any?, property: KProperty<*>, value: V)
    {
        // Implementation...
    }
}
An extension operator that allows delegating a read-only property of type [V] to a property reference to a property of type [V] or its subtype. The reference is without a receiver, i.e. it either references a top-level property or has the receiver bound to it. The reference to a property reference to a mutable property of the same type [V] * @receiver A property reference to a mutable property of type [V] * The reference is without a receiver, i.e. it either references a top-level property or has the receiver bound to it. The reference has an unbound receiver of type [T].

```kotlin
/**
 * An abstract class that is implemented by random number generator algorithms.
 * The companion object `Random.Default` is the default instance of `Random`. To get a seeded instance of random number generator use `Random(function: (Random, bitCount: Int, samples: List<Int>) -> Random)`.

```
Generates an `Int` random value uniformly distributed between the specified `[from]` (inclusive) and `[until]` (exclusive) bounds.

*throws IllegalArgumentException if `[from]` is greater than or equal to `[until]`.*

```kotlin
public open fun nextInt(from: Int, until: Int): Int {
    checkRangeBounds(from, until)
    val n = until - from
    if (n > 0 || n == Int.MIN_VALUE) {
        val bitCount = fastLog2(n)
        nextBits(bitCount)
    } else {
        var v: Int
        do {
            val rnd = nextInt().ushr(1)
            v = bits % n
        } while (bits - v + (n - 1) < 0)
        val v = bits % n
        return from + v
    }
}
```

### `nextDouble()` Method

Gets the next random `Double` value uniformly distributed between 0 (inclusive) and 1 (exclusive).

```kotlin
public open fun nextDouble(): Double = nextBits(26) / (1 shl 24).toFloat()
```

### `nextLong()` Method

**Generates a `Long` random value uniformly distributed between `0` (inclusive) and the specified `[until]` bound (exclusive).**

```kotlin
public open fun nextLong(from: Long, until: Long): Long {
    checkRangeBounds(from, until)
    val n = until - from
    if (n > 0 || n == Long.MIN_VALUE) {
        val nLow = n.toInt()
        val nHigh = (n ushr 32).toInt()
        if (n and -n == n) {
            val nLow = n.toInt()
            val nHigh = (n ushr 32).toInt()
            when {
                nLow != 0 -> {
                    val bitCount = fastLog2(nLow)
                    nextBits(bitCount)
                }
                nHigh == 1 -> {
                    // toUInt().toLong()
                    nextInt().toLong() and 0xFFF_FFFF
                }
                else -> {
                    val bitCount = fastLog2(n)
                    nextBits(bitCount).toLong() and 0xFFF_FFFF
                }
            } else {
                var v: Long
                do {
                    val bits = nextLong().ushr(1)
                    v = bits % n
                } while (bits - v + (n - 1) < 0)
                v
            }
        }
    } else {
        var v: Long
        do {
            val bits = nextLong().ushr(1)
            v = bits % n
        } while (bits - v + (n - 1) < 0)
        val rnd = when {
            nLow != 0 -> {
                val bitCount = fastLog2(nLow)
                nextBits(bitCount)
            }
            nHigh == 1 -> {
                // toUInt().toLong()
                nextInt().toLong() and 0xFFF_FFFF
            }
            else -> {
                val bitCount = fastLog2(n)
                nextBits(bitCount).toLong() and 0xFFF_FFFF
            }
        } else {
            val bitCount = fastLog2(n)
            nextBits(bitCount)
        }
        return from + rnd
    }
}
```

### `nextBigDecimal()` Method

**Generates a `BigDecimal` random value uniformly distributed between 0 (inclusive) and 1 (exclusive).**

```kotlin
public open fun nextBigDecimal(): BigDecimal = nextDouble() / (1 shl 24).toFloat()
```

### `nextBoolean()` Method

**Generates a `Boolean` random value.**

```kotlin
public open fun nextBoolean(): Boolean = nextBits(1) != 0
```

### `nextInt()` Method

**Fills a subrange of the specified byte array starting from `[fromIndex]` inclusive and ending `[toIndex]` exclusive with random bytes.**

```kotlin
public open fun nextBytes(array: ByteArray, fromIndex: Int, toIndex: Int): ByteArray {
    require(fromIndex in 0..array.size && toIndex in 0..array.size) { "fromIndex ($fromIndex) or toIndex ($toIndex) are out of range: 0..<array.size." }
    val from = fromIndex
    val to = toIndex
    var v = nextDouble() * (to - from)
    if (v >= until) v = until.nextDown()
    return if (toIndex - fromIndex > 0) {
        createArray(array.size) { i -> array[(from + v).toInt()] }
    } else {
        createArray(array.size) { i -> 0 }
    }
}
```
```kotlin
fun randomBytes(array: ByteArray): ByteArray = random.nextInt(array.size) / 4

fun randomNumbers(fromIndex: Int, toIndex: Int): Int = random.nextInt(fromIndex, toIndex)

fun randomDouble(): Double = random.nextDouble()

fun randomInt(): Int = random.nextInt()

fun randomLong(): Long = random.nextLong()

fun randomBoolean(): Boolean = random.nextBoolean()

fun randomBytes(size: Int): ByteArray = random.nextBytes(size)

fun randomBytes(array: ByteArray, fromIndex: Int, toIndex: Int): ByteArray = random.nextBytes(array, fromIndex, toIndex)
```

---

```kotlin
fun randomString(length: Int): String = random.randomString(length)

fun randomList(size: Int): MutableList = random.randomList(size)

fun randomSet(size: Int): MutableSet = random.randomSet(size)
```
IllegalArgumentException("Cannot get random in empty range: $range")

the next random [ULong] from the random number generator in the specified [range].

* Generates a [ULong]

random value uniformly distributed in the specified [range]:

* from `range.start` inclusive to `range.endInclusive`

* @throws IllegalArgumentException if [range] is empty.

* @SinceKotlin("1.3")

* @ExperimentalUnsignedTypes

*public fun Random.nextULong(range: ULongRange):

* ULong = when {

* range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range: $range")

* range.last < ULong.MAX_VALUE -> nextULong(range.first, range.last + 1u)

* else -> nextULong(range.first - 1u, range.last) + 1u

* else -> nextULong()

*/

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextULong(range: ULongRange):

ULong = when {

range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range: $range")

range.last < ULong.MAX_VALUE -> nextULong(range.first, range.last + 1u)

else -> nextULong(range.first - 1u, range.last) + 1u

else -> nextULong()

*/

/**

* Fills

* the specified unsigned byte [array] with random bytes and returns it.

* @return [array] filled with random bytes.

*/

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUBytes(array: UByteArray): UByteArray {

nextBytes(array.asByteArray())

return array
}

/**

* Creates an unsigned byte array of the specified [size], filled with random bytes.

*/

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUBytes(size: Int): UByteArray = nextBytes(size).asUByteArray(

/**

* Fills a subrange of the specified `UByte` [array] starting from

* [fromIndex] inclusive and ending [toIndex] exclusive with random UBytes.

* @return [array] with the

* subrange filled with random bytes.

*/

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUBytes(array: UByteArray, fromIndex: Int = 0, toIndex: Int = array.size): UByteArray {

nextBytes(array.asByteArray(), fromIndex, toIndex)

return array
}

/**

* Represents a range of 

* [Comparable] values.

*/

private open class ComparableRange<T : Comparable<T>>(

override val start: T,

override val endInclusive: T
)

: ClosedRange<T> {

override fun equals(other: Any?): Boolean {

return other is ComparableRange<*> && (isEmpty() && other.isEmpty()

||

start == other.start && endInclusive == other.endInclusive)

}

override fun hashCode(): Int {

return if (isEmpty()) -1 else 31 * start.hashCode() + endInclusive.hashCode()

}

override fun toString(): String = "$start..$endInclusive"

}

/**

* Creates a range from this [Comparable] value to the

* specified [that] value.

* @sample samples.ranges.Ranges.rangeFromComparable

*/

public operator fun <T : Comparable<T>> T.rangeTo(that: T): ClosedRange<T> = ComparableRange(this, that)


/**

* Represents a range of floating point numbers.

*/

class Ranges


ClosedFloatingPointRange<

ClosedRange<

ClosedDoubleRange<

ClosedFloatRange<

// TODO: replace with qualifiedName

"Step must be positive, was: $step."

"Value cannot be cast to $qualifiedOrSimpleName"

"Can only be set at Kotlin level"
Open Source Used In DNAC Platform Antelope 9702
that can be found in the license/LICENSE.txt file.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

---

Open Source Used In DNAC Platform Antelope

---

* Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

---

Open Source Used In DNAC Platform Antelope

---

* Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors
are the same (as compared by the == operator)

- Applying the method [toUpperCae] to each character
  produces the same result
- Applying the method [toLowerCae] to each character
  produces the same result

* @sample samples.text.Chrs.equals
* @numpy fun Char.equals(\text{other} \colon \text{Char}, \text{ignoreCase} \colon \text{Boolean} = \text{false})

\begin{verbatim}
Boolean \text{true} \text{if (this == other) return true}
\text{else if (!ignoreCase) return false}
\text{else if (this.toUpperCase() == other.toUpperCase()) return true}
\text{else if (this.toLowerCase() == other.toLowerCase()) return true}
\text{return false}
\end{verbatim}

* @sample samples.text.Chars.equals

*/

public fun Char.equals(other: Char, ignoreCase: Boolean = false): Boolean {
    if (this == other) return true
    if (!ignoreCase) return false

    if (this.toUpperCase() == other.toUpperCase()) return true
    if (this.toLowerCase() == other.toLowerCase()) return true
    return false
}

/**
 * Returns `true` if this character is a Unicode surrogate code unit.
 */
public fun Char.isSurrogate(): Boolean = this in Char.MIN_SURROGATE..Char.MAX_SURROGATE

package kotlin.text

/**
 * Trims leading whitespace characters followed by \text{[marginPrefix]} from every line of a source
 * string and removes \text{\*first and the last blank lines.} \text{\*Doesn't affect a line if it doesn't contain [marginPrefix] except the first and the last blank lines.} \text{\*Doesn't preserve the original line endings.} \text{\*Param marginPrefix non-blank string, which is used as a margin delimiter. Default is } |
 */
public fun String.trimMargin(marginPrefix: String = "|"): String =
    replaceIndentByMargin("", marginPrefix)

/**
 * Replace indent by \text{newIndent} as it does \text{trimMargin} and replace it with \text{newIndent}.
 *
 * @param marginPrefix non-blank string, which is used as a margin delimiter.
 */
public fun String.replaceIndentByMargin(newIndent: String = " ", marginPrefix: String = "|"): String {
    require(marginPrefix.isNotBlank()) { "marginPrefix must be non-blank string." }
    val lines = lines()

    return lines.reindent(length + newIndent.length * lines.size,
        getIndentFunction(newIndent),
        [line ->
            line
        ]
    )
}

/**
 * Detects a common minimal indent of all the input lines, removes it from every line and
 * also removes the first and the last line if they are blank (notice difference blank vs empty).
 * @param marginPrefix non-blank string, which is used as a margin delimiter. Default is \text{\*Doesn't affect a line if it doesn't contain [marginPrefix] except the first and the last blank lines.} \text{\*Doesn't preserve the original line endings.} \text{\*Param marginPrefix non-blank string, which is used as a margin delimiter. Default is } |
 */
public fun String.trimIndent(): String = replaceIndent(""")

/**
 * Replace indent with \text{newIndent} as it does \text{trimIndent} and replaces it with \text{newIndent}.
 */
public fun String.replaceIndent(newIndent: String = " "): String {
    val lines = lines()

    val minCommonIndent = lines
        .filter(String::isNotBlank)
        .map(String::indentWidth)
        .minOrNull() ?: 0

    return lines.reindent(length + newIndent.length * lines.size,
        getIndentFunction(newIndent),
        [line ->
            line
        ]
    )
}

/**
 * Prepends \text{indent} to every line of the original string.
 * @param indent non-blank string, which is used as a margin delimiter. Default is \"\"
 */
public fun String.prependIndent(indent: String = "    "): String =
    lineSequence()
        .map {
            when {
                it.isBlank() -> {
                    when {
                        it.length < indent.length -> indent
                        else -> it
                    }
                }
                else -> indent + it
            }
        }
        .joinToString("\n")

private fun String.indentWidth(): Int = indexOfFirst { !it.isWhitespace() }.let { if (it == -1) length else it }

private fun getIndentFunction(indent: String) = when {
    indent.isEmpty() -> { line: String -> line }
    else -> { line: String -> indent + line }
}

private inline fun List<String>.reindent(
    resultSizeEstimate: Int,
    indentAddFunction: (String) -> String,
    indentCutFunction: (String) -> String?
): String {
    val lastIndex = lastIndex
    return mapIndexedNotNull { index, value ->
        if ((index == 0 || index == lastIndex) && value.isBlank())
            null
        else
            indentCutFunction(value)?.let(indentAddFunction) ?: value
    }
    .joinTo(StringBuilder(resultSizeEstimate), "\n")
    .toString()
Unicode symbols used in proper Typography.

```kotlin
public object Typography {
    /** The character \" — quotation mark */
    public const val quote: Char = '\u0022'
    /** The character "$ — dollar sign */
    public const val dollar: Char = '\u0024'
    /** The character & — ampersand */
    public const val amp: Char = '\u0026'
    /** The character < — less-than sign */
    public const val less: Char = '\u003C'
    /** The character > — greater-than sign */
    public const val greater: Char = '\u003E'
    /** The non-breaking space character */
    public const val nbsp: Char = '\u00A0'
    /** The character × */
    public const val times: Char = '\u00D7'
    /** The character ¢ */
    public const val cent: Char = '\u00A2'
    /** The character £ */
    public const val pound: Char = '\u00A3'
    /** The character § */
    public const val section: Char = '\u00A7'
    /** The character © */
    public const val copyright: Char = '\u00A9'
    /** The character « */
    public const val leftGuillemete: Char = '\u00AB'
    /** The character » */
    public const val rightGuillemete: Char = '\u00BB'
    /** The character ® */
    public const val registered: Char = '\u00AE'
    /** The character ° */
    public const val degree: Char = '\u00B0'
    /** The character ± */
    public const val plusMinus: Char = '\u00B1'
    /** The character ¶ */
    public const val paragraph: Char = '\u00B6'
    /** The character · */
    public const val middleDot: Char = '\u00B7'
    /** The character ½ */
    public const val half: Char = '\u00BD'
    /** The character – */
    public const val ndash: Char = '\u2013'
    /** The character — */
    public const val mdash: Char = '\u2014'
    /** The character ’ */
    public const val leftSingleQuote: Char = '\u2018'
    /** The character ' */
    public const val rightSingleQuote: Char = '\u2019'
    /** The character ' */
    public const val lowSingleQuote: Char = '\u201A'
    /** The character “ */
    public const val leftDoubleQuote: Char = '\u201C'
    /** The character ” */
    public const val rightDoubleQuote: Char = '\u201D'
    /** The character " */
    public const val lowDoubleQuote: Char = '\u201E'

    /** The character † */
    public const val dagger: Char = '\u2020'
    /** The character ‡ */
    public const val doubleDagger: Char = '\u2021'
    /** The character • */
    public const val bullet: Char = '\u2022'
    /** The character … */
    public const val ellipsis: Char = '\u2026'
    /** The character ′ */
    public const val prime: Char = '\u2032'
    /** The character ″ */
    public const val doublePrime: Char = '\u2033'
    /** The character € */
    public const val euro: Char = '\u20AC'
    /** The character ™ */
    public const val tm: Char = '\u2122'
    /** The character ≈ */
    public const val almostEqual: Char = '\u2248'
    /** The character ≠ */
    public const val notEqual: Char = '\u2260'
    /** The character ≤ */
    public const val lessOrEqual: Char = '\u2264'
    /** The character ≥ */
    public const val greaterOrEqual: Char = '\u2265'
}
```

*/

```kotlin
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.text

/**
 * Represents a collection of captured groups in a single match of a regular expression.
 * An element of the collection at the particular index can be `null`, if the corresponding group in the regular expression is optional and there was no match captured by that group.
 */
public abstract class MatchGroupCollection : Collection<MatchGroup?> {

    /**
     * Returns a group with the specified [index].
     * Groups are indexed from 1 to `groupCount` and group with the index 0 corresponds to the entire match.
     * An element of the collection at the particular index can be `null`, if the corresponding group in the regular expression is optional and there was no match captured by that group.
     */
    public abstract operator fun get(index: Int): MatchGroup?
}

/**
 * Extends [MatchGroupCollection] by introducing a way to get matched groups by name, when regex supports it.
 */
@SinceKotlin("1.1")
public interface MatchNamedGroupCollection : MatchGroupCollection {
    /**
     * Returns a named group with the specified [name].
     * @returns An instance of [MatchGroup] if the group with the specified [name] was matched or `null` otherwise.
     * @throws IllegalArgumentException if there is no group with the specified [name] defined in the regex pattern.
     * @throws UnsupportedOperationException if getting named groups isn't supported on the current platform.
     */
    public abstract fun get(name: String): MatchGroup?
}

/**
 * Represents the results from a single regular expression match.
 */
public interface
```
MatchResult

/** The range of indices in the original string where match was captured. */
public val range: IntRange

/** The substring from the input string captured by this match. */
public val value: String

*A collection of groups matched by the regular expression. This collection has size of `groupCount + 1` where `groupCount` is the count of groups in the regular expression. * Groups are indexed from 1 to `groupCount` and group with the index 0 corresponds to the entire match.

*public val public val groups: MatchGroupCollection

*A list of matched indexed group values. This list has size of `groupCount + 1` where `groupCount` is the count of groups in the regular expression. * Groups are indexed from 1 to `groupCount` and group with the index 0 corresponds to the entire match.

*If the group in the regular expression is optional and there were no match captured by that group, corresponding item in [groupValues] is an empty string.

*An instance of [MatchResult.Destructured] wrapper providing components for destructuring assignment of group values.

*component1 corresponds to the value of the first group, component2 \u2014 of the second, and so on.

*An instance of [MatchResult.Destructured] wrapper providing components for destructuring assignment of group values.

*Destructured get() = Destructured(this) * Returns a new [MatchResult] with the results for the next match, starting at the position

*at which the last match ended (at the character after the last matched character).

*public fun next(): MatchResult? * Provides components for destructuring assignment of group values.

*component1 corresponds to the value of the first group, \[component2 \u2014 of the second, and so on.

*If the group in the regular expression is optional and there were no match captured by that group, corresponding component value is an empty string.

*An instance of [MatchResult.Destructured] wrapper providing components for destructuring assignment of group values.

*Destructured get() = Destructured(this) * Returns a new [MatchResult] with the results for the next match, starting at the position

*at which the last match ended (at the character after the last matched character).

*public fun next(): MatchResult? * Provides components for destructuring assignment of group values.

*component1 corresponds to the value of the first group, \[component2 \u2014 of the second, and so on.

*If the group in the regular expression is optional and there were no match captured by that group, corresponding component value is an empty string.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.

*Destructured constructor(public val match: MatchResult) * Returns destructured group values as a list of strings.

* First value in the returned list corresponds to the value of the first group, and so on.
represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;
- `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;
- `minutes` represents the whole number of minutes in this duration.
- If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE], it is coerced into that range.

Public inline fun <T> toComponents(action: (minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {
    contract {
        callsInPlace(action, InvocationKind.EXACTLY_ONCE)
    }
    return action(inMinutes.toInt(),
        nanosecondsComponent, nanosecondsComponent) // Splits this duration into seconds, and nanoseconds and executes the given [action] with these components.
    * The result of [action] is returned as the result of this function.
    * `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000; `seconds` represents the whole number of seconds in this duration.
    * If the value doesn't fit in [Long] range, i.e. it's greater than [Long.MAX_VALUE] or less than [Long.MIN_VALUE], it is coerced into that range.
    * public inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Int) -> T): T {
        contract {
            callsInPlace(action, InvocationKind.EXACTLY_ONCE)
        }
        return action(inSeconds.toLong(),
            nanosecondsComponent) // conversion to units
    }

    @PublishedApi
    internal val hoursComponent: Int get() = (inHours % 24).toInt()
    @PublishedApi
    internal val minutesComponent: Int get() = (inMinutes % 60).toInt()
    @PublishedApi
    internal val secondsComponent: Int get() = (inSeconds % 60).toInt()
    @PublishedApi
    internal val nanosecondsComponent: Int get() = (inNanoseconds % 1e9).toInt()

    // conversion to units
    public fun toDouble(unit: DurationUnit): Double = convertDurationUnit(value, storageUnit, unit)
    public fun toLong(unit: DurationUnit): Long = toDouble(unit).toLong()
    public fun toInt(unit: DurationUnit): Int = toDouble(unit).toInt()
    public val inDays: Double get() = toDouble(DurationUnit.DAYS)
    public val inHours: Double get() = toDouble(DurationUnit.HOURS)
    public val inMinutes: Double get() = toDouble(DurationUnit.MINUTES)
    public val inSeconds: Double get() = toDouble(DurationUnit.SECONDS)
    public val inMilliseconds: Double get() = toDouble(DurationUnit.MILLISECONDS)
    public val inMicroseconds: Double get() = toDouble(DurationUnit.MICROSECONDS)
    public val inNanoseconds: Double get() = toDouble(DurationUnit.NANOSECONDS)

    // shortcuts
    public fun toLongNanoseconds(): Long = toLong(DurationUnit.NANOSECONDS)
    public fun toLongMilliseconds(): Long = toLong(DurationUnit.MILLISECONDS)
    // string representation
    public fun toString(): String = when {
        value < 1e-15 -> "0s"
        value < 1e+7 -> "Infinity" or "-Infinity"
        else -> value.toString() // using scientific notation
    }

Open Source Used In DNAC Platform Antelope 9708
```kotlin
/** Returns a [Duration] equal to this [Int] number of nanoseconds.

`Duration(convertDurationUnit(this, unit, storageUnit))`

// constructing from number of units
// extension functions

```
public val Int.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS)

/** Returns a [Duration] equal to this [Long] number of nanoseconds. */
public val Long.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS)

/** Returns a [Duration] equal to this [Double] number of nanoseconds. */
public val Double.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS)

/** Returns a [Duration] equal to this [Int] number of microseconds. */
public val Int.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)

/** Returns a [Duration] equal to this [Long] number of microseconds. */
public val Long.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)

/** Returns a [Duration] equal to this [Double] number of microseconds. */
public val Double.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)

/** Returns a [Duration] equal to this [Int] number of milliseconds. */
public val Int.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)

/** Returns a [Duration] equal to this [Long] number of milliseconds. */
public val Long.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)

/** Returns a [Duration] equal to this [Double] number of milliseconds. */
public val Double.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)

/** Returns a [Duration] equal to this [Int] number of seconds. */
public val Int.seconds get() =
toDuration(DurationUnit.SECONDS)

/** Returns a [Duration] equal to this [Long] number of seconds. */
public val Long.seconds get() =
toDuration(DurationUnit.SECONDS)

/** Returns a [Duration] equal to this [Double] number of seconds. */
public val Double.seconds get() =
toDuration(DurationUnit.SECONDS)

/** Returns a [Duration] equal to this [Int] number of minutes. */
public val Int.minutes get() =
toDuration(DurationUnit.MINUTES)

/** Returns a [Duration] equal to this [Long] number of minutes. */
public val Long.minutes get() =
toDuration(DurationUnit.MINUTES)

/** Returns a [Duration] equal to this [Double] number of minutes. */
public val Double.minutes get() =
toDuration(DurationUnit.MINUTES)

/** Returns a [Duration] equal to this [Int] number of hours. */
public val Int.hours get() =
toDuration(DurationUnit.HOURS)

/** Returns a [Duration] equal to this [Long] number of hours. */
public val Long.hours get() =
toDuration(DurationUnit.HOURS)

/** Returns a [Duration] equal to this [Double] number of hours. */
public val Double.hours get() =
toDuration(DurationUnit.HOURS)

/** Returns a duration whose value is the specified [duration] value multiplied by this number. */
public val Int@SinceKotlin("1.3")@n@ExperimentalTime

inline operator fun Int.times(duration: Duration): Duration = duration * this

/** Returns a duration whose value is the specified [duration] value multiplied by this number. */
public inline operator fun Double.times(duration: Duration): Duration = duration * this

// Other duration methods and properties...

// Other methods and properties...

// Other inline operator fun...
The list of possible time measurement units, in which a duration can be expressed:
* The smallest time unit is [NANOSECONDS] and the largest is [DAYS], which corresponds to exactly 24 [HOURS].
* Time unit representing one nanosecond, which is 1/1000 of a microsecond.
* Time unit representing one microsecond, which is 1/1000 of a millisecond.
* Time unit representing one millisecond, which is 1/1000 of a second.
* Time unit representing one second.
* Time unit representing one minute.
* Time unit representing one hour.
* Time unit representing one day, which is always equal to 24 hours.

The only operation provided by the time source is [markNow]. It returns a [TimeMark], which can be used to query the elapsed time later.

The returned [TimeMark] instance encapsulates the captured time point and allows querying the duration of time interval [elapsed][TimeMark.elapsedNow] from that point.

A source of time for measuring time intervals.

The only operation provided by the time source is [markNow]. It returns a [TimeMark], which can be used to query the elapsed time later.

The returned [TimeMark] instance encapsulates the captured time point and allows querying the duration of time interval [elapsed][TimeMark.elapsedNow] from that point.

A source of time for measuring time intervals.

The only operation provided by the time source is [markNow]. It returns a [TimeMark], which can be used to query the elapsed time later.

The returned [TimeMark] instance encapsulates the captured time point and allows querying the duration of time interval [elapsed][TimeMark.elapsedNow] from that point.

A source of time for measuring time intervals.

The only operation provided by the time source is [markNow]. It returns a [TimeMark], which can be used to query the elapsed time later.

The returned [TimeMark] instance encapsulates the captured time point and allows querying the duration of time interval [elapsed][TimeMark.elapsedNow] from that point.

A source of time for measuring time intervals.

The only operation provided by the time source is [markNow]. It returns a [TimeMark], which can be used to query the elapsed time later.

The returned [TimeMark] instance encapsulates the captured time point and allows querying the duration of time interval [elapsed][TimeMark.elapsedNow] from that point.

A source of time for measuring time intervals.
that point (see the function \([\text{elapsedNow}]\)).

```kotlin
/**
 * Returns the amount of time passed from this mark measured with the time source from which this mark was taken.
 */
@public abstract fun elapsedNow(): Duration
```

```kotlin
/**
 * Returns a time mark on the same time source that is ahead of this time mark by the specified \([\text{duration}]\).
 */
public open operator fun plus(duration: Duration): TimeMark = AdjustedTimeMark(this, duration)
```

```kotlin
/**
 * Returns a time mark on the same time source that is behind this time mark by the specified \([\text{duration}]\).
 */
public open operator fun minus(duration: Duration): TimeMark = plus(-duration)
```

```kotlin
/**
 * Returns true if this time mark has passed according to the time source from which this mark was taken.
 */
public fun hasPassedNow(): Boolean = !elapsedNow().isNegative()
```

```kotlin
/**
 * Returns false if this time mark has not passed according to the time source from which this mark was taken.
 */
public fun hasNotPassedNow(): Boolean = elapsedNow().isNegative()
```
An abstract class used to implement time sources that return their readings as [Double] values in the specified [unit].

This protected method should be overridden to return the current reading of the time source expressed as a [Double] number in the unit specified by the [unit] property.

A time source that has programmatically updatable readings. It is useful as a predictable source of time in tests.

The current reading value is stored as a [Long] number of nanoseconds, thus it's capable to represent a time range of approximately 292 years. Should the reading value overflow as the result of a [plusAssign] operation, an [IllegalStateException] is thrown.

This protected method should be overridden to return the current reading of the time source expressed as a [Double] number in the unit specified by the [unit] property.

A time source that has programmatically updatable readings. It is useful as a predictable source of time in tests.

The current reading value is stored as a [Long] number of nanoseconds, thus it's capable to represent a time range of approximately 292 years. Should the reading value overflow as the result of a [plusAssign] operation, an [IllegalStateException] is thrown.

A time source that has programmatically updatable readings. It is useful as a predictable source of time in tests.

The current reading value is stored as a [Long] number of nanoseconds, thus it's capable to represent a time range of approximately 292 years. Should the reading value overflow as the result of a [plusAssign] operation, an [IllegalStateException] is thrown.
InvocationKind.EXACTLY_ONCE)
    return TimeSource.Monotonic.measureTime(block)
}

/**
 * Executes the given function [block] and returns the duration of elapsed time interval.
 * The elapsed time is measured with the specified `this` [TimeSource] instance.
 */
@SinceKotlin("1.3")
@ExperimentalTime
public inline fun TimeSource.measureTime(block: () -> Unit): Duration {
    contract {
        callInPlace(block, InvocationKind.EXACTLY_ONCE)
    }

    val mark = markNow()
    block()
    return mark.elapsedNow()
}

/**
 * Data class representing a result of executing an action, along with the duration of elapsed time interval.
 * The elapsed time is measured with [TimeSource.Monotonic].
 */
@SinceKotlin("1.3")
@ExperimentalTime
public data class TimedValue<T>(val value: T, val duration: Duration)

/**
 * Executes the given function [block] and returns an instance of [TimedValue] class, containing both the result of the function execution and the duration of elapsed time interval.
 * The elapsed time is measured with [TimeSource.Monotonic].
 */
@SinceKotlin("1.3")
@ExperimentalTime
public inline fun <T> TimeSource.measureTimedValue(block: () -> T): TimedValue<T> {
    contract {
        callInPlace(block, InvocationKind.EXACTLY_ONCE)
    }

    val mark = markNow()
    val result = block()
    return TimedValue(result, mark.elapsedNow())
}

/**
 * Defines deep recursive function that keeps its stack on the heap, which allows very deep recursive computations that do not use the actual call stack.
 * To initiate a call to this deep recursive function use its [invoke] function.
 * As a rule of thumb, it should be used if recursion goes deeper than a thousand calls.
 * The [DeepRecursiveFunction] takes one parameter of type [T] and returns a result of type [R].
 * The [block] of code defines the body of a recursive function. In this block, the [callRecursive] extension can be used to make a recursive call to the declared function. Other instances of [DeepRecursiveFunction] can be called in this scope with the `callRecursive` extension, too.
 * For example, take a look at the following recursive tree class and a deeply recursive instance of this tree with 100K nodes:
 * ```
 * class Tree(val left: Tree? = null, val right: Tree? = null)
 * val deepTree = generateSequence(Tree()) { Tree(it) }.take(100_000).last()
 * ```
 * A regular recursive function can be defined to compute a depth of a tree:
 * ```
 * fun depth(t: Tree?): Int =
 *     if (t == null) 0 else max(depth(t.left), depth(t.right)) + 1
 * println(depth(deepTree)) // StackOverflowError
 * ```
 * However, the `depth` function can be rewritten using `DeepRecursiveFunction` in the following way, and then it successfully computes its value:
 * ```
 * val depth = DeepRecursiveFunction { t ->
 *     if (t == null) 0 else max(depth(t.left), depth(t.right)) + 1
 * }
 * println(depth(deepTree)) // Ok
 * ```
 * Deep recursive functions can also mutually call each other using a heap for the stack via the `callRecursive` extension. For example, the following pair of mutually recursive functions computes the number of tree nodes at even depth in the tree:
 * ```
 * val even = object {
 *     val odd: DeepRecursiveFunction = DeepRecursiveFunction { t ->
 *         if (t == null) 0 else odd.callRecursive(t.left) + odd.callRecursive(t.right) + 1
 *     }
 *     val deepTree = generateSequence(Tree()) { Tree(it) }.take(100_000).last()
 *     println(odd.callRecursive(deepTree).count()) // Ok
 *     println(odd.callRecursive(deepTree).count()) // Ok
 * ```
 * If this `depth` function is called for a `deepTree` it produces [StackOverflowError] because of deep recursion.
 * However, the `depth` function can be rewritten using `DeepRecursiveFunction` in the following way, and then it successfully computes its value:
 * ```
 * val even = DeepRecursiveFunction { t ->
 *     if (t == null) 0 else odd.callRecursive(t.left) + odd.callRecursive(t.right) + 1
 * }
 * ```
 * The function parameter type is [T] and it returns a result of type [R].
block: suspend DeepRecursiveScope<T, R>.deepRecursiveFunction.<T, R>(T) -> R

/**
 * Initiates a call to this deep recursive function,
 * forming a root of the call tree.
 * This operator should not be used from inside of [DeepRecursiveFunction] as it
 * uses the call stack slot for
 * initial recursive invocation. From inside of [DeepRecursiveFunction] use
 * [callRecursive][DeepRecursiveFunction.callRecursive].
 */
@SinceKotlin("1.4")
@ExperimentalStdlibApi
@RestrictsSuspension
public operator fun <T, R> DeepRecursiveFunction<T, R>.deepRecursiveFunction.<T, R>(T) -> R

/**
 * A scope class for [DeepRecursiveFunction] function declaration that defines [callRecursive] methods to
 * recursively call this function or another [DeepRecursiveFunction] putting the call activation frame on the heap.
 * @param [T] function parameter type.
 */
@RestrictsSuspension
@SinceKotlin("1.4")
public sealed class DeepRecursiveScope<T, R> {
    /**
     * Makes recursive call to this [DeepRecursiveFunction] function putting the call activation frame on the heap.
     * as opposed to the actual call stack that is used by a regular recursive
     * call.
     * @param [T] function parameter type.
     * @param [R] function result type.
     */
    public abstract suspend fun callRecursive(value: T): R

    /**
     * Makes call to the specified [DeepRecursiveFunction] function putting the call activation frame on the heap.
     * as opposed to the actual call stack that is used by a regular call.
     * @param [T] function parameter type.
     * @param [R] function result type.
     */

    @Deprecated(
        level = DeprecationLevel.ERROR,
        message =
"'invoke' should not be called from DeepRecursiveScope. " +
message = "Use 'callRecursive' to do recursion in
the heap instead of the call stack."

@Suppress("UNUSED_PARAMETER")
public operator fun DeepRecursiveFunction<Any, Any>.invoke(value: Any?): Nothing =
    throw UnsupportedOperationException("Should not be called from
DeepRecursiveScope\")
}

// ================== Implementation

@ExperimentalStdlibApi
private typealias DeepRecursiveFunctionBlock = suspend
DeepRecursiveScope<Any, Any>.deepRecursiveFunction.<Any, Any>(Any?) -> Any?

private val UNDEFINED_RESULT = Result.success(COROUTINE_SUSPENDED)

@Suppress("UNCHECKED_CAST")
@ExperimentalStdlibApi
private class DeepRecursiveScopeImpl<T, R>(
    block: suspend DeepRecursiveScope<T, R>(T) -> R,
    value: T
) : DeepRecursiveScope<T, R>(), Continuation<R> {
    // Active function block
    private var block: DeepRecursiveFunctionBlock = block as DeepRecursiveFunctionBlock

    // Value to call function
    private var value: Any? = value

    // Continuation of the current call
    private var cont: Continuation<Any?>? = this as Continuation<Any?>

    // Completion result (completion of the whole call stack)
    private var result: Result<Any?> = UNDEFINED_RESULT

    override val context: CoroutineContext
        get() = EmptyCoroutineContext

    override fun resumeWith(result: Result<R>) {
        this.cont = null
        this.result = result
    }

    override suspend fun callRecursive(value: T): R =
suspendCoroutineUninterceptedOrReturn {
        this.cont = this.cont as Continuation<Any?>
        this.value = value
        this.result = COROUTINE_SUSPENDED
    }

suspendCoroutineUninterceptedOrReturn {
        this.cont = this.cont as Continuation<Any?>
        this.value = value
        this.result = COROUTINE_SUSPENDED
    }

    private fun crossFunctionCompletion(
        currentFunction: DeepRecursiveFunctionBlock,
        cont: Continuation<Any?>
    ): Continuation<Any?> =
        Continuation(EmptyCoroutineContext) {
            this.cont = this.cont as Continuation<Any?>
            this.result = Result.success(COROUTINE_SUSPENDED)
        }

    @Suppress("UNCHECKED_CAST")
    private fun runCallLoop(): R {
        // Note: cont is set to null in DeepRecursiveScopeImpl.resumeWith
        when the whole computation completes
        val result = this.result
        val cont = this.cont
    }
return (result as Result<R>).getOrThrow() // done -- final result
   // The order of comparison is important
   here for that case of rogue class with broken equals
   if (UNDEFINED_RESULT == result) {
       // call \"function\" with \"value\" using \"completion\nval r = try {
       // This is
block.startCoroutine(this, value, cont)
function.startCoroutineUninterruptedOrReturn(this, value, cont)
   } catch (e: Throwable) {
   // return (result as Result<R>).getOrThrow() // done -- final result
   // The order of comparison is important
   here for that case of rogue class with broken equals
   if (UNDEFINED_RESULT == result) {
       // call "function" with "value" using "completion" call resume here
this.result = UNDEFINED_RESULT // reset
result back
cont.resumeWith(result)
}
}

// If the function returns without suspension -- calls its continuation immediately
if (r !== COROUTINE_SUSPENDED)
   return r as R
   // we returned from a
crossFunctionCompletion trampoline -- call resume here
this.result = UNDEFINED_RESULT // reset
result back
cont.resumeWith(result)
}
KProperty0<*>::isInitialized: Boolean
    get() = throw NotImplementedError("Implementation is intrinsic")

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 */

@file:kotlin.jvm.JvmName("LazyKt")
@file:kotlin.jvm.JvmMultifileClass
package kotlin

import kotlin.reflect.KProperty

/**
 * Represents a value with lazy initialization.
 *
 * To create an instance of [Lazy] use the [lazy] function.
 */
public interface Lazy<out T> {
    /**
     * Gets the lazily initialized value of the current Lazy instance.
     */
    public val value: T

    /**
     * Returns `true` if a value for this Lazy instance has been already initialized, and `false` otherwise.
     */
    public fun isInitialized(): Boolean
        = throw NotImplementedError("Implementation is intrinsic")

    // An extension to delegate a read-only property of type [T] to an instance of [Lazy].
    * *
    * This extension allows to use instances of Lazy for property delegation:
    * ```
    * prop: Lazy<Number>
    * prop.value
    * ```
    *
    * String by lazy { initializer }

    /**
     * Creates a new instance of the [Lazy] that is already initialized with the specified [value].
     */
    public fun lazyOf(value: T): Lazy<T> = InitializedLazyImpl(value)

    /**
     * An extension to delegate a read-only property of type [T] to an instance of [Lazy].
     * ```
     * prop: Lazy<Number>
     * prop.getValue()
     * ```
     */
    @kotlin.internal.InlineOnly
    public inline operator fun <T> Lazy<T>.getValue(thisRef: Any?, property: KProperty<*>): T = value

    /**
     * Specifies how a [Lazy] instance synchronizes initialization among multiple threads.
     */
    public enum class LazyThreadSafetyMode {

        /**
         * Locks are used to ensure that only a single thread can initialize the [Lazy] instance.
         */
        SYNCHRONIZED,

        /**
         * Initializer function can be called several times on concurrent access to uninitialized [Lazy] instance value;
         * but only the first returned value will be used as the value of [Lazy] instance value;
         * if the instance is accessed from multiple threads, its behavior is undefined.
         */
        PUBLICATION,

        /**
         * No locks are used to synchronize an access to the [Lazy] instance value;
         * if the instance is accessed from multiple threads, its behavior is undefined.
         */
        NONE,
    }

    /*
     * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
     */

    @file:kotlin.jvm.JvmMultifileClass
    @file:kotlin.jvm.JvmName("NumbersKt")
    package kotlin

    /**
     * Counts the number of set bits in the binary representation of this [Int] number.
     */
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun Int.countOneBits(): Int

    /**
     * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.
     */
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun Int.countLeadingZeroBits(): Int

    /**
     * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Int] number.
     */
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun Int.countTrailingZeroBits(): Int

    /**
     * Returns a number having a single bit set in the position of the most significant set bit of this [Int] number, or zero, if this number is zero.
     */
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun Int.takeHighestOneBit(): Int

    /**
     * Returns a number having a single bit set in the position of the least significant set bit of this [Int] number, or zero, if this number is zero.
     */
    @SinceKotlin("1.4")
    @WasExperimental(ExperimentalStdlibApi::class)
    public expect fun Int.takeLowestOneBit(): Int

    /*
     * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
     */

    Open Source Used In DNAC Platform Antelope 9717
*\@SinceKotlin("1.4.\")\@WasExperimental(ExperimentalStdlibApi::class)\@public expect fun Int.takeLowestOneBit(): Int\n**\n* Rotates the binary representation of this [Int] number left by the specified [bitCount] number of bits.\n\n*\n* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n\n* number.rotateLeft(-n) == number.rotateRight(n)\n\n* \n* Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n\n* number.rotateLeft(n) == number.rotateLeft(n % 32)\n\n* \n* number.rotateRight(n % 32) == number.rotateRight(n)\n\n\*/
@SinceKotlin("1.3.\")\@ExperimentalStdlibApi\@public expect fun Int.rotateLeft(bitCount: Int): Int\n\n**\n* Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits.\n\n* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n\n* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n\n* number.rotateRight(-n) == number.rotateLeft(n)\n\n* \n* Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n\n* number.rotateRight(n) == number.rotateRight(n % 32)\n\n\*/
@SinceKotlin("1.3.\")\@ExperimentalStdlibApi\@public expect fun Int.rotateRight(bitCount: Int): Int\n**\n* Counts the number of set bits in the binary representation of this [Int] number by right shifting by the specified [bitCount] number of bits.\n\n* The number is zero if the number is zero.\n\n@SinceKotlin("1.4.\")\@WasExperimental(ExperimentalStdlibApi::class)\@public expect fun Long.countOneBits(): Int\n**\n* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.\n\n* \n* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.\n\n@SinceKotlin("1.4.\")\@WasExperimental(ExperimentalStdlibApi::class)\@public expect fun Long.countLeadingZeroBits(): Int\n**\n* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.\n\n* \n* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.\n\n@SinceKotlin("1.4.\")\@WasExperimental(ExperimentalStdlibApi::class)\@public expect fun Long.countTrailingZeroBits(): Int\n**\n* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.\n\n* \n* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.\n\n@SinceKotlin("1.4.\")\@WasExperimental(ExperimentalStdlibApi::class)\@public inline fun Byte.countOneBits(): Int = (toInt() and 0xFF).countOneBits()\n**\n* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Byte] number.\n\n* \n* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Byte] number.\n\n@SinceKotlin("1.4.\")\@WasExperimental(ExperimentalStdlibApi::class)\@public inline fun Byte.countLeadingZeroBits(): Int = (toInt() and 0xFF).countLeadingZeroBits() - (Int.SIZE_BITS - Byte.SIZE_BITS)\n**\n* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Byte] number.
public inline fun Byte.countTrailingZeroBits(): Int = (toInt() or 0x100).countTrailingZeroBits() \n * Returns a number having a single bit set in the position of the most significant set bit of this [Byte] number, \n * or zero, if this number is zero. \n
public inline fun Byte.takeHighestOneBit(): Byte = (toInt() and 0xFF).takeHighestOneBit().toByte() \n * Returns a number having a single bit set in the position of the least significant set bit of this [Byte] number, \n * or zero, if this number is zero. \n
public inline fun Byte.takeLowestOneBit(): Byte = toInt().takeLowestOneBit().toByte() \n * Rotates the binary representation of this [Byte] number left by the specified [bitCount] number of bits, \n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side, \n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count, \n * number.rotateLeft(-n) == number.rotateRight(n) \n * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally \n * number.rotateLeft(n % 8) == number.rotateLeft(n) \n
public fun Byte.rotateLeft(bitCount: Int): Byte = (toInt().shl(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7))).toByte() \n * Rotates the number right by the specified [bitCount] number of bits, \n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side, \n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count, \n * number.rotateRight(-n) == number.rotateLeft(n) \n * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally \n * number.rotateRight(n % 8) == number.rotateRight(n) \n
public fun Byte.rotateRight(bitCount: Int): Byte = (toInt().shl(8 - (bitCount and 7)) or (toInt() and 0xFF).ushr(bitCount and 7)).toByte() \n * Rotates the number left by the specified [bitCount] number of bits, \n * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally \n * number.rotateRight(n % 8) == number.rotateRight(n) \n
public inline fun Short.countOneBits(): Int = (toInt() and 0xFFFF).countOneBits() \n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Short] number. \n
public inline fun Short.countLeadingZeroBits(): Int = (toInt() and 0xFFFF).countLeadingZeroBits() \n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Short] number. \n
public inline fun Short.countTrailingZeroBits(): Int = (toInt() or 0x10000).countTrailingZeroBits() \n * Returns a number having a single bit set in the position of the most significant set bit of this [Short] number, \n * or zero, if this number is zero. \n
public inline fun Short.takeHighestOneBit(): Short = (toInt() and 0xFFFF).takeHighestOneBit().toShort() \n * Rotates the binary representation of this [Short] number left by the specified [bitCount] number of bits, \n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side, \n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count, \n * number.rotateLeft(-n) == number.rotateRight(n) \n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally \n * number.rotateLeft(n % 16) == number.rotateLeft(n) \n
public fun Short.rotateLeft(bitCount: Int): Short =
(toInt().shl(bitCount and 15) or (toInt() and 0xFFFF).ushr(bitCount and 15)).toShort()

/**
 * Rotates the binary representation of this [Short] number right by the specified [bitCount] number of bits.
 * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.
 * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:
 * number.rotateRight(-n) == number.rotateLeft(n)
 * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally:
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun Short.rotateRight(bitCount: Int): Short =
    (toInt().shl(16 - (bitCount and 15)) or (toInt() and 0xFFFF).ushr(bitCount and 15)).toShort()

package kotlin
import kotlin.internal.RequireKotlin
import kotlin.internal.RequireKotlinVersionKind
import kotlin.ranges

@file:kotlin.jvm.JvmName("TuplesKt")

package kotlin

/**
 * Represents a generic pair of two values.
 * There is no meaning attached to values in this class, it can be used for any purpose.
 * Pair exhibits value semantics, i.e. two pairs are equal if both components are equal.
 * An example of decomposing it into values:
 * @sample samples.misc.Tuples.pairDestructuring
 *
 * @param A type of the first value.
 * @param B type of the second value.
 * @constructor Creates a new instance of Pair.
 */
public data class Pair<out A, out B>(
    public val first: A,
    public val second: B
) : Serializable {

    /**
     * Returns string representation of the [Pair] including its [first] and [second] values.
     */
    public override fun toString(): String = "($first, $second)"
}

/**
 * Creates a tuple of type [Pair] from this and [that].
 * This can be useful for creating [Map] literals with less noise, for example:
 * @sample samples.collections.Maps.Instantiation.mapFromPairs
 */
public infix fun <A, B> A.to(that: B): Pair<A, B> = Pair(this, that)

/**
 * Converts this pair into a list.
 * @sample samples.misc.Tuples.pairToList
 */
public fun <T> Pair<T, T>.toList(): List<T> = listOf(first, second)

package kotlin

/**
 * Represents a triad of values
 * There is no meaning attached to values in this class, it can be used for any purpose.
 * Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.
 * An example of decomposing it into values:
 * @sample samples.misc.Tuples.tripleDestructuring
 *
 * @param A type of the first value.
 * @param B type of the second value.
 * @param C type of the third value.
 */
public data class Triple<out A, out B, out C>(
    public val first: A,
    public val second: B,
    public val third: C
) : Serializable {

    /**
     * Returns string representation of the [Triple] including its [first], [second] and [third] values.
     */
    public override fun toString(): String = "($first, $second, $third)"
}

/**
 * Converts this triple into a list.
 * @sample samples.misc.Tuples.tripleToList
 */
public fun <T> Triple<T, T, T>.toList(): List<T> = listOf(first, second, third)
toString(): String = "$first..$last"

    companion object {
        /** An empty range of values of type UInt. */

        public val EMPTY: UIntRange = UIntRange(UInt.MAX_VALUE, UInt.MIN_VALUE)
    }

    /**
     * A progression of values of type `UInt`.
     */
    public open class UIntProgression
    internal constructor(
        start: UInt,
        endInclusive: UInt,
        step: Int
    ) : Iterable<UInt>{
        init {
            if (step == 0.toInt()) throw kotlin.IllegalArgumentException("Step must be non-zero.")
            if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to avoid overflow on negation.")
            if (step > 0) first <= last or else first >= last
        }

        /**
         * The first element in the progression.
         */
        public val first: UInt = start

        /**
         * The last element in the progression.
         */
        public val last: UInt = getProgressionLastElement(start, endInclusive, step)

        /**
         * The step of the progression.
         */
        public val step: Int = step

        override fun iterator(): UIntIterator = UIntProgressionIterator(first, last, step)

        /** Checks if the progression is empty. */
        public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last

        override fun equals(other: Any?): Boolean =
            other is UIntProgression && (isEmpty() && other.isEmpty() ||
            first == other.first && last == other.last && step == other.step)

        override fun hashCode(): Int =
            if (isEmpty()) -1 else (31 * (31 * first.toInt() + last.toInt()) + step.toInt())

        override fun toString(): String = if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"

        companion object {
            /**
             * Creates UIntProgression within the specified bounds of a closed range.
             * * A progression starts with the `rangeStart` value and goes toward the `rangeEnd` value not excluding it, with the specified `step`.
             * * In order to go backwards the `step` must be negative.
             * * An iterator over a progression of values of type `UInt`, @property step is the number by which the value is incremented on each step.
             * * An iterator over a sequence of values of type `UByte`.
             * * An iterator over a sequence of values of type `UShort`.
             * * An iterator over a sequence of values of type `UInt`.
             * * An iterator over a sequence of values of type `ULong`.
             */

            public fun fromClosedRange(rangeStart: UInt, rangeEnd: UInt, step: Int): UIntProgression = UIntProgression(rangeStart, rangeEnd, step)
        }
    }

    /** An iterator over a sequence of values of type `UByte`. */
    @SinceKotlin("1.3")
    @ExperimentalUnsignedTypes
    private class UByteIterator : Iterator<UByte> {
        override final fun next() = nextUByte()

        /** Returns the next value in the sequence without boxing. */
        public abstract fun nextUByte(): UByte
    }

    /** An iterator over a sequence of values of type `UShort`. */
    @SinceKotlin("1.3")
    @ExperimentalUnsignedTypes
    private class UShortIterator : Iterator<UShort> {
        override final fun next() = nextUShort()

        /** Returns the next value in the sequence without boxing. */
        public abstract fun nextUShort(): UShort
    }

    /** An iterator over a sequence of values of type `UInt`. */
    @SinceKotlin("1.3")
    @ExperimentalUnsignedTypes
    private class UIntIterator : Iterator<UInt> {
        override final fun next() = nextUInt()

        /** Returns the next value in the sequence without boxing. */
        public abstract fun nextUInt(): UInt
    }

    /** An iterator over a sequence of values of type `ULong`. */
    @SinceKotlin("1.3")
    @ExperimentalUnsignedTypes
    private class ULongIterator : Iterator<ULong> {
        override final fun next() = nextULong()

        /** Returns the next value in the sequence without boxing. */
        public abstract fun nextULong(): ULong
    }

    import kotlin.internal.*
/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun min(a: UInt, b: UInt): UInt {
    return minOf(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun max(a: UInt, b: UInt): UInt {
    return maxOf(a, b)
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun min(a: ULong, b: ULong): ULong {
    return minOf(a, b)
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun max(a: ULong, b: ULong): ULong {
    return maxOf(a, b)
}
*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun UInt.countOneBits(): Int = toInt().countOneBits()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun UInt.countLeadingZeroBits(): Int = toInt().countLeadingZeroBits()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun UInt.countTrailingZeroBits(): Int = toInt().countTrailingZeroBits()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun UInt.takeHighestOneBit(): UInt = toInt().takeHighestOneBit().toUInt()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun UInt.takeLowestOneBit(): UInt = toInt().takeLowestOneBit().toUInt()

*\@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun UInt.rotateLeft(bitCount: Int): UInt = toInt().rotateLeft(bitCount).toUInt()

*\@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun UInt.rotateRight(bitCount: Int): UInt = toInt().rotateRight(bitCount).toUInt()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun ULong.countOneBits(): Int = toLong().countOneBits()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun ULong.countLeadingZeroBits(): Int = toLong().countLeadingZeroBits()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun ULong.countTrailingZeroBits(): Int = toLong().countTrailingZeroBits()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun ULong.takeHighestOneBit(): ULong = toLong().takeHighestOneBit().toULong()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun ULong.takeLowestOneBit(): ULong = toLong().takeLowestOneBit().toULong()

*\@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun ULong.rotateLeft(bitCount: Int): ULong = toLong().rotateLeft(bitCount).toULong()

*\@SinceKotlin("1.3")\n@ExperimentalStdlibApi\n@ExperimentalUnsigned Types\n@kotlin.internal.InlineOnly\n
public inline fun ULong.rotateRight(bitCount: Int): ULong = toLong().rotateRight(bitCount).toULong()

*\@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n
public inline fun ULong.countOneBits(): Int = toLong().countOneBits()
least significant set bit of this [ULong] number,\n or zero, if this number is zero.\n
`takeLowestOneBit` function takes a ULong number and returns the number with the least significant bit set to zero. If the number is zero, it returns zero.

`takeLowestOneBit` function takes a ULong number and returns the number with the least significant bit set to zero. If the number is zero, it returns zero.

`rotateLeft` function rotates a ULong number left by a specified number of bits. It returns the resulting number.

`rotateRight` function rotates a ULong number right by a specified number of bits. It returns the resulting number.

`countOneBits` function counts the number of set bits in a UByte number.

`countLeadingZeroBits` function counts the number of consecutive most significant bits that are zero in a UByte number.

`countTrailingZeroBits` function counts the number of consecutive least significant bits that are zero in a UByte number.

`takeHighestOneBit` function takes a UByte number and returns the number with the most significant set bit set to one. If the number is zero, it returns zero.

`takeLowestOneBit` function takes a UByte number and returns the number with the least significant set bit set to one. If the number is zero, it returns zero.

`rotateLeft` function rotates a UByte number left by a specified number of bits. It returns the resulting number.

`rotateRight` function rotates a UByte number right by a specified number of bits. It returns the resulting number.
'number.rotateRight(-n) == number.rotateLeft(n)' *n * Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally

'number.rotateRight(n) == number.rotateRight(n % 8)' *n

Counts the number of set bits in the binary representation of this [UShort] number.

'number.rotateRight(bitCount: Int) UByte = toByte().rotateRight(bitCount).toUByte()' *n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [UShort] number.

'number.rotateRight(bitCount: Int) UShort = toShort().rotateRight(bitCount).toUShort()' *n

Counts the number of consecutive least significant bits that are zero in the binary representation of this [UShort] number.

'differenceModulo(a: UInt, b: UInt, c: UInt): UInt {
    val ac = a % c
    val bc = b % c
    return if (ac >= bc) ac - bc else ac - bc + c
}'

'differenceModulo(a: ULong, b: ULong, c: ULong): ULong {
    val ac = a % c
    val bc = b % c
    return if (ac >= bc) ac - bc else ac - bc + c
}'

'UShort.countOneBits(): Int = toUInt().countOneBits()' *n

'UShort.countLeadingZeroBits(): Int = toShort().countLeadingZeroBits()' *n

'UShort.countTrailingZeroBits(): Int = toShort().countTrailingZeroBits()' *n

'UShort.takeHighestOneBit(): UShort = toInt().takeHighestOneBit().toUShort()' *n

'UShort.takeLowestOneBit(): UShort = toInt().takeLowestOneBit().toUShort()' *n

'UShort.rotateLeft(bitCount: Int) UShort = toShort().rotateLeft(bitCount).toUShort()' *n

'UShort.rotateRight(bitCount: Int) UShort = toShort().rotateRight(bitCount).toUShort()' *n

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. *n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
ending bound for the progression
* @param step increment, or difference of successive elements in the
progression
* @return the final element of the progression
* @suppress

@PublishedApi
@SinceKotlin("1.3")
unint fun getProgressionLastElement(start: UInt, end: UInt, step: Int): UInt = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toUInt())
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toUInt())
    else -> throw kotlin.IllegalArgumentException("Step is zero.")
}

/**
 * Calculates the final element of a bounded
 * arithmetic progression, i.e. the last element of the progression which is in the range
 * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative
 * [step].
 * No validation on passed parameters is performed. The given parameters should satisfy the condition:
 * - either `step > 0` and `start <= end`,
 * - or `step < 0` and `start >= end`.
 * @param start first element of the progression
 * @param end ending bound for the progression
 * @param step increment, or difference of successive elements in the
 * progression
 * @return the final element of the progression
 * @suppress

@PublishedApi
@SinceKotlin("1.3")
unint fun getProgressionLastElement(start: ULong, end: ULong, step: Long): ULong = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toULong())
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toULong())
    else -> throw kotlin.IllegalArgumentException("Step is zero.")

@file:kotlin.jvm.JvmName("UStringsKt")  // string
representation of unsigned numbers

package kotlin.text

/**
 * Returns a string representation of this [Byte]
 * value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix
 * for number to string conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public /*inline*/ fun UByte.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Short]
 * value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix
 * for number to string conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public /*inline*/ fun UShort.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Int]
 * value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix
 * for number to string conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public /*inline*/ fun UInt.toString(radix: Int): String = this.toLong().toString(radix)

/**
 * Returns a string representation of this [Long]
 * value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix
 * for number to string conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULong.toString(radix: Int): String = ulongToString(this.toLong(), checkRadix(radix))

/**
 * Parses the string as a signed [UByte] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation
 * of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUByte(): UByte = toUByteOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a signed [UByte] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation
 * of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix
 * for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUByte(radix: Int): UByte = toUByteOrNull(radix) ?: numberFormatError(this)

/**
 * Parses the string as a [UShort] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation
 * of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUShort(): UShort = toUShortOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a [UShort] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation
 * of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix
 * for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUShort(radix: Int): UShort = toUShortOrNull(radix) ?: numberFormatError(this)
result.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

* @throws NumberFormatException if the string is not a valid representation of a number.

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
```kotlin
val length = this.length
if (length == 0) return null

val limit: ULong = ULong.MAX_VALUE
val start: Int
val firstChar = this[0]
if (firstChar < '0') {
  if (length == 1 || firstChar != '+') return null
  start = 1
} else {
  start = 0
}

val limitForMaxRadix = 51240955760343100UL // limit / 36
val limitBeforeMul = limitForMaxRadix
val uradix = radix.toULong()
val result = 0UL
for (i in start until length) {
  val digit = digitOf(this[i], radix)
  if (digit < 0) return null
  if (result > limitBeforeMul) {
    if (limitBeforeMul == limitForMaxRadix) {
      limitBeforeMul = limit / uradix
    } else {
      return null
    }
  }

  result *= uradix
  val beforeAdding = result
  result += digit.toUInt()
  if (result < beforeAdding) return null // overflow has happened
}

return result
```

---

```
@Target(CLASS, ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, TYPEALIAS)
@Retention(AnnotationRetention.BINARY)
@RequireKotlin("1.2.50")
@RequireKotlinVersionKind

public annotation class ExperimentalUnsignedTypes
```

---

```
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("MathKt")

package kotlin.math

/** Ratio of the circumference of a circle to its diameter, approximately 3.14159. */
@SinceKotlin("1.2")
public const val PI: Double = 3.141592653589793
/** Base of the natural logarithms, approximately 2.71828. */
@SinceKotlin("1.2")
public const val E: Double = 2.718281828459045

```

---

```
fun sin(x: Double): Double
fun cos(x: Double): Double
fun tan(x: Double): Double
fun asin(x: Double): Double
fun acos(x: Double): Double
fun atan(x: Double): Double
fun atan2(y: Double, x: Double): Double
```
for \( x < 0 \) \( \Rightarrow \) \( \arctan(-0.0, x) \) is \(-0.0\) for \( x > 0 \) and \( \arctan(-0.0, x) \) is \(0.0\) for \( 0 < x < +\text{Inf}\) and \(\arctan(-0.0, x) \) is \(-0.0\) for \(-\text{Inf} < y < 0.0\) \( \Rightarrow \) \( \arctan(-0.0, y) \) is \(\text{PI}\) for \( 0 < y < +\text{Inf}\) and \(\arctan(-0.0, y) \) is \(-0.0\) for \(-\text{Inf} < y < 0.0\) \( \Rightarrow \) \( \arctan(-0.0, y) \) is \(\text{PI}/2\) for \( y > 0 \) and \(\arctan(-0.0, y) \) is \(-\text{PI}/2\) for \( y < 0 \) \( \Rightarrow \) \( \arctan(0.0, x) \) is \(-\text{PI}/2\) for finite \( x \)\( \Rightarrow \) \( \arctan(0.0, x) \) is \(\text{PI}/2\) for finite \( x \)\n
\( \arctan(-\text{Inf} \cdot x) \) is \(-\text{PI}/2\) for finite \( x \)\n
\( \arctan(\text{NaN} \cdot x) \) and \( \arctan(\text{NaN}, y) \) is \(\text{NaN}\)

\( \\text{atan2}(y, x) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(y, 0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(y, -0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(y, +\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(y, -\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(y, 0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(y, +\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(y, -\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \)

\( \\text{atan2}(x, y) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, 0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, -0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, +\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, -\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, 0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, +\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, -\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \)

\( \\text{atan2}(x, y) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, 0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, -0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, +\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, -\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, 0.0) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, +\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \) \( \Rightarrow \) \( \text{atan2}(x, -\text{Inf}) \) is \(\text{NaN}\) for \( x = 0 \) and \( y \neq 0 \)
integer towards negative infinity.

* @return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.

* Special cases:
- floor(x) is 'x' where 'x' is 'NaN' or '+Inf' or '-Inf' or already a mathematical integer.

* Returns the absolute value of the given value [x].

* Special cases:
- abs(NaN) is 'NaN'
- if the result is 'NaN', then the result is 'NaN'

* Returns the value of this value: x

* Special case: NaN.sign is NaN

* Returns the absolute value of this value: x

* Special cases:
- NaN.absValue is NaN

* Returns the sign of this value:
- NaN.sign is NaN

* Returns the ulp (unit in the last place) of this value.

* Returns the Double value nearest to this value in direction of positive infinity.

* Returns the Double value nearest to this value in direction of negative infinity.
Double.roundToLong(): Long

// endregion

// region ================ Float Math

/** Computes the sine of the angle [x] given in radians.
 * @param x the angle whose sine is to be computed.
 * @return the sine of [x].
 */
public expect fun sin(x: Float): Float

/** Computes the cosine of the angle [x] given in radians.
 * @param x the angle whose cosine is to be computed.
 * @return the cosine of [x].
 */
public expect fun cos(x: Float): Float

/** Computes the tangent of the angle [x] given in radians.
 * @param x the angle whose tangent is to be computed.
 * @return the tangent of [x].
 */
public expect fun tan(x: Float): Float

/** Computes the secant of the angle [x] given in radians.
 * @param x the angle whose secant is to be computed.
 * @return the secant of [x].
 */
public expect fun sec(x: Float): Float

/** Computes the cosecant of the angle [x] given in radians.
 * @param x the angle whose cosecant is to be computed.
 * @return the cosecant of [x].
 */
public expect fun csc(x: Float): Float

/** Computes the cotangent of the angle [x] given in radians.
 * @param x the angle whose cotangent is to be computed.
 * @return the cotangent of [x].
 */
public expect fun cot(x: Float): Float

/** Computes the arcsine of the value [x];
 * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.
 * @param x the value whose arcsine is to be computed.
 * @return the arcsine of [x].
 */
public expect fun asin(x: Float): Float

/** Computes the arccosine of the value [x];
 * the returned value is an angle in the range from `0.0` to `PI` radians.
 * @param x the value whose arccosine is to be computed.
 * @return the arccosine of [x].
 */
public expect fun acos(x: Float): Float

/** Computes the arctangent of the value [x];
 * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.
 * @param x the value whose arctangent is to be computed.
 * @return the arctangent of [x].
 */
public expect fun atan(x: Float): Float

/** Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond to the rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x];
 * @param x the first coordinate.
 * @param y the second coordinate.
 * @return the angle `theta`.
 */
public expect fun atan2(y: Float, x: Float): Float

/** Computes the hyperbolic sine of the value [x].
 * @param x the value whose hyperbolic sine is to be computed.
 * @return the hyperbolic sine of [x].
 */
public expect fun sinh(x: Float): Float

/** Computes the hyperbolic cosine of the value [x].
 * @param x the value whose hyperbolic cosine is to be computed.
 * @return the hyperbolic cosine of [x].
 */
public expect fun cosh(x: Float): Float

/** Computes the hyperbolic tangent of the value [x].
 * @param x the value whose hyperbolic tangent is to be computed.
 * @return the hyperbolic tangent of [x].
 */
public expect fun tanh(x: Float): Float

/** Computes the inverse hyperbolic sine of the value [x];
 * @param x the value whose inverse hyperbolic sine is to be computed.
 * @return the inverse hyperbolic sine of [x].
 */
public expect fun asinh(x: Float): Float

/** Computes the inverse hyperbolic cosine of the value [x];
 * @param x the value whose inverse hyperbolic cosine is to be computed.
 * @return the inverse hyperbolic cosine of [x].
 */
public expect fun acosh(x: Float): Float

/** Computes the inverse hyperbolic tangent of the value [x];
 * @param x the value whose inverse hyperbolic tangent is to be computed.
 * @return the inverse hyperbolic tangent of [x].
 */
public expect fun atanh(x: Float): Float

/** Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.
 * @param x the first argument.
 * @param y the second argument.
 * @return `sqrt(x^2 + y^2)`.
 */
public expect fun hypot(x: Float, y: Float): Float

/** Computes the positive square root of the value [x].
 * @param x the value whose square root is to be computed.
 * @return the positive square root of [x].
 */
public expect fun sqrt(x: Float): Float

/** Computes Euler's number `e` raised to the power of the value [x].
 * @param x the value whose exponential is to be computed.
 * @return `e` raised to the power of [x].
 */
public expect fun exp(x: Float): Float

/** Computes `exp(x) - 1`.
 * @param x the value whose exponential minus one is to be computed.
 * @return `exp(x) - 1`.
 */
public expect fun expm1(x: Float): Float

/** Computes the logarithm of the value [x] to the given [base].
 * @param x the value whose logarithm is to be computed.
 * @param base the base of the logarithm.
 * @return the logarithm of [x] to the given [base].
 */
public expect fun log(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun ln(x: Float): Float

/** Computes the logarithm of the value [x] to the given [base].
 * @param x the value whose logarithm is to be computed.
 * @param base the base of the logarithm.
 * @return the logarithm of [x] to the given [base].
 */
public expect fun log10(x: Float, base: Float): Float

/** Computes the logarithm of the value [x] to the given [base].
 * @param x the value whose logarithm is to be computed.
 * @param base the base of the logarithm.
 * @return the logarithm of [x] to the given [base].
 */
public expect fun log2(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log1p(x: Float): Float

/** Computes the exponential of the value [x].
 * @param x the value whose exponential is to be computed.
 * @return the exponential of [x].
 */
public expect fun expm1(x: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun ln(x: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log10(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log2(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log1p(x: Float): Float

/** Computes the exponential of the value [x].
 * @param x the value whose exponential is to be computed.
 * @return the exponential of [x].
 */
public expect fun expm1(x: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun ln(x: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log10(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log2(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log1p(x: Float): Float

/** Computes the exponential of the value [x].
 * @param x the value whose exponential is to be computed.
 * @return the exponential of [x].
 */
public expect fun expm1(x: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun ln(x: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
 * @return the natural logarithm of [x].
 */
public expect fun log10(x: Float, base: Float): Float

/** Computes the natural logarithm of the value [x].
 * @param x the value whose natural logarithm is to be computed.
`log(x, b)` is `NaN` when `x < 0` or `b <= 0` or `b == 1.0` in * - `log(+Inf, +Inf)` is `NaN` in * - `log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1` in * - `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1` in *

See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

`log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b < 1` in *

* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

`log(0.0, +Inf)` is `+Inf`

* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

`log(0.0, +Inf)` is `+Inf`

* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

`log(0.0, +Inf)` is `+Inf`

* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

`log(0.0, +Inf)` is `+Inf`

* See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

See also logarithm functions for common fixed bases: [ln], [log10] and [log2].

`log(0.0, +Inf)` is `+Inf`
Int.MAX_VALUE <= x roundedToLong() == Int.MIN_VALUE * * @throws IllegalArgumentException when this value is NaN
*

* @SinceKotlin("1.2")
public expect fun Float.roundToLong(): Long

/**
 * Rounds this [Float] value to the nearest integer and converts the result to
 * [Long]. * * Ties are rounded towards positive infinity. * * Special cases:
 * - x roundedToLong() == Long.MAX_VALUE when x > Long.MAX_VALUE * * - x roundedToLong() == Long.MIN_VALUE when x < Long.MIN_VALUE * * @throws IllegalArgumentException when this value is NaN
 *
* @SinceKotlin("1.2")
public expect fun Float.roundToInt(): Int

/**
 * Returns the absolute value of the given value [n].
 * * Special cases:
 * - abs(Int.MIN_VALUE) is Int.MIN_VALUE due to an overflow
 *
* @see abs function
 */
@SinceKotlin("1.2")
public expect val Int.abs: Int

/**
 * Returns the sign of this value:
 * -1 if the value is negative,
 * 0 if the value is zero,
 * 1 if the value is positive
 */
@SinceKotlin("1.2")
public expect val Int.sign: Int


/**
 * Returns the absolute value of the given value [n].
 *
 * Special cases:
 * - abs(Long.MIN_VALUE) is Long.MIN_VALUE due to an overflow
 *
* @see abs function
 */
@SinceKotlin("1.2")
public expect val Long.abs: Long

/**
 * Returns the sign of this value:
 * -1 if the value is negative,
 * 0 if the value is zero,
 * 1 if the value is positive
 */
@SinceKotlin("1.2")
public expect val Long.sign: Int


// endregion

// region


// endregion

// endregion


Open Source Used In DNAC Platform Antelope 9735
Open Source Used In DNAC Platform Antelope 9740
Open Source Used In DNAC Platform Antelope 9784
Open Source Used In DNAC Platform Antelope
AhB,gB;QAaGB,cAAA,SAAHB,M;QAaSB,AIAI,UAAB,O;AAV,CAAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
;iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;:
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
iFAGX,gC;MAMOB,Q;MAAhB,IAUAB,O;AAV,C;AAJ,C;UAAwB,O;AAO,K;:MACRd,DAOAOi,K;
Open Source Used In DNAC Platform Antelope 9795
Open Source Used In DNAC Platform Antelope
Open Source Used In DNAC Platform Antelope 9831
Open Source Used In DNAC Platform Antelope 9888

Ag,C,mB;K;8DAEhC,Y;MACI,IAAJ,CAAC,kBAAL,C;QAAoB,MAAM,6B;MAE1B,eAAO,mCAAP,EAAO,YAA P;MACA,OAoA,wBAAL,WAJJ,CK;mAEGX,Y;MAAOc,sBAAQ,CAAR,I;K;+DAEpC,mB;MACI,wBAAL,YA AJ,EAaW,OAAX,C;MACA,mC;MACA,cAAO,E;K;+DGX,mB;M1DhFIJ,IAAIe0dIFu,gBAQQ,E1DjFIB,CAA J,C;QACl,c0DgFwB,4E;QfID/ExB,MAAM,6BAAsB,OAQQ,WA9B,C;00DGfW,BwBAAL,WAJJ,EAUU,OAACv C;/k;1Algb,B;+D;MACAf8,B;MAAFF,kB;MAAOc,4B;MACD,eaAY,B;MACGb,oCAAa,2BAkB,gBAAlB,EA6B O,OAAB7,E,AAC,cWAAK,KA3AC,C;MACH,eAAA,UAuG,wBAAV,IK;wDAGjB,0B;MACI,oCAAA,4B,AMmB,K ABnB,EA0B,Y,AIA1B,C;MAeB,WAAK,aaAI,mBAAY,KAAZ,IAAJ,EAUuB,OAAsB,MACL,mC;K;wDAgj,IB ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,OAoA,wBAk,MAAY,KAAZ,IAAL,C;K;6DAgX jB;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX jB;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX ;MACI,c0AAa,2BAkB,KAAlB,EAAY,B,YAZB,C;MAeB,AAWA,KAAZ,IAAJ,C;K;6DAgX 

Open Source Used in DNAC Platform Antelope 9888
Open Source Used In DNAC Platform Antelope

A system for managed and automated network services has been developed using open source products. The following sources have been used in the DNAC platform:

- Antelope

The system uses open source software to provide a scalable, efficient, and reliable service delivery platform. By leveraging open source products, the platform can be adapted to new requirements and standards, ensuring flexibility and cost-effectiveness.

This integration of open source components enhances the platform's capabilities, allowing for efficient resource management and improved network performance. The use of open source technology also facilitates collaboration and innovation within the network management community.
Found in path(s):
* /opt/cola/permits/1179183528_1625897252.78/0/kotlin-1-4-0-tgz/package/kotlin.js.map

1.2403 ajv 4.11.8
1.2403.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2404 esutils 2.0.3
1.2404.1 Available under license :

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2405 redux-thunk 2.3.0

1.2405.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2406 circular-json 0.3.3

1.2406.1 Available under license:

/*! 
* Copyright (C) 2013-2017 by Andrea Giammarchi - @WebReflection
*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

/*!

1.2407 listenablefuture 9999.0-empty-to-avoid-conflict-with-guava
1.2407.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of
fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source
code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form,
including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
1.2408 amqplib-tutorials 0.0.1 0.0.1

1.2408.1 Available under license:

amqplib copyright (c) 2013, 2014
    Michael Bridgen <mikeb@squarembius.net>

This package, "amqplib", is licensed under the MIT License. A copy may be found in the file LICENSE-MIT in this directory, or downloaded from http://opensource.org/licenses/MIT.
The MIT License (MIT)

Copyright (c) 2013, 2014 Michael Bridgen <mikeb@squarembius.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2409 set-immediate-shim 1.0.1

1.2409.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2410 object-map 1.0.1
1.2410.1 Available under license:

Copyright (c) 2014-2017, Jon Schlinkert, contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2411 js-base64 2.6.4
1.2411.1 Available under license:

Copyright (c) 2014, Dan Kogai

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of {{{project}}} nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2412 pause 0.0.1

1.2412.1 Available under license :
Copyright (c) 2012 TODO: Write your name

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.2413 cookie 0.3.1
1.2413.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2414 grpc-netty-shaded 1.27.1
1.2414.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.2415 @babel/code-frame 7.12.13

1.2415.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2416 has-symbols 1.0.1 1.0.1

1.2416.1 Available under license:
MIT License

Copyright (c) 2016 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2417 content-disposition 0.5.2 0.5.2
1.2417.1 Available under license :
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2418 yargs-parser 20.2.7 20.2.7
1.2418.1 Available under license :
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
1.2419 parquet-avro 1.8.1
1.2419.1 Available under license:

Apache Parquet Avro
Copyright 2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2420 url-parse 1.4.7

1.2420.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
1.2421 hive-contrib 3.1.2
1.2421.1 Available under license:

Hive Contrib
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2422 ansi-regex 2.1.1 2.1.1

1.2422.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2423 babel-preset-es2015 6.24.1

1.2423.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Jason Miller

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2424 react-tagsinput 3.19.0 3.19.0

1.2424.1 Available under license:

The MIT License

Copyright (c) 2015 Ola Holmström <olaholmstrom+github@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2425 path-is-absolute 1.0.1

1.2425.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2426 lodash-set 4.3.2

1.2426.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2427 object-keys 0.4.0 0.4.0

1.2427.1 Available under license :

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2428 json3 3.3.2 3.3.2
1.2428.1 Available under license:
Copyright (c) 2012-2014 Kit Cambridge.
http://kitcambridge.be/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2429 string_decoder 1.1.1
1.2429.1 Available under license:
Node.js is licensed for use as follows:

---

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2430 jackson-annotations 2.12.3
1.2430.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2431 forwarded 0.1.2

1.2431.1 Available under license :
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2432 commons-digester 2.1

1.2432.1 Available under license :
Apache Commons Digester
Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.

1.2433 es6-promise 4.2.8
1.2433.1 Available under license:
Copyright (c) 2014 Yehuda Katz, Tom Dale, Stefan Penner and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of 
this software and associated documentation files (the "Software"), to deal in 
the Software without restriction, including without limitation the rights to 
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies 
of the Software, and to permit persons to whom the Software is furnished to do 
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all 
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR 
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, 
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE 
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER 
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, 
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE 
SOFTWARE.

1.2434 repeat-element 1.1.3
1.2434.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of 
this software and associated documentation files (the "Software"), to deal in 
the Software without restriction, including without limitation the rights to 
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is 
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all 
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

------------------------------------------------------------------------

This product includes code from Twitter's ElephantBird project.
* parquet-hadoop's UnmaterializableRecordCounter.java includes code from ElephantBird's LzoRecordReader.java

Copyright: 2012-2014 Twitter
Home page: https://github.com/twitter/elephant-bird
License: http://www.apache.org/licenses/LICENSE-2.0

1.2436 es6-iterator 2.0.3
1.2436.1 Available under license:
The MIT License (MIT)

Copyright (C) 2013-2017 Mariusz Nowak (www.medikoo.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2437 poi 5.0.0
1.2437.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Open Source Used In DNAC Platform Antelope 10000
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and eachContributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
APACHE POI SUBCOMPONENTS:

Apache POI includes subcomponents with separate copyright notices and license terms. Your use of these subcomponents is subject to the terms and conditions of the following licenses:

Office Open XML schemas (poi-ooxml-full-*-jar)

The Office Open XML schema definitions used by Apache POI are a part of the Office Open XML ECMA Specification (ECMA-376, [1]). As defined in section 9.4 of the ECMA bylaws [2], this specification is available to all interested parties without restriction:

9.4 All documents when approved shall be made available to all interested parties without restriction.

Furthermore, both Microsoft and Adobe have granted patent licenses to this work [3,4,5].


Bouncy Castle library (bcprov-*-jar, bcpg-*-jar, bcpkix-*-jar)

Copyright (c) 2000 - 2021 The Legion of the Bouncy Castle Inc. (https://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CurvesAPI / Curve API

BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SLF4J library (slf4j-api-*.jar)

Copyright (c) 2004-2017 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

inbot-utils (https://github.com/Inbot/inbot-utils)

The MIT License (MIT)

Copyright (c) 2015 Inbot

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache POI
Copyright 2003-2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

This product contains parts that were originally based on software from BEA.
Copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/> (dead link),
which was acquired by Oracle Corporation in 2008.
<https://en.wikipedia.org/wiki/BEA_Systems>
This product contains W3C XML Schema documents. Copyright 2001-2003 (c) World Wide Web Consortium (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University)

This product contains the chunks_parse_cmds.tbl file from the vsdump program. Copyright (C) 2006-2007 Valek Filippov (frob@df.ru)


1.2438 rimraf 2.7.1
1.2438.1 Available under license :
   The ISC License

   Copyright (c) Isaac Z. Schlueter and Contributors

   Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

   THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2439 socket.io-adapter 0.5.0
1.2439.1 Available under license :
   (The MIT License)

   Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2440 parse-json 2.2.0
1.2440.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2441 @babel/plugin-proposal-logical-assignment-operators 7.13.8
1.2441.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2442 doctrine 1.5.0

1.2442.1 Available under license:

Doctrine
Copyright jQuery Foundation and other contributors, https://jquery.org/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

---

Open Source Used in DNAC Platform Antelope 10009
## Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. **Definitions.**

   "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your").) shall mean an individual or Legal Entity exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

   "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

   - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

   - THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2443 dns-prefetch-control 0.1.0 0.1.0

1.2443.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Evan Hahn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.


1.2444.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2445 is-my-json-valid 2.20.5

1.2445.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2446 kerby-asn1 1.0.0

1.2446.1 Available under license:

Kerby ASN1 Project
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
   distribution, then any Derivative Works that You distribute must
   include a readable copy of the attribution notices contained
   within such NOTICE file, excluding those notices that do not
   pertain to any part of the Derivative Works, in at least one
   of the following places: within a NOTICE text file distributed
   as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
   within a display generated by the Derivative Works, if and
   wherever such third-party notices normally appear. The contents
   of the NOTICE file are for informational purposes only and
   do not modify the License. You may add Your own attribution
   notices within Derivative Works that You distribute, alongside
   or as an addendum to the NOTICE text from the Work, provided
   that such additional attribution notices cannot be construed
   as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.2447 ansi-styles 4.3.0

1.2447.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2448 express 4.13.4 4.13.4
1.2448.1 Available under license:
(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2013-2014 Roman Shylman <shtylman+expressjs@gmail.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2449 react-highlight-words 0.17.0 0.17.0
1.2449.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Treasure Data

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2450 zstd-jni 1.4.9-1

1.2450.1 Available under license:

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Automatic-Module-Name: com.github.luben.zstd_jni
Bnd-LastModified: 1615191527618
Bundle-Description: JNI bindings for Zstd native library that provides fast and high compression lossless algorithm for Java and all JVM languages.
Bundle-License: https://opensource.org/licenses/BSD-2-Clause;description=BSD 2-Clause License
Bundle-ManifestVersion: 2
Bundle-Name: zstd-jni
Bundle-NativeCode: aix/ppc64/libzstd-jni.so;osname=AIX;processor=ppc64 , darwin/x86_64/libzstd-jni.dylib;osname=MacOS;osname=MacOSX;processor=x86_64 , darwin/aarch64/libzstd-jni.dylib;osname=MacOS;osname=MacOSX;processor=aarch64, freebsd/amd64/libzstd-jni.so;osname=FreeBSD;processor=amd64, freebsd/i386/libzstd-jni.so;osname=FreeBSD;processor=i386 , linux/aarch64/libzstd-jni.so;osname=Linux;processor=aarch64, linux/amd64/libzstd-jni.so;osname=Linux;processor=amd64, linux/arm/libzstd-jni.so;osname=Linux;processor=arm, linux/i386/libzstd-jni.so;osname=Linux;processor=i386, linux/mips64/libzstd-jni.so;osname=Linux;processor=mips64 , linux/ppc64/libzstd-jni.so;osname=Linux;processor=ppc64, linux/ppc64le/libzstd-jni.so;osname=Linux;processor=ppc64le, linux/i390/libzstd-jni.so;osname=Linux;processor=s390x, win/amd64/libzstd-jni.dll;osname=Win32;processor=amd64, win/x86/libzstd-jni.dll;osname=Win32;processor=x86
Bundle-SymbolicName: com.github.luben.zstd_jni
Bundle-Vendor: com.github.luben
Bundle-Version: 1.4.9.1
Created-By: 1.8.0_275 (Debian)
Export-Package: com.github.luben.zstd;version="1.4.9.1",com.github.luben.zstd.util;version="1.4.9.1"
Implementation-Title: zstd-jni
Implementation-Vendor: com.github.luben
Implementation-Vendor-Id: com.github.luben
Implementation-Version: 1.4.9-1
Import-Package: org.osgi.framework;resolution:=optional
Private-Package: linux.amd64,linux.i386,linux.aarch64,linux.arm,linux.ppc64,linux.ppc64le,linux.mips64,linux.s390x,aix.ppc64,darwin.x86_64,darwin.aarch64,win.amd64,win.x86,freebsd.amd64,freebsd.i386
Require-Capability: osgi.ee;filter="(osgi.ee=JavaSE)(version=1.8)"
Specification-Title: zstd-jni
Specification-Vendor: com.github.luben
Specification-Version: 1.4.9-1
Tool: Bnd-4.0.0.201805111645

Found in path(s):
* /opt/cola/permits/1176808707_1624967190.04/0/zstd-jni-1-4-9-1-jar/META-INF/MANIFEST.MF

1.2451 ret 0.1.15

1.2451.1 Available under license:
Copyright (C) 2011 by Roly Fentanes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2452 extglob 0.3.2

1.2452.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2453 figures 1.7.0
1.2453.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2454 mockito-core 2.23.4
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

====
Copyright (c) 2016 Mockito contributors
This program is made available under the terms of the MIT License.
====

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2455 lodash.flow 3.5.0 3.5.0

1.2455.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2456 aspect-j 1.8.13
1.2456.1 Available under license:

/* *******************************************************************
* Copyright (c) 2005 Contributors.
* All rights reserved.
* This program and the accompanying materials are made available
* under the terms of the Eclipse Public License v1.0
* which accompanies this distribution and is available at
*
* Contributors:
* Adrian ColyerInitial implementation
* *******************************************************************/
package org.aspectj.internal.lang.annotation;

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.
"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.
6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY
CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS),
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or
enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such
provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit)
alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes
such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such
litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or
conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and
distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this
Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the
Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the
right to publish new versions (including revisions) of this Agreement from time to time. No one other than the
Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement
Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable
separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program
(including Contributions) may always be distributed subject to the version of the Agreement under which it was
received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the
Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b)
above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this
Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United
States of America. No party to this Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.2457 elasticsearch-rest-client 7.0.1
1.2457.1 Available under license:

Elasticsearch
Copyright 2009-2018 Elasticsearch

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

This product includes software developed by
Joda.org (http://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate.
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets! The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2458 @babel/plugin-transform-regenerator

7.12.13

1.2458.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2459 amqplib 0.5.2 0.5.2

1.2459.1 Available under license:

amqplib copyright (c) 2013, 2014
Michael Bridgen <mikeb@squaremobius.net>

This package, "amqplib", is licensed under the MIT License. A copy may be found in the file LICENSE-MIT in this directory, or downloaded from http://opensource.org/licenses/MIT.
The MIT License (MIT)

Copyright (c) 2013, 2014 Michael Bridgen <mikeb@squaremobius.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2460 curl 7.51.0

1.2460.1 Available under license:

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2020, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

1.2461 xmpcore 6.0.6
1.2461.1 Available under license :
ADOBE CONFIDENTIAL

Copyright 2011-2016 Adobe Systems Incorporated
All Rights Reserved.

NOTICE: All information contained herein is, and remains the property of Adobe Systems Incorporated and its suppliers, if any. The intellectual and technical concepts contained herein are proprietary to Adobe Systems Incorporated and its suppliers and may be covered by U.S. and Foreign Patents, patents in process, and are protected by trade secret or copyright law. Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from Adobe Systems Incorporated.

1.2462 rx-lite 3.1.2
1.2462.1 Available under license :
No license file was found, but licenses were detected in source scan.

/* Copyright (c) Microsoft Open Technologies, Inc. All rights reserved. See License.txt in the project root for license information.*/

Found in path(s):
* /opt/cola/permits/1136088163_1613675018.74/0/rx-lite-3-1-2-2-tgz/package/rx.lite.min.js
No license file was found, but licenses were detected in source scan.

Copyright (c) Microsoft Open Technologies, Inc. All rights reserved.
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may
http://www.apache.org/licenses/LICENSE-2.0
distributed under the License is distributed on an "AS IS" BASIS.

Found in path(s):
* /opt/cola/permits/1136088163_1613675018.74/0/rx-lite-3-1-2-2-tgz/package/readme.md

No license file was found, but licenses were detected in source scan.

// Copyright (c) Microsoft Open Technologies, Inc. All rights reserved. See License.txt in the project root for license information.

Found in path(s):
* /opt/cola/permits/1136088163_1613675018.74/0/rx-lite-3-1-2-2-tgz/package/rx.lite.js

1.2463 is-path-inside 3.0.3
1.2463.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2464 @babel/helper-remap-async-to-generator 7.13.0
1.2464.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2465 asynckit 0.4.0

1.2465.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Alex Indigo

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2466 ipaddr.js 1.0.5 1.0.5
1.2466.1 Available under license:
Copyright (C) 2011 Peter Zotov <whitequark@whitequark.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2467 combined-stream 1.0.8
1.2467.1 Available under license:
Copyright (c) 2011 Debuggable Limited <felix@debuggable.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2468 simple-odata-server 1.1.1

1.2468.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Jan Blaha

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2469 @babel/plugin-transform-async-to-generator 7.13.0

1.2469.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2470 @babel/plugin-transform-new-target

7.12.13

1.2470.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2471 drill-format-httpd 1.19.0

1.2471.1 Available under license:

Drill : Contrib : Format : Httpd/Nginx Access Log
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2472 jetty-servlets 9.3.27.v20190418

1.2473 byte-buddy 1.9.3

1.2474 derby 10.14.1.0

1.2474.1 Available under license :

=====================================================================
== NOTICE file corresponding to section 4(d) of the Apache License,==
== Version 2.0, in this case for the Apache Derby distribution.==
==
== DO NOT EDIT THIS FILE DIRECTLY. IT IS GENERATED==
== BY THE buildnotice TARGET IN THE TOP LEVEL build.xml FILE.==
==
=====================================================================

Apache Derby
Copyright 2004-2017 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

=====================================================================

Portions of Derby were originally developed by
International Business Machines Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement", informally known as the "Derby CLA".
The following copyright notice(s) were affixed to portions of the code with which this file is now or was at one time distributed and are placed here unaltered.
The portion of the functionTests under 'nist' was originally developed by the National Institute of Standards and Technology (NIST), an agency of the United States Department of Commerce, and adapted by International Business Machines Corporation in accordance with the NIST Software Acknowledgment and Redistribution document at http://www.itl.nist.gov/div897/ctg/sql_form.htm

The Derby build relies on source files supplied by the Apache Felix project. The following notice covers the Felix files:

Apache Felix Main
Copyright 2008 The Apache Software Foundation

I. Included Software

This product includes software developed at The Apache Software Foundation (http://www.apache.org/). Licensed under the Apache License 2.0.

This product includes software developed at The OSGi Alliance (http://www.osgi.org/). Copyright (c) OSGi Alliance (2000, 2007). Licensed under the Apache License 2.0.

This product includes software from http://kxml.sourceforge.net. Copyright (c) 2002,2003, Stefan Haustein, Oberhausen, Rhld., Germany. Licensed under BSD License.

II. Used Software

This product uses software developed at The OSGi Alliance (http://www.osgi.org/). Copyright (c) OSGi Alliance (2000, 2007).
Licensed under the Apache License 2.0.

III. License Summary
- Apache License 2.0
- BSD License

The Derby build relies on jar files supplied by the Apache Lucene project. The following notice covers the Lucene files:

Apache Lucene
Copyright 2013 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/
JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
  analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
  analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
  http://snowball.tartarus.org/

The KStem stemmer in
  analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic, Persian, Romanian, Bulgarian, and Hindi analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
  analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
  analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
  analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
  analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
  analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt

The German, Spanish, Finnish, French, Hungarian, Italian, Portuguese, Russian and Swedish light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
  analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/es/SpanishMinimalStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianMinimalStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/it/ItalianMinimalStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
  analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
The Polish analyzer (stempel) comes with a default stopword list that is BSD-licensed created by the Carrot2 project. The file resides in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt. See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/) is derived from Unicode data such as the Unicode Character Database. See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary (http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP) (http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===========================================================================
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===========================================================================
mecab-ipadic-2.7.0-20070801 Notice
===========================================================================

Nara Institute of Science and Technology (NAIST),

and Edmond Nolan.
the copyright holders, disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall NAIST be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortuous action, arising out of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the
Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

The Derby build relies on a jar file supplied by the JSON Simple project, hosted at https://code.google.com/p/json-simple/. The JSON simple jar file is licensed under the Apache 2.0 License. No other notice covers that jar file.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2475 base64id 2.0.0

1.2475.1 Available under license :
(The MIT License)

Copyright (c) 2012-2016 Kristian Faeldt <faeldt_kristian@cyberagent.co.jp>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2476 react-redux 5.0.7 5.0.7

1.2476.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2477 p-cancelable 1.1.0

1.2477.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2478 metrics 4.1.0

1.2478.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
Copyright 2015 MongoDB Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2479 object.assign 4.1.0
1.2479.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
1.2480 serve-static 1.10.3
1.2480.1 Available under license:
(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2481 lodash 4.17.20 4.17.20
1.2481.1 Available under license:
Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2482 univocity-parsers 2.8.3

1.2482.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*******************************************************************************
 * Copyright 2015 Univocity Software Pty Ltd
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
*******************************************************************************/
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

******************************************************************************/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_29.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/record/RecordMetaData.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_34.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/ConcurrentRowProcessorTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/common/processor/OutputValueSwitch.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/OutputValueSwitch.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_21.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_23.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/ProcessorErrorHandler.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/commons/headers/Headers.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/annotations/EnumOptions.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/conversions/EnumSelector.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/conversions/EnumConversion.java
Open Source Used In DNAC Platform Antelope 10075

* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/support/Ticket_3.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/common/fields/FieldEnumSelectorTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_19.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_30.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_32.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_33.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/processor/ConcurrentRowProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/processor/NoopProcessorErrorHandler.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/examples/AnotherTestBean.java
No license file was found, but licenses were detected in source scan.

/*******************************************************************************
* Copyright 2014 Univocity Software Pty Ltd
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*******************************************************************************/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/sv/TsvParser.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
  gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/annotations/Trim.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/csv/CsvWriterTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/BatchedColumnProcessorTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/BeanListProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/core/AbstractObjectListProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_7.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/core/ColumnSplitter.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/input/concurrent/Entry.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/ArgumentUtils.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/fields/ExcludeFieldEnumSelector.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/input/CharAppender.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-target-
  uniVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/examples/Example.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/fixed/FixedWidthParser.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/MasterDetailProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/core/AbstractMasterDetailListProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/annotations/Convert.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_3.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/core/AbstractBeanProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/core/AbstractBeanListProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/ColumnProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/core/ColumnOrderDependent.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/conversions/LowerCaseConversion.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/conversions/UpperCaseConversion.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/conversions/FormattedConversion.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/conversions/CharacterConversion.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/csv/TestUtils.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/input/EOFException.java
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2017 Univocity Software Pty Ltd
 */
No license file was found, but licenses were detected in source scan.

*******************************************************************************
* Copyright 2016 Univocity Software Pty Ltd
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*******************************************************************************/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/examples/samples/CsvSearchExample.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_212.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_139.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_186.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_196.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/ResultIterator.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/ContextSnapshot.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/csv/Csv.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/examples/FixedWidthWriterExamples.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/csv/CsvSearchExample.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/ContextSnapshot.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-
gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/uniVocity/parsers/csv/UnescapedQuoteHandling.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/uniVocity/parsers/issues/support/Ticket_7.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/uniVocity/parsers/issues/github/Github_121.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/uniVocity/parsers/common/ParsingContextSnapshot.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/uniVocity/parsers/common/DataProcessingExceptionTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/uniVocity/parsers/examples/RoutineExamples.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/uniVocity/parsers/annotations/meta/MyReplacement.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/uniVocity/parsers/issues/github/Github_69.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/uniVocity/parsers/common/routine/InputDimension.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/uniVocity/parsers/issues/github/Github_201.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/uniVocity/parsers/issues/github/Github_100.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/test/java/com/uniVocity/parsers/common/AbstractWriterTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/uniVocity/parsers/fixed/FixedWidthFields.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/uniVocity/parsers/common/RetryableErrorHandler.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/uniVocity-univocity-parsers-v2-8-3-0-gcaf1775-targ
gz/uniVocity-univocity-parsers-a19fc0e/src/main/java/com/uniVocity/parsers/annotations/Copy.java
No license file was found, but licenses were detected in source scan.

/******************************************************************************
* Copyright 2018 Univocity Software Pty Ltd
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
***********************************************************************
No license file was found, but licenses were detected in source scan.

/**
 * ****************************************************************************
 * Copyright 2014 Univocity Software Pty Ltd
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the
 * License at
 */

Open Source Used In DNAC Platform Antelope 10092
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/TextParsingException.java
No license file was found, but licenses were detected in source scan.

/*******************************************************************************
 * Copyright 2015 Univocity Software Pty Ltd
 * <p>
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * <p>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p>
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
*******************************************************************************/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_24.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 Univocity Software Pty Ltd
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/record/RecordFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2015 Univocity Software Pty Ltd
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/beans/PropertyWrapper.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/beans/BeanHelper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 Univocity Software Pty Ltd
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/record/RecordFactory.java
*/
* Copyright 2016 Univocity Software Pty Ltd.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
*   http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/unIVocity-univocity-parsers-
a19fc0e/src/test/java/com/univocity/parsers/common/input/WriterCharAppenderTest.java

No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2019 Univocity Software Pty Ltd
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_309.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/unIVocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_315.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_328.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_337.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_336.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/fields/ColumnMapping.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_314.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/NormalizedStringTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/StringCache.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/fields/AbstractColumnMapping.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/fields/ColumnMapper.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/annotations/helpers/MethodDescriptor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/common/NormalizedStringTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/NormalizedString.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/fields/ColumnMapper.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/fields/AbstractColumnMapping.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_287.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2013 uniVocity Software Pty Ltd. All rights reserved.
 * This file is subject to the terms and conditions defined in file
 * 'LICENSE.txt', which is part of this source code package.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_299.java

No license file was found, but licenses were detected in source scan.

<title>Apache License, Version 2.0 - The Apache Software Foundation</title>
Apache License<br><a href="http://www.apache.org/licenses/">http://www.apache.org/licenses/</a><br>
Subject to the terms and conditions of attribution notices from the Source form of the Work, include a readable copy of the attribution notices contained do not modify the License. You may add Your own attribution that such additional attribution notices cannot be construed This License does not grant permission to use the trade Found in path(s):
*/
* Copyright (c) 2015. Univocity Software Pty Ltd
* <p/>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p/>
* http://www.apache.org/licenses/LICENSE-2.0
* <p/>
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
*/
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ.gz/univocity-univocity-parsers-a19fc0e/LICENSE-2.0.html
No license file was found, but licenses were detected in source scan.
*/
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ.gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/CustomMatcher.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ.gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/ConversionProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ.gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/common/processor/MultiBeanListProcessorTest.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ.gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/MultiBeanRowProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ.gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/MultiBeanProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ.gz/univocity-univocity-parsers-
a19fc0e/src/main/java/com/univocity/parsers/common/processor/MultiBeanListProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-
a19fc0e/src/main/java/com/univocity/parsers/common/processor/core/AbstractMultiBeanListProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_37.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_39.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/conversions/ToStringConversion.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/core/AbstractMultiBeanProcessor.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/processor/core/AbstractMultiBeanRowProcessor.java
No license file was found, but licenses were detected in source scan.

/
* Copyright (c) 2018. Univocity Software Pty Ltd
* <p/>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p/>
* http://www.apache.org/licenses/LICENSE-2.0
* <p/>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_255.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/common/ParsingContext.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/test/java/com/univocity/parsers/issues/github/Github_258.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/conversions/ValidatedConversion.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-targ-
gz/univocity-univocity-parsers-a19fc0e/src/main/java/com/univocity/parsers/annotations/Validate.java
* /opt/ws_local/PERMITS_SQL/1092334798_1600713115.55/0/univocity-univocity-parsers-v2-8-3-0-gcaf1775-tar-
No license file was found, but licenses were detected in source scan.

1.2483 llvm 6.0-1ubuntu2~16.04.1

1.2483.1 Available under license:

; RUN: llc < %s -mtriple=s390x-linux-gnu -mcpu=zEC12 -verify-machineinstrs | FileCheck %s
;
; Test that early if conversion produces LOCR with operands of the right
; register classes.

define void @autogen_SD4739(i8*) {
; CHECK-NOT: Expected a GR32Bit register, but got a GRX32Bit register
BB:
%L34 = load i8, i8* %0
%Cmp56 = icmp sgt i8 undef, %L34
br label %CF246

CF246:
; preds = %CF246, %BB
%Sl163 = select i1 %Cmp56, i8 %L34, i8 undef
br i1 undef, label %CF246, label %CF248

CF248:
; preds = %CF248, %CF246
store i8 %Sl163, i8* %0
br label %CF248
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=avx512vl,avx512bw,avx512dq,prefer-256-bit | FileCheck %s

; This file primarily contains tests for specific places in X86ISelLowering.cpp that needed be made aware of the
; legalizer not allowing 512-bit vectors due to prefer-256-bit even though AVX512 is enabled.

define void @add256(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-width"="256" {
; CHECK-LABEL: add256:
; CHECK: # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpadd (%rsi), %ymm0, %ymm0
; CHECK-NEXT: vpadd 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%d = load <16 x i32>, <16 x i32>* %a
%e = load <16 x i32>, <16 x i32>* %b
%f = add <16 x i32> %d, %e
store <16 x i32> %f, <16 x i32>* %c
ret void
}

define void @add512(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-width"="512" {
; CHECK-LABEL: add512:
; CHECK: # %bb.0:
; CHECK-NEXT: vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT: vpadd (%rsi), %zmm0, %zmm0
; CHECK-NEXT: vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%d = load <16 x i32>, <16 x i32>* %a
%e = load <16 x i32>, <16 x i32>* %b
%f = add <16 x i32> %d, %e
store <16 x i32> %f, <16 x i32>* %c
ret void
}

define void @avg_v64i8_256(<64 x i8>* %a, <64 x i8>* %b) "min-legal-vector-width"="256" {
; CHECK-LABEL: avg_v64i8_256:
; CHECK: # %bb.0:
; CHECK-NEXT: vmovdqa (%rsi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT: vpavgb (%rdi), %ymm0, %ymm0
; CHECK-NEXT: vpavgb 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT: vmovdqu %ymm1, (%rax)
; CHECK-NEXT: vmovdqu %ymm0, (%rax)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%1 = load <64 x i8>, <64 x i8>* %a
%2 = load <64 x i8>, <64 x i8>* %b
%3 = zext <64 x i8> %1 to <64 x i32>
%4 = zext <64 x i8> %2 to <64 x i32>
%5 = add nuw nsw <64 x i32> %3, %3.<i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
%6 = add nuw nsw <64 x i32> %5, %4
%7 = lshr <64 x i32> %6, %6.<i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
%8 = trunc <64 x i32> %7 to <64 x i8>
store <64 x i8> %8, <64 x i8>* undef, align 4
ret void
define void @avg_v64i8_512(<64 x i8>* %a, <64 x i8>* %b) "min-legal-vector-width"="512" {
    ; CHECK-LABEL: avg_v64i8_512:
    ; CHECK:       # %bb.0:
    ; CHECK-NEXT: vmovdqa64 (%rsi), %zmm0
    ; CHECK-NEXT: vpavgb (%rdi), %zmm0, %zmm0
    ; CHECK-NEXT: vmovdqu64 %zmm0, (%rax)
    ; CHECK-NEXT: vzeroupper
    ; CHECK-NEXT: retq
    %1 = load <64 x i8>, <64 x i8>* %a
    %2 = load <64 x i8>, <64 x i8>* %b
    %3 = zext <64 x i8> %1 to <64 x i32>
    %4 = zext <64 x i8> %2 to <64 x i32>
    %5 = add nuw nsw <64 x i32> %3, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1, %i32 1>, %odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32 12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
    store <64 x i8> %8, <64 x i8>* undef, align 4
    retq
}
%even = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 1, i32 3, i32 5, i32 7, i32 9, i32 11, i32 13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32 27, i32 29, i32 31>
%ret = add <16 x i32> %odd, %even
store <16 x i32> %ret, <16 x i32>* %CPtr
ret void
}
define void @pmaddwd_32_512(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr) "min-legal-vector-width"="512" {
; CHECK-LABEL: pmaddwd_32_512:
; CHECK:    # %bb.0:
; CHECK-NEXT: vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT: vpmaddwd (%rsi), %zmm0, %zmm0
; CHECK-NEXT: vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%A = load <32 x i16>, <32 x i16>* %APtr
%B = load <32 x i16>, <32 x i16>* %BPtr
%a = sext <32 x i16> %A to <32 x i32>
%b = sext <32 x i16> %B to <32 x i32>
%m = mul nsw <32 x i32> %a, %b
%odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32 12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
%even = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 1, i32 3, i32 5, i32 7, i32 9, i32 11, i32 13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32 27, i32 29, i32 31>
%ret = add <16 x i32> %odd, %even
store <16 x i32> %ret, <16 x i32>* %CPtr
ret void
}
define void @psubus_64i8_max_256(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-vector-width"="256" {
; CHECK-LABEL: psubus_64i8_max_256:
; CHECK:    # %bb.0:
; CHECK-NEXT: vmovdqa (%rdi), %ymm0
; CHECK-NEXT: vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT: vpsubusb (%rsi), %ymm0, %ymm0
; CHECK-NEXT: vpsubusb 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT: vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT: vmovdqa %ymm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%x = load <64 x i8>, <64 x i8>* %xptr
%y = load <64 x i8>, <64 x i8>* %yptr
%cmp = icmp ult <64 x i8> %x, %y
%max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
%res = sub <64 x i8> %max, %y
store <64 x i8> %res, <64 x i8>* %zptr
define void @psubus_64i8_max_512(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-vector-width"="512" {
    ; CHECK-LABEL: psubus_64i8_max_512:
    ; CHECK:       # %bb.0:
    ; CHECK-NEXT:    vmovdqa64 (%rdi), %zmm0
    ; CHECK-NEXT:    vpsubusb (%rsi), %zmm0, %zmm0
    ; CHECK-NEXT:    vmovdqa64 %zmm0, (%rdx)
    ; CHECK-NEXT:    vzeroupper
    ; CHECK-NEXT:    retq
    %x = load <64 x i8>, <64 x i8>* %xptr
    %y = load <64 x i8>, <64 x i8>* %yptr
    %cmp = icmp ult <64 x i8> %x, %y
    %max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
    %res = sub <64 x i8> %max, %y
    store <64 x i8> %res, <64 x i8>* %zptr
    ret void
}

define i32 @_Z9test_charPcS_i_256(i8* nocapture readonly, i8* nocapture readonly, i32) "min-legal-vector-width"="256" {
    ; CHECK-LABEL: _Z9test_charPcS_i_256:
    ; CHECK:       # %bb.0: # %entry
    ; CHECK-NEXT:    movl %edx, %eax
    ; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
    ; CHECK-NEXT:    xorl %ecx, %ecx
    ; CHECK-NEXT:    vpxor %xmm1, %xmm1, %xmm1
    ; CHECK-NEXT:    vpxor %xmm2, %xmm2, %xmm2
    ; .p2align 4, 0x90
    ; .LBB8_1: # %vector.body
    ; CHECK-NEXT:    # =>This Inner Loop Header: Depth=1
    ; CHECK-NEXT:    vpmovsxbw (%rdi,%rcx), %ymm3
    ; CHECK-NEXT:    vpmovsxbw 16(%rdi,%rcx), %ymm4
    ; CHECK-NEXT:    vpmovsxbw (%rsi,%rcx), %ymm5
    ; CHECK-NEXT:    vpmaddwd %ymm3, %ymm3, %ymm5, %ymm5
    ; CHECK-NEXT:    vpmaddwd %ymm1, %ymm1, %ymm1
    ; CHECK-NEXT:    vpmovsxbw 16(%rsi,%rcx), %ymm3
    ; CHECK-NEXT:    vpmaddwd %ymm2, %ymm2, %ymm2
    ; CHECK-NEXT:    addq $32, %rcx
    ; CHECK-NEXT:    cmpq %rcx, %rax
    ; CHECK-NEXT:    jne .LBB8_1
    ; CHECK-NEXT:    # %bb.2: # %middle.block
    ; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm2
    ; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm1
    ; CHECK-NEXT:    addq $32, %rcx
    ; CHECK-NEXT:    cmpq %rcx, %rax
    ; CHECK-NEXT:    jne .LBB8_1
    ; CHECK-NEXT:    # %bb.3: # %final.block
    ; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm1
; CHECK-NEXT:  vextracti128 $1, %ymm0, %xmm1
; CHECK-NEXT:  vpadd %ymm1, %ymm0, %ymm0
; CHECK-NEXT:  vpshufd {{.*#+}} %xmm1 = %xmm0[2,3,0,1]
; CHECK-NEXT:  vpadd %ymm1, %ymm0, %ymm0
; CHECK-NEXT:  vpshufd {{.*#+}} %xmm1 = %xmm0[1,1,2,3]
; CHECK-NEXT:  vmovd %xmm0, %eax
; CHECK-NEXT:  vzeroupper
entry:
  %3 = zext i32 %2 to i64
br label %vector.body

vector.body:
  %index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]
  %vec.phi = phi <32 x i32> [ %11, %vector.body ], [ zero initializer, %entry ]
  %4 = getelementptr inbounds i8, i8* %0, i64 %index
  %5 = bitcast i8* %4 to <32 x i8>*
  %wide.load = load <32 x i8>, <32 x i8>* %5, align 1
  %6 = sext <32 x i8> %wide.load to <32 x i32>
  %7 = getelementptr inbounds i8, i8* %1, i64 %index
  %8 = bitcast i8* %7 to <32 x i8>*
  %wide.load14 = load <32 x i8>, <32 x i8>* %8, align 1
  %9 = sext <32 x i8> %wide.load14 to <32 x i32>
  %10 = mul nsw <32 x i32> %9, %6
  %11 = add nsw <32 x i32> %10, %vec.phi
  %index.next = add i64 %index, 32
  %12 = icmp eq i64 %index.next, %3
br i1 %12, label %middle.block, label %vector.body

middle.block:
  %rdx.shuf1 = shufflevector <32 x i32> %11, <32 x i32> undef, <32 x i32> %i32 16, 132 17, i32 18, i32 19, i32 20, i32 21, i32 22, i32 23, i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 unde
define i32 @_Z9test_charPcS_i_512(i8* nocapture readonly, i8* nocapture readonly, i32) "min-legal-vector-width"="512" {
  ; CHECK-LABEL: _Z9test_charPcS_i_512:
  ; CHECK:       # %bb.0: # %entry
  ; CHECK-NEXT:    movl %edx, %eax
  ; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
  ; CHECK-NEXT:    xorl %ecx, %ecx
  ; CHECK-NEXT:    vpxor %xmm1, %xmm1, %xmm1
  ; CHECK-NEXT:    .p2align 4, 0x90
  ; CHECK-NEXT:  .LBB9_1: # %vector.body
  ; CHECK-NEXT: # =>This Inner Loop Header: Depth=1
  ; CHECK-NEXT:    vpmovsxbw (%rdi,%rcx), %zmm2
  ; CHECK-NEXT:    vpmovsxbw (%rsi,%rcx), %zmm3
  ; CHECK-NEXT:    vpmaddwd %zmm2, %zmm3, %zmm2
  ; CHECK-NEXT:    vpaddd %zmm1, %zmm2, %zmm1
  ; CHECK-NEXT:    addq $32, %rcx
  ; CHECK-NEXT:    cmpq %rcx, %rax
  ; CHECK-NEXT:    jne .LBB9_1
  ; CHECK-NEXT:  # %bb.2: # %middle.block
  ; CHECK-NEXT:    vpaddd %zmm0, %zmm1, %zmm0
  ; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm1
  ; CHECK-NEXT:    vpaddd %zmm0, %zmm1, %zmm0
  ; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
  ; CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
  ; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,0,1]
  ; CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
  ; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[1,1,2,3]
  ; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
  ; CHECK-NEXT:    vmovd %xmm0, %eax
  ; CHECK-NEXT:    vzeroupper
  ; CHECK-NEXT:    retq
  entry:
  %3 = zext i32 %2 to i64
  br label %vector.body

  vector.body:
@a = global [1024 x i8] zeroinitializer, align 16
@b = global [1024 x i8] zeroinitializer, align 16
define i32 @sad_16i8_256() "min-legal-vector-width"="256" {
    ; CHECK-LABEL: sad_16i8_256:
    ; CHECK: # %bb.0: # %entry
    ; CHECK-NEXT: vpxor %xmm0, %xmm0, %xmm0
    ; CHECK-NEXT: movq $-1024, %rax # imm = 0xFF00
    ; CHECK-NEXT: vpxor %xmm1, %xmm1, %xmm1
    ; CHECK-NEXT: .p2align 4, 0x90
    ; CHECK-NEXT: .LBB10_1: # %vector.body
    ; CHECK-NEXT: # =>This Inner Loop Header: Depth=1
    ; CHECK-NEXT: vmovdqu a+1024(%rax), %xmm2
    ; CHECK-NEXT: vpsadbw b+1024(%rax), %xmm2, %xmm2
    ; CHECK-NEXT: vpaddd %ymm1, %ymm2, %ymm1
    ; CHECK-NEXT: addq $4, %rax
    ; CHECK-NEXT: jne .LBB10_1
    ; CHECK-NEXT: # %bb.2: # %middle.block
    ; CHECK-NEXT: vpaddd %ymm0, %ymm1, %ymm0
    ; CHECK-NEXT: vextracti128 %ymm0, %ymm0, %ymm1
    ; CHECK-NEXT: vpaddd %ymm1, %ymm0, %ymm0
    ; CHECK-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[2,3,0,1]
    ; CHECK-NEXT: vpaddd %ymm1, %ymm0, %ymm0
    ; CHECK-NEXT: vpshufd {{.*#+}} xmm1 = xmm0[1,1,2,3]
    ; CHECK-NEXT: vpaddd %xmm1, %xmm0, %xmm0
    ; CHECK-NEXT: vmovd %xmm0, %eax
    ; CHECK-NEXT: vzeroupper
    ; CHECK-NEXT: retq

    entry:
    br label %vector.body
}

vector.body:
%index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
%vec.phi = phi <16 x i32> [ zeroinit, %entry ], [ %10, %vector.body ]
%0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
%1 = bitcast i8* %0 to <16 x i8>*
%wide.load = load <16 x i8>, <16 x i8>* %1, align 4
%2 = zext <16 x i8> %wide.load to <16 x i32>
%3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
%4 = bitcast i8* %3 to <16 x i8>*
%wide.load1 = load <16 x i8>, <16 x i8>* %4, align 4
%5 = zext <16 x i8> %wide.load1 to <16 x i32>
%6 = sub nsw <16 x i32> %2, %5
%7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
%8 = sub nsw <16 x i32> zeroinit, %6
%9 = select <16 x i1> %7, <16 x i32> %6, <16 x i32> %8
%10 = add nsw <16 x i32> %9, %vec.phi
%index.next = add i64 %index, 4
%11 = icmp eq i64 %index.next, 1024
begin %middle.block, label %vector.body

middle.block:
%.lcssa = phi <16 x i32> [* %10, %vector.body ]
%rdx.shuf = shufflevector <16 x i32> %.lcssa, <16 x i32> undef, <16 x i32> i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx = add <16 x i32> %.lcssa, %rdx.shuf
%rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
%rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
%rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
%bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
%12 = extractelement <16 x i32> %bin.rdx4, i32 0
ret i32 %12
}

define i32 @sad_16i8_512() "min-legal-vector-width"="512" {
; CHECK-LABEL: sad_16i8_512:
; CHECK: # %bb.0: # %entry
; CHECK-NEXT: vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT: movq $-1024, %rax # imm = 0xFC00
; CHECK-NEXT: .p2align 4, 0x90
; CHECK-NEXT: .LBB11_1: # %vector.body
; CHECK-NEXT: # =>This Inner Loop Header: Depth=1
; CHECK-NEXT: vmovdqu a+1024(%rax), %xmm1
; CHECK-NEXT: vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-NEXT: vpadd %zmm0, %zmm1, %zmm0
; CHECK-NEXT: addq $4, %rax
; CHECK-NEXT: jne .LBB11_1
; CHECK-NEXT: # %bb.2: # %middle.block
; CHECK-NEXT: vextracti64x4 %y $1, %zmm0, %ymm1
; CHECK-NEXT: vpadd %zmm1, %zmm0, %zmm0
; CHECK-NEXT: vextracti128 %y0 $1, %ymm0, %xmm1
; CHECK-NEXT: vpadd %zmm1, %zmm0, %zmm0
; CHECK-NEXT: vpsufd {{.*#+}} %xmm1 = %xmm0[2,3,0,1]
; CHECK-NEXT: vpadd %zmm1, %zmm0, %zmm0
; CHECK-NEXT: vpsufd {{.*#+}} %xmm1 = %xmm0[1,1,2,3]
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
entry:
  br label %vector.body

vector.body:
  %index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
  %vec.phi = phi <16 x i32> [ zeroinitialize, %entry ], [ %10, %vector.body ]
  %0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
  %1 = bitcast i8* %0 to <16 x i8>*
  %wide.load = load <16 x i8>, <16 x i8>* %1, align 4
  %2 = zext <16 x i8> %wide.load to <16 x i32>
  %3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
  %4 = bitcast i8* %3 to <16 x i8>*
  %wide.load1 = load <16 x i8>, <16 x i8>* %4, align 4
  %5 = zext <16 x i8> %wide.load1 to <16 x i32>
  %6 = sub nsw <16 x i32> %2, %5
  %7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
  %8 = sub nsw <16 x i32> zeroinitializer, %6
  %9 = select <16 x i1> %7, <16 x i32> %6, <16 x i32> %8
  %10 = add nsw <16 x i32> %9, %vec.phi
  %index.next = add i64 %index, 4
  br i1 %11, label %middle.block, label %vector.body

middle.block:
  %.lcssa = phi <16 x i32> [ %10, %vector.body ]
  %rdx.shuf = shufflevector <16 x i32> %.lcssa, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
  %bin.rdx = add <16 x i32> %.lcssa, %rdx.shuf
  %rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
  %bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
  %rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
  %bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
  %rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
  %bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
  %12 = extractelement <16 x i32> %bin.rdx4, i32 0
  ret i32 %12
}

define void @sbto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
  ; CHECK-LABEL: sbto16f32_256:
  ; CHECK:   # %bb.0:

define void @sbto16f32_512(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="512" {
    ; CHECK-LABEL: sbto16f32_512:
    ; CHECK:       # %bb.0:
    ; CHECK-NEXT:    vpmovw2m %ymm0, %k0
    ; CHECK-NEXT:    vpmovm2d %k0, %zmm0
    ; CHECK-NEXT:    vcvtdq2ps %zmm0, %zmm0
    ; CHECK-NEXT:    vmovaps %zmm0, (%rdi)
    ; CHECK-NEXT:    vzeroupper
    ; CHECK-NEXT:    retq
    %mask = icmp slt <16 x i16> %a, zeroinitializer
    %1 = sitofp <16 x i1> %mask to <16 x float>
    store <16 x float> %1, <16 x float>* %res
    ret void
}

define void @sbto16f64_256(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="256" {
    ; CHECK-LABEL: sbto16f64_256:
    ; CHECK:       # %bb.0:
    ; CHECK-NEXT:    vpmovw2m %ymm0, %k0
    ; CHECK-NEXT:    kshiftrw $8, %k0, %k1
    ; CHECK-NEXT:    vpmovm2d %k1, %ymm0
    ; CHECK-NEXT:    vcvtdq2ps %ymm0, %ymm0
    ; CHECK-NEXT:    vpmovm2d %k0, %ymm1
    ; CHECK-NEXT:    vcvtdq2ps %ymm1, %ymm1
    ; CHECK-NEXT:    vmovaps %ymm1, (%rdi)
    ; CHECK-NEXT:    vmovaps %ymm0, 32(%rdi)
    ; CHECK-NEXT:    vzeroupper
    ; CHECK-NEXT:    retq
    %mask = icmp slt <16 x i16> %a, zeroinitializer
    %1 = sitofp <16 x i1> %mask to <16 x float>
    store <16 x float> %1, <16 x float>* %res
    ret void
}
define void @sbtto16f64_512(<16 x i16> %a, <16 x double>* %res) { "min-legal-vector-width"="512" { 
; CHECK-LABEL: sbtto16f64_512:
; CHECK: # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: vpmovm2d %k0, %zmm0
; CHECK-NEXT: vcvtqd2pd %ymm0, %zmm1
; CHECK-NEXT: vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT: vcvtqd2pd %ymm0, %zmm0
; CHECK-NEXT: vmovaps %zmm0, 64(%rdi)
; CHECK-NEXT: vmovaps %zmm1, (%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = sitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

define void @ubtto16f32_512(<16 x i16> %a, <16 x float>* %res) { "min-legal-vector-width"="512" { 
; CHECK-LABEL: ubtto16f32_512:
; CHECK: # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: kshiftrw $8, %k0, %k1
; CHECK-NEXT: vpmovm2d %k1, %ymm0
; CHECK-NEXT: vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT: vcvtqd2ps %ymm0, %ymm0
; CHECK-NEXT: vpmovm2d %k0, %ymm0
; CHECK-NEXT: vpmovm2d %k0, %ymm1
; CHECK-NEXT: vpsrld $31, %ymm1, %ymm1
; CHECK-NEXT: vcvtqd2ps %ymm1, %ymm1
; CHECK-NEXT: vmovaps %ymm1, (%rdi)
; CHECK-NEXT: vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

define void @ubtto16f32_256(<16 x i16> %a, <16 x float>* %res) { "min-legal-vector-width"="256" { 
; CHECK-LABEL: ubtto16f32_256:
; CHECK: # %bb.0:
; CHECK-NEXT: vpmovw2m %ymm0, %k0
; CHECK-NEXT: kshiftrw $8, %k0, %k1
; CHECK-NEXT: vpmovm2d %k1, %ymm0
; CHECK-NEXT: vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT: vcvtqd2ps %ymm0, %ymm0
; CHECK-NEXT: vpmovm2d %k0, %ymm0
; CHECK-NEXT: vpmovm2d %k0, %ymm1
; CHECK-NEXT: vpsrld $31, %ymm1, %ymm1
; CHECK-NEXT: vcvtqd2ps %ymm1, %ymm1
; CHECK-NEXT: vmovaps %ymm1, (%rdi)
; CHECK-NEXT: vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
%mask = icmp slt <16 x i16> %a, zeroinitializer
%1 = uitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

define void @ubtto16f32_512(<16 x i16> %a, <16 x float>* %res) { "min-legal-vector-width"="512" {
define void @ubto16f32_512(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="512" [ 
; CHECK-LABEL: ubto16f32_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:  vpmovw2m %ymm0, %k0
; CHECK-NEXT:  vpmovm2d %k0, %zmm0
; CHECK-NEXT:  vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT:  vcvtdq2ps %zmm0, %zmm0
; CHECK-NEXT:  vmovaps %zmm0, (%rdi)
; CHECK-NEXT:  vzeroupper
; CHECK-NEXT:  retq
%mask = icmp slt <16 x i16> %a, zeroinitialize
%1 = uitofp <16 x i1> %mask to <16 x float>
store <16 x float> %1, <16 x float>* %res
ret void
}

define void @ubto16f64_256(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="256" [ 
; CHECK-LABEL: ubto16f64_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:  vpmovw2m %ymm0, %k0
; CHECK-NEXT:  kshiftrw $8, %k0, %k1
; CHECK-NEXT:  vpmovm2d %k1, %ymm0
; CHECK-NEXT:  vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT:  vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT:  vextracti128 $1, %ymm0, %xmm0
; CHECK-NEXT:  vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT:  vpmovm2d %k0, %ymm2
; CHECK-NEXT:  vpsrld $31, %ymm2, %ymm2
; CHECK-NEXT:  vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT:  vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT:  vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT:  vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT:  vmovaps %ymm3, %ymm3
; CHECK-NEXT:  vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT:  vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT:  vzeroupper
; CHECK-NEXT:  retq
%mask = icmp slt <16 x i16> %a, zeroinitialize
%1 = uitofp <16 x i1> %mask to <16 x double>
store <16 x double> %1, <16 x double>* %res
ret void
}

define void @ubto16f64_512(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="512" [ 
; CHECK-LABEL: ubto16f64_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:  vpmovw2m %ymm0, %k0
; CHECK-NEXT:  vpmovm2d %k0, %zmm0
; CHECK-NEXT:  vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT:  vzeroupper
}
define <16 x i16> @test_16f32toub_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-width"="256" {
; CHECK-LABEL: test_16f32toub_256:
; CHECK:
    # %bb.0:
    ; CHECK-NEXT:  vcvtps2dq (%rdi), %ymm1
; CHECK-NEXT:  vpslld $31, %ymm1, %ymm1
; CHECK-NEXT:  vpmovd2m %ymm1, %k1
; CHECK-NEXT:  vcvtps2dq 32(%rdi), %ymm1
; CHECK-NEXT:  vpslld $31, %ymm1, %ymm1
; CHECK-NEXT:  vpmovd2m %ymm1, %k1
; CHECK-NEXT:  kunpckbw %k0, %k1, %k1
; CHECK-NEXT:  vmovdq16 %ymm0, %ymm0 [%k1] {z}
    ; CHECK-NEXT:  retq
    %a = load <16 x float>, <16 x float>* %ptr
    %mask = fptoui <16 x float> %a to <16 x i1>
    %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
    ret <16 x i16> %select
}

define <16 x i16> @test_16f32tohsb_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-width"="512" {
; CHECK-LABEL: test_16f32tohsb_512:
; CHECK:
    # %bb.0:
    ; CHECK-NEXT:  vcvtps2dq (%rdi), %zmm1
; CHECK-NEXT:  vpslld $31, %zmm1, %zmm1
; CHECK-NEXT:  vpmovd2m %zmm1, %k1
; CHECK-NEXT:  vmovdq16 %ymm0, %ymm0 [%k1] {z}
    ; CHECK-NEXT:  retq
    %a = load <16 x float>, <16 x float>* %ptr
    %mask = fptoui <16 x float> %a to <16 x i1>
    %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
    ret <16 x i16> %select
}
define <16 x i16> @test_16f32tosb_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-width"="512" { 
  ; CHECK-LABEL: test_16f32tosb_512:
  ; CHECK:       # %bb.0:
  ; CHECK-NEXT:    vcvttps2dq (%rdi), %zmm1
  ; CHECK-NEXT:    vpmovd2m %zmm1, %k1
  ; CHECK-NEXT:    vvcvttps2dq 32(%rdi), %zmm1
  ; CHECK-NEXT:    vpmovd2m %zmm1, %k1
  ; CHECK-NEXT:    kunpckbw %k0, %k1, %k1
  ; CHECK-NEXT:    vmmovdqu16 %zmm0, %zmm0 {%k1} {z}
  ; CHECK-NEXT:    retq
  %a = load <16 x float>, <16 x float>* %ptr
  %mask = fptosi <16 x float> %a to <16 x i1>
  %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
  ret <16 x i16> %select
}

define void @mul256(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="256" { 
  ; CHECK-LABEL: mul256:
  ; CHECK:       # %bb.0:
  ; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
  ; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm3
  ; CHECK-NEXT:    vpunpckhbw {{.*#+}} ymm4 = ymm2[8],ymm0[8],ymm2[9],ymm0[9],ymm2[10],ymm0[10],ymm2[11],ymm0[11],ymm2[12],ymm0[12],ymm2[13],ymm0[13],ymm2[14],ymm0[14],ymm2[15],ymm0[15],ymm2[24],ymm0[24],ymm2[25],ymm0[25],ymm2[26],ymm0[26],ymm2[27],ymm0[27],ymm2[28],ymm0[28],ymm2[29],ymm0[29],ymm2[30],ymm0[30],ymm2[31],ymm0[31]
  ; CHECK-NEXT:    vpunpckhbw {{.*#+}} ymm5 = ymm0[8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
  ; CHECK-NEXT:    vpmullw %ymm4, %ymm5, %ymm4
  ; CHECK-NEXT:    vpmullw %ymm4, %ymm5, %ymm4

; CHECK-NEXT:  vpunpcklbw {.*#} ymm2 = ymm2[0], ymm0[0], ymm2[1], ymm0[1], ymm2[2], ymm0[2], ymm2[3], ymm0[3], ymm2[4], ymm0[4], ymm2[5], ymm0[5], ymm2[6], ymm0[6], ymm2[7], ymm0[7], ymm2[16], ymm0[16], ymm2[17], ymm0[17], ymm2[18], ymm0[18], ymm2[19], ymm0[19], ymm2[20], ymm0[20], ymm2[21], ymm0[21], ymm2[22], ymm0[22], ymm2[23], ymm0[23]
; CHECK-NEXT:  vpunpcklbw {.*#} ymm3 = ymm3[0], ymm0[0], ymm3[1], ymm0[1], ymm3[2], ymm0[2], ymm3[3], ymm0[3], ymm3[4], ymm0[4], ymm3[5], ymm0[5], ymm3[6], ymm0[6], ymm3[7], ymm0[7], ymm3[16], ymm0[16], ymm3[17], ymm0[17], ymm3[18], ymm0[18], ymm3[19], ymm0[19], ymm3[20], ymm0[20], ymm3[21], ymm0[21], ymm3[22], ymm0[22], ymm3[23], ymm0[23]
; CHECK-NEXT:  vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:  vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:  vzeroupper
; CHECK-NEXT:  retq

%d = load <64 x i8>, <64 x i8>*, %a
%e = load <64 x i8>, <64 x i8>*, %b
%f = mul <64 x i8> %d, %e
store <64 x i8> %f, <64 x i8>*, %c
ret void
}

define void @mul512(<64 x i8>*, %a, <64 x i8>*, %b, <64 x i8>*, %c) "min-legal-vector-width"="512" {
; CHECK-LABEL: mul512:
; CHECK:  # %bb.0:
; CHECK-NEXT:  vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:  vmovdqa64 (%rsi), %zmm1
; CHECK-NEXT:  vpunpckhbw {{.*#+}} zmm2 =
zmm1[8],zmm0[8],zmm1[9],zmm0[9],zmm1[10],zmm0[10],zmm1[11],zmm0[11],zmm1[12],zmm0[12],zmm1[13],
zmm0[13],zmm1[14],zmm0[14],zmm1[15],zmm0[15],zmm1[24],zmm0[24],zmm1[25],zmm0[25],zmm1[26],zmm0[26],
zmm1[27],zmm0[27],zmm1[28],zmm0[28],zmm1[29],zmm0[29],zmm1[30],zmm0[30],zmm1[31],zmm0[31],zmm1[40],
zmm0[40],zmm1[41],zmm0[41],zmm1[42],zmm0[42],zmm1[43],zmm0[43],zmm1[44],zmm0[44],zmm1[45],
zmm0[45],zmm1[46],zmm0[46],zmm1[47],zmm0[47],zmm1[56],zmm0[56],zmm1[57],zmm0[57],zmm1[58],zmm0[58],
zmm1[59],zmm0[59],zmm1[60],zmm0[60],zmm1[61],zmm0[61],zmm1[62],zmm0[62],zmm1[63],zmm0[63]

; CHECK-NEXT:  vpunpckhbw {{.*#+}} zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4

; CHECK-NEXT:  vpmullw %zmm2, %zmm3, %zmm2

; CHECK-NEXT:  vmovdqa64 {%zmm3} =

; CHECK-NEXT:  vpandq %zmm3, %zmm2, %zmm2

; CHECK-NEXT:  vpunpcklbw {{.*#+}} zmm1 =
zmm1[0],zmm0[0],zmm1[1],zmm0[1],zmm1[2],zmm0[2],zmm1[3],zmm0[3],zmm1[4],zmm0[4],zmm1[5],zmm0[5],
zmm1[6],zmm0[6],zmm1[7],zmm0[7],zmm1[16],zmm0[16],zmm1[17],zmm0[17],zmm1[18],zmm0[18],zmm1[19],
zmm0[19],zmm1[20],zmm0[20],zmm1[21],zmm0[21],zmm1[22],zmm0[22],zmm1[23],zmm0[23],zmm1[32],zmm0[32],
zmm1[33],zmm0[33],zmm1[34],zmm0[34],zmm1[35],zmm0[35],zmm1[36],zmm0[36],zmm1[37],zmm0[37],
zmm1[38],zmm0[38],zmm1[39],zmm0[39],zmm1[48],zmm0[48],zmm1[49],zmm0[49],zmm1[50],zmm0[50],zmm1[51],zmm0[51],zmm1[52],zmm0[52],zmm1[53],zmm0[53],zmm1[54],zmm0[54],zmm1[55],zmm0[55]

; CHECK-NEXT:  vpunpcklbw {{.*#+}} zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55,55

; CHECK-NEXT:  vpmullw %zmm0, %zmm1, %zmm0

; CHECK-NEXT:  vpandq %zmm0, %zmm0, %zmm0

; CHECK-NEXT:  vpackuswb %zmm2, %zmm0, %zmm0

; CHECK-NEXT:  vmovdqa64 {%zmm0}, (%rdx)

; CHECK-NEXT:  vzeroupper

; CHECK-NEXT:  retq

%d = load <64 x i8>, <64 x i8>* %a
%e = load <64 x i8>, <64 x i8>* %b
%f = mul <64 x i8> %d, %e
store <64 x i8> %f, <64 x i8>* %c
ret void
}

Copyright (c) 2006 Kirill Simonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====================
LLVM Release License
====================
University of Illinois/NCSA
Open Source License

Copyright (c) 2007-2018 University of Illinois at Urbana-Champaign.
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.
The LLVM software contains code written by third parties. Such software will have its own individual LICENSE.TXT file in the directory in which it appears. This file will describe the copyrights, license, and restrictions which apply to that code.

The disclaimer of warranty in the University of Illinois Open Source License applies to all code in the LLVM Distribution, and nothing in any of the other licenses gives permission to use the names of the LLVM Team or the University of Illinois to endorse or promote products derived from this Software.

The following pieces of software have additional or alternate copyrights, licenses, and/or restrictions:

<table>
<thead>
<tr>
<th>Program</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;none yet&gt;</td>
<td>&lt;none yet&gt;</td>
</tr>
</tbody>
</table>

; NOTE: Assertions have been autogenerated by utils/update_analyze_test_checks.py
; RUN: opt < %s -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -mattr=+avx2 | FileCheck %s --check-prefixes=CHECK,VEC256,AVX
; RUN: opt < %s -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -mattr=+avx512vl,+avx512bw,+avx512dq,+prefer-256-bit | FileCheck %s --check-prefixes=CHECK,VEC256,SKX256
; RUN: opt < %s -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -mattr=+avx512vl,+avx512bw,+avx512dq,-prefer-256-bit | FileCheck %s --check-prefixes=CHECK,VEC512

define void @zext256() "min-legal-vector-width"="256" {
  ; VEC256-LABEL: 'zext256'
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %A = zext <8 x i16> undef to <8 x i64>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to <8 x i64>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = zext <16 x i16> undef to <16 x i32>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x i16>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
  
  ; VEC512-LABEL: 'zext256'
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x i32>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x i16>
i16>
; VEC512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = zext <8 x i16> undef to <8 x i64>
%B = zext <8 x i32> undef to <8 x i64>
%C = zext <16 x i8> undef to <16 x i32>
%D = zext <16 x i16> undef to <16 x i32>
%E = zext <32 x i8> undef to <32 x i16>
ret void
}

define void @zext512() "min-legal-vector-width"="512" {
 ; AVX-LABEL: 'zext512'
 ; AVX-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %A = zext <8 x i16> undef to <8 x i64>
 ; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to <8 x i64>
 ; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
 ; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = zext <16 x i16> undef to <16 x i32>
 ; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x i16>
 ; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
 ;
 ; SKX256-LABEL: 'zext512'
 ; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
 ; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
 ; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x i32>
 ; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
 ; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x i16>
 ; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
 ;
 ; VEC512-LABEL: 'zext512'
 ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
 ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
 ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x i32>
 ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
 ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x i16>
 ; VEC512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
 ;
%A = zext <8 x i16> undef to <8 x i64>
define void @sext256() "min-legal-vector-width"="256" {
  ; VEC256-LABEL: 'sext256'
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %A = sext <8 x i8> undef to <8 x i64>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %B = sext <8 x i16> undef to <8 x i64>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x i64>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %E = sext <16 x i16> undef to <16 x i32>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
  ; VEC256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void

  ; VEC512-LABEL: 'sext256'
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 24 for instruction: %A = sext <8 x i8> undef to <8 x i64>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>
  ; VEC512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void

  ; AVX-LABEL: 'sext512'
  ; AVX-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %A = sext <8 x i8> undef to <8 x i64>
  ; AVX-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %B = sext <8 x i16> undef to <8 x i64>
  ; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x i64>
  ; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
  ; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %E = sext <16 x i16> undef to <16 x i32>
  ; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
  ; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void

define void @sext512() "min-legal-vector-width"="512" {
  ; AVX-LABEL: 'sext512'}
; AVX-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'sext512'
; SKX256-NEXT: Cost Model: Found an estimated cost of 24 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; VEC512-LABEL: 'sext512'
; VEC512-NEXT: Cost Model: Found an estimated cost of 24 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; VEC512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
%A = sext <8 x i8> undef to <8 x i64>
%B = sext <8 x i16> undef to <8 x i64>
%C = sext <8 x i32> undef to <8 x i64>
%D = sext <16 x i8> undef to <16 x i32>
%E = sext <16 x i16> undef to <16 x i32>
%F = sext <32 x i8> undef to <32 x i16>
ret void
} $OpenBSD: COPYRIGHT,v 1.3 2003/06/02 20:18:36 millert Exp $
Copyright 1992, 1993, 1994 Henry Spencer. All rights reserved.
This software is not subject to any license of the American Telephone
and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on
any computer system, and to alter it and redistribute it, subject
to the following restrictions:

1. The author is not responsible for the consequences of use of this
software, no matter how awful, even if they arise from flaws in it.

2. The origin of this software must not be misrepresented, either by
explicit claim or by omission. Since few users ever read sources,
credits must appear in the documentation.

3. Altered versions must be plainly marked as such, and must not be
 misrepresented as being the original software. Since few users
ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
/*
 * Copyright (c) 1994
 * The Regents of the University of California. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)COPYRIGHT8.1 (Berkeley) 3/16/94
*/

; RUN: llc -march=hexagon < %s
; REQUIRES: asserts

; The two loads based on %struct.0, loading two different data types
; cause LSR to assume type "void" for the memory type. This would then
; cause an assert in isLegalAddressingMode. Make sure we no longer crash.

target triple = "hexagon"

%struct.0 = type { i8*, i8, %union.anon.0 }
%union.anon.0 = type { i8* }

define hidden fastcc void @fred() unnamed_addr #0 {
  entry:
  br i1 undef, label %while.end, label %while.body.lr.ph

  while.body.lr.ph:  ; preds = %entry
    br label %while.body

  while.body:  ; preds = %exit.2, %while.body.lr.ph
    %lsr.iv = phi %struct.0* [ %cgep22, %exit.2 ], [ undef, %while.body.lr.ph ]
    switch i32 undef, label %exit [ i32 1, label %sw.bb.i 
                              i32 2, label %sw.bb3.i
                          ]

  sw.bb.i:  ; preds = %while.body
    unreachable

  sw.bb3.i:  ; preds = %while.body
    unreachable

  exit:  ; preds = %while.body
    switch i32 undef, label %exit.2 [ i32 1, label %sw.bb.i17
                                       i32 2, label %sw.bb3.i20
                          ]

  sw.bb.i17:  ; preds = %exit
    %0 = bitcast %struct.0* %lsr.iv to i32*
    %1 = load i32, i32* %0, align 4
    unreachable

  sw.bb3.i20:  ; preds = %exit

%2 = bitcast %struct.0* %lsr.iv to i8**
%3 = load i8*, i8** %2, align 4
unreachable

exit.2: ; preds = %exit
%cegp22 = getelementptr %struct.0, %struct.0* %lsr.iv, i32 1
br label %while.body

while.end: ; preds = %entry
ret void
}

attributes #0 = { nounwind optsize "target-cpu"="hexagonv55" }
; RUN: llc -mtriple=aarch64-apple-ios %s -o - | FileCheck %s

define <16 x double> @test_sitofp_fixed(<16 x i32> %in) {
; CHECK-LABEL: test_sitofp_fixed:
; First, extend each i32 to i64
; CHECK-DAG: sshll2.2d [[BLOCK0_HI:v[0-9]+]], v0, #0
; CHECK-DAG: sshll2.2d [[BLOCK1_HI:v[0-9]+]], v1, #0
; CHECK-DAG: sshll2.2d [[BLOCK2_HI:v[0-9]+]], v2, #0
; CHECK-DAG: sshll2.2d [[BLOCK3_HI:v[0-9]+]], v3, #0
; CHECK-DAG: sshll.2d [[BLOCK0_LO:v[0-9]+]], v0, #0
; CHECK-DAG: sshll.2d [[BLOCK1_LO:v[0-9]+]], v1, #0
; CHECK-DAG: sshll.2d [[BLOCK2_LO:v[0-9]+]], v2, #0
; CHECK-DAG: sshll.2d [[BLOCK3_LO:v[0-9]+]], v3, #0
; Next, convert each to double.
; CHECK-DAG: scvtf.2d v0, [[BLOCK0_LO]]
; CHECK-DAG: scvtf.2d v1, [[BLOCK0_HI]]
; CHECK-DAG: scvtf.2d v2, [[BLOCK1_LO]]
; CHECK-DAG: scvtf.2d v3, [[BLOCK1_HI]]
; CHECK-DAG: scvtf.2d v4, [[BLOCK2_LO]]
; CHECK-DAG: scvtf.2d v5, [[BLOCK2_HI]]
; CHECK-DAG: scvtf.2d v6, [[BLOCK3_LO]]
; CHECK-DAG: scvtf.2d v7, [[BLOCK3_HI]]
; CHECK: ret
%flt = sitofp <16 x i32> %in to <16 x double>
%res = fdiv <16 x double> %flt, <double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0>
ret <16 x double> %res
}

; This one is small enough to satisfy isSimple, but still illegally large.
define <4 x double> @test_sitofp_fixed_shortish(<4 x i64> %in) {
; CHECK-LABEL: test_sitofp_fixed_shortish:

; CHECK-DAG: scvtf.2d v0, v0
; CHECK-DAG: scvtf.2d v1, v1

; CHECK: ret
%flt = sitofp <4 x i64> %in to <4 x double>
%res = fdiv <4 x double> %flt, <double 64.0, double 64.0, double 64.0, double 64.0>
ret <4 x double> %res
}
; RUN: opt %s -inline -S | FileCheck %s

define internal void @innerSmall() "min-legal-vector-width"="128" {
  ret void
}

define internal void @innerLarge() "min-legal-vector-width"="512" {
  ret void
}

define internal void @innerNoAttribute() {
  ret void
}

; We should not add an attribute during inlining. No attribute means unknown.
; Inlining doesn't change the fact that we don't know anything about this
; function.
define void @outerNoAttribute() {
call void @innerLarge()
ret void
}

define void @outerConflictingAttributeSmall() "min-legal-vector-width"="128" {
call void @innerLarge()
ret void
}

define void @outerConflictingAttributeLarge() "min-legal-vector-width"="512" {
call void @innerSmall()
ret void
}

; We should remove the attribute after inlining since the callee's
; vector width requirements are unknown.
define void @outerAttribute() "min-legal-vector-width"="128" {
call void @innerNoAttribute()
ret void
}
; CHECK: define void @outerNoAttribute() {
; CHECK: define void @outerConflictingAttributeSmall() #0
; CHECK: define void @outerConflictingAttributeLarge() #0
; CHECK: define void @outerAttribute() {
; CHECK: attributes #0 = { "min-legal-vector-width"="512" }
; RUN: opt -consthoist -S -o - %s | FileCheck %s
target triple = "thumbv6m-none--musleabi"

; Check that for i8 type, the maximum legal offset is 31.
; Also check that an constant used as value to be stored rather than
; pointer in a store instruction is hoisted.
; CHECK: foo_i8
; CHECK-DAG:  %[[C1:const[0-9]?]] = bitcast i32 805874720 to i32
; CHECK-DAG:  %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK-DAG:  %[[C3:const[0-9]?]] = bitcast i32 805873720 to i32
; CHECK-DAG:  %[[C4:const[0-9]?]] = bitcast i32 805873688 to i32
; CHECK:  %0 = inttoptr i32 %[[C2]] to i8*
; CHECK-NEXT:  %1 = load volatile i8, i8* %0
; CHECK-NEXT:  %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT:  %2 = inttoptr i32 %[[M1]] to i8*
; CHECK-NEXT:  %3 = load volatile i8, i8* %2
; CHECK-NEXT:  %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 31
; CHECK-NEXT:  %4 = inttoptr i32 %[[M2]] to i8*
; CHECK-NEXT:  %5 = load volatile i8, i8* %4
; CHECK-NEXT:  %6 = inttoptr i32 %[[C1]] to i8*
; CHECK-NEXT:  %7 = load volatile i8, i8* %6
; CHECK-NEXT:  %10 = inttoptr i32 %[[C4]] to i8*
; CHECK-NEXT:  store i8 %9, i8* %10
; CHECK-NEXT:  %[[M4:const_mat[0-9]?]] = add i32 %[[C4]], 31
; CHECK-NEXT:  %11 = inttoptr i32 %[[M4]] to i8*
; CHECK-NEXT:  store i8 %7, i8* %11
; CHECK-NEXT:  %12 = inttoptr i32 %[[C3]] to i8*
; CHECK-NEXT:  store i8 %5, i8* %12
; CHECK-NEXT:  %[[M5:const_mat[0-9]?]] = add i32 %[[C3]], 7
; CHECK-NEXT:  %13 = inttoptr i32 %[[M5]] to i8*
; CHECK-NEXT:  store i8 %3, i8* %13
; CHECK-NEXT:  %[[M6:const_mat[0-9]?]] = add i32 %[[C1]], 80
; CHECK-NEXT:  %14 = inttoptr i32 %[[M6]] to i8*
; CHECK-NEXT:  store i8* %14, i8** @goo

@goo = global i8* undef

define void @foo_i8() {

entry:
%0 = load volatile i8, i8* inttoptr (i32 805874688 to i8*)
%1 = load volatile i8, i8* inttoptr (i32 805874692 to i8*)
%2 = load volatile i8, i8* inttoptr (i32 805874719 to i8*)
%3 = load volatile i8, i8* inttoptr (i32 805874720 to i8*)
%4 = load volatile i8, i8* inttoptr (i32 805874727 to i8*)
store i8 %4, i8* inttoptr(i32 805873688 to i8*)
store i8 %3, i8* inttoptr(i32 805873719 to i8*)
store i8 %2, i8* inttoptr(i32 805873720 to i8*)
store i8 %1, i8* inttoptr(i32 805873727 to i8*)
store i8* inttoptr(i32 805874800 to i8*), i8** @goo
ret void
}

; Check that for i16 type, the maximum legal offset is 62.
; CHECK: foo_i16
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874752 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to i16*
; CHECK-NEXT: %1 = load volatile i16, i16* %0, align 2
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to i16*
; CHECK-NEXT: %3 = load volatile i16, i16* %2, align 2
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 32
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to i16*
; CHECK-NEXT: %5 = load volatile i16, i16* %4, align 2
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C2]], 62
; CHECK-NEXT: %6 = inttoptr i32 %[[M3]] to i16*
; CHECK-NEXT: %7 = load volatile i16, i16* %6, align 2
; CHECK-NEXT: %8 = inttoptr i32 %[[C1]] to i16*
; CHECK-NEXT: %9 = load volatile i16, i16* %8, align 2
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C1]], 22
; CHECK-NEXT: %10 = inttoptr i32 %[[M4]] to i16*
; CHECK-NEXT: %11 = load volatile i16, i16* %10, align 2

define void @foo_i16() {
entry:
%0 = load volatile i16, i16* inttoptr (i32 805874688 to i16*), align 2
%1 = load volatile i16, i16* inttoptr (i32 805874692 to i16*), align 2
%2 = load volatile i16, i16* inttoptr (i32 805874720 to i16*), align 2
%3 = load volatile i16, i16* inttoptr (i32 805874750 to i16*), align 2
%4 = load volatile i16, i16* inttoptr (i32 805874752 to i16*), align 2
%5 = load volatile i16, i16* inttoptr (i32 805874774 to i16*), align 2
ret void
}

; Check that for i32 type, the maximum legal offset is 124.
; CHECK: foo_i32

Open Source Used In DNAC Platform Antelope 10128
define void @foo_i32() {
entry:
%0 = load volatile i32, i32* inttoptr (i32 805874688 to i32*), align 4
%1 = load volatile i32, i32* inttoptr (i32 805874692 to i32*), align 4
%2 = load volatile i32, i32* inttoptr (i32 805874812 to i32*), align 4
%3 = load volatile i32, i32* inttoptr (i32 805874816 to i32*), align 4
%4 = load volatile i32, i32* inttoptr (i32 805874824 to i32*), align 4
%5 = load volatile i32, i32* inttoptr (i32 805874828 to i32*), align 4
ret void
}

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
# https://developers.google.com/open-source/clia/individual
# https://developers.google.com/open-source/clia/corporate
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate CLA was used.
#
# Names should be added to this file as:
#     Name <email address>
#
# Please keep the list sorted.

Albert Pretorius <pretoalb@gmail.com>
Arne Beer <arne@twobeer.de>
Billy Robert O'Neal III <billy.oneal@gmail.com> <bion@microsoft.com>
Chris Kennelly <ckennelly@google.com> <ckennelly@ckennelly.com>
Christopher Seymour <chris.j.seymour@hotmail.com>
David Coeurjolly <david.coeurjolly@liris.cnrs.fr>
Deniz Evrenci <denizevreneci@gmail.com>
Dominic Hamon <dma@stripysock.com> <dominic@google.com>
Dominik Czarnota <dominik.b.czarnota@gmail.com>
Eric Fiselier <eric@efcs.ca>
Eugene Zhuk <eugene.zhuk@gmail.com>
Evgeny Safronov <division494@gmail.com>
Felix Homann <linuxaudio@showlabor.de>
Ismael Jimenez Martinez <ismael.jimenez.martinez@gmail.com>
Jern-Kuan Leong <jernkuan@gmail.com>
LLVM Release License

University of Illinois/NCSA
Open Source License

Copyright (c) 2003-2019 University of Illinois at Urbana-Champaign.
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==============================================================================
Copyrights and Licenses for Third Party Software Distributed with LLVM:
==============================================================================
The LLVM software contains code written by third parties. Such software will have its own individual LICENSE.TXT file in the directory in which it appears. This file will describe the copyrights, license, and restrictions which apply to that code.

The disclaimer of warranty in the University of Illinois Open Source License applies to all code in the LLVM Distribution, and nothing in any of the other licenses gives permission to use the names of the LLVM Team or the University of Illinois to endorse or promote products derived from this Software.

The following pieces of software have additional or alternate copyrights, licenses, and/or restrictions:

<table>
<thead>
<tr>
<th>Program</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Test</td>
<td>llvm/utils/unittest/googletest</td>
</tr>
<tr>
<td>OpenBSD regex</td>
<td>llvm/lib/Support/[reg*, COPYRIGHT.regex]</td>
</tr>
<tr>
<td>pyyaml tests</td>
<td>llvm/test/YAMLParser/{*.data, LICENSE.TXT}</td>
</tr>
<tr>
<td>ARM contributions</td>
<td>llvm/lib/Target/ARM/LICENSE.TXT</td>
</tr>
<tr>
<td>md5 contributions</td>
<td>llvm/lib/Support/MD5.cpp llvm/include/llvm/Support/MD5.h</td>
</tr>
<tr>
<td>LLVM System Interface Library</td>
<td></td>
</tr>
</tbody>
</table>
The LLVM System Interface Library is licensed under the Illinois Open Source License and has the following additional copyright:

Copyright (C) 2004 eXtensible Systems, Inc.

; RUN: llc -mtriple=x86_64-linux-android -mattr=+mmx -enable-legalize-types-checking | FileCheck %s
; D31946
; Check that we dont end up with the "LLVM ERROR: Cannot select" error.
; Additionally ensure that the output code actually put fp128 values in SSE registers.

declare fp128 @llvm.fabs.f128(fp128)
declare fp128 @llvm.copysign.f128(fp128, fp128)

define fp128 @TestSelect(fp128 %a, fp128 %b) {
  %cmp = fcmp ogt fp128 %a, %b
  %sub = fsub fp128 %a, %b
  %res = select i1 %cmp, fp128 %sub, fp128 0xL00000000000000000000000000000000
  ret fp128 %res
}

define fp128 @TestFabs(fp128 %a) {
  %res = call fp128 @llvm.fabs.f128(fp128 %a)
  ret fp128 %res
}

define fp128 @TestCopysign(fp128 %a, fp128 %b) {
  %res = call fp128 @llvm.copysign.f128(fp128 %a, fp128 %b)
  ret fp128 %res
}

define fp128 @TestFabs(fp128 %a) {
  %res = call fp128 @llvm.fabs.f128(fp128 %a)
  ret fp128 %res
}

define fp128 @TestCopysign(fp128 %a, fp128 %b) {
  %res = call fp128 @llvm.copysign.f128(fp128 %a, fp128 %b)
  ret fp128 %res
}
define fp128 @TestFneg(fp128 %a) {
    %mul = fmul fp128 %a, %a
    %res = fsub fp128 0xL00000000000000008000000000000000, %mul
    ret fp128 %res
}

Software Grant License Agreement ("Agreement")

Except for the license granted herein to you, ARM Limited ("ARM") reserves all right, title, and interest in and to the Software (defined below).

Definition

"Software" means the code and documentation as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by ARM to llvm.org (http://llvm.org) ("LLVM") for inclusion in, or documentation of, any of the products owned or managed by LLVM (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to LLVM or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, LLVM for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked otherwise.

1. Grant of Copyright License. Subject to the terms and conditions of this Agreement, ARM hereby grants to you and to recipients of the Software distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute the Software and such derivative works.

2. Grant of Patent License. Subject to the terms and conditions of this Agreement, ARM hereby grants you and to recipients of the Software distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by ARM that are necessarily infringed by ARM's Software alone or by combination of the Software with the Work to which such Software was submitted. If any entity institutes patent litigation against ARM or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging
that ARM’s Software, or the Work to which ARM has contributed constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for the Software or Work shall terminate as of the date such litigation is filed.

Unless required by applicable law or agreed to in writing, the software is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2484 sprintf-js 1.0.3 1.0.3

1.2484.1 Available under license :

Copyright (c) 2007-2014, Alexandru Marasteanu <hello [at) alexei (dot) ro>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of this software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2485 @babel/plugin-transform-parameters
7.13.0

1.2485.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2486 pip 9.0.1

1.2486.1 Available under license:
Copyright (c) 2008-2016 The pip developers (see AUTHORS.txt file)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2487 mimic-response 2.1.0
1.2487.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2488 aws4 1.11.0 1.11.0
1.2488.1 Available under license:
Copyright 2013 Michael Hart (michael.hart.au@gmail.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.2489 jetty-webapp 9.4.34.v20201102

1.2489.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.
"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or
conditions of merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for
damages, including direct, indirect, special, incidental and
consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are
offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such
Contributor, and informs licensees how to obtain it in a reasonable
manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained
within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore,
if a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and
other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such Commercial
Contributor in connection with its distribution of the Program in a commercial
product offering. The obligations in this section do not apply to any claims
or Losses relating to any actual or alleged intellectual property
infringement. In order to qualify, an Indemnified Contributor must:
a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the
Commercial Contributor in, the defense and any related settlement
negotiations. The Indemnified Contributor may participate in any such claim at
its own expense.

For example, a Contributor might include the Program in a commercial product
offering, Product X. That Contributor is then a Commercial Contributor. If
that Commercial Contributor then makes performance claims, or offers
warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=====================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.
Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications
are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay
The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL/GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2490 locate-path 5.0.0
1.2490.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2491 json3 3.3.2
1.2491.1 Available under license:

No license file was found, but licenses were detected in source scan.

# JSON 3 #

![JSON 3 Logo](https://bestiejs.github.io/json3/page/logo.png)
**JSON 3** is a modern JSON implementation compatible with a variety of JavaScript platforms. The current version is **3.3.2**.

- [Development Version](http://cdnjs.cloudflare.com/ajax/libs/json3/3.3.2/json3.js) *(43 KB; uncompressed with comments)*
- [Production Version](http://cdnjs.cloudflare.com/ajax/libs/json3/3.3.2/json3.min.js) *(3.5 KB; compressed and `gzip`-ped)*
- [Change Log](https://bestiejs.github.io/json3/changes.html)

Special thanks to [cdnjs](http://cdnjs.com/libraries/json3/) and [jsDelivr](http://www.jsdelivr.com/#!json3) for hosting CDN copies of JSON 3.

[JSON](http://json.org/) is a language-independent data interchange format based on a loose subset of the JavaScript grammar. Originally popularized by [Douglas Crockford](http://www.crockford.com/), the format was standardized in the [fifth edition](http://es5.github.io/) of the ECMAScript specification. The 5.1 edition, ratified in June 2011, incorporates several modifications to the grammar pertaining to the serialization of dates.

JSON 3 exposes two functions: `stringify()` for serializing a JavaScript value to JSON, and `parse()` for producing a JavaScript value from a JSON source string. It is a **drop-in replacement** for [JSON 2](http://json.org/js). The functions behave exactly as described in the ECMAScript spec, **except** for the date serialization discrepancy noted below.

The JSON 3 parser does **not** use `eval` or regular expressions. This provides security and performance benefits in obsolete and mobile environments, where the margin is particularly significant. The complete [benchmark suite](http://jsperf.com/json3) is available on [jsPerf](http://jsperf.com/).

The project is [hosted on GitHub](http://git.io/json3), along with the [unit tests](http://bestiejs.github.io/json3/test/test_browser.html). It is part of the [BestieJS](https://github.com/bestiejs) family, a collection of best-in-class JavaScript libraries that promote cross-platform support, specification precedents, unit testing, and plenty of documentation.

# Changes from JSON 2 #

JSON 3

* Correctly serializes primitive wrapper objects.
* Throws a 'TypeError' when serializing cyclic structures (JSON 2 recurses until the call stack overflows).
* Utilizes **feature tests** to detect broken or incomplete *native* JSON implementations (JSON 2 only checks for the presence of the native functions). The tests are only executed once at runtime, so there is no additional performance cost when parsing or serializing values.

**As of v3.2.3**, JSON 3 is compatible with [Prototype](http://prototypejs.org) 1.6.1 and older.
In contrast to JSON 2, JSON 3 **does not**

* Add `toJSON()` methods to the `Boolean`, `Number`, and `String` prototypes. These are not part of any standard, and are made redundant by the design of the `stringify()` implementation.
* Add `toJSON()` or `toISOString()` methods to `Date.prototype`. See the note about date serialization below.

## Date Serialization

**JSON 3 deviates from the specification in one important way**: it does not define `Date#toISOString()` or `Date#toJSON()`. This preserves CommonJS compatibility and avoids polluting native prototypes. Instead, date serialization is performed internally by the `stringify()` implementation: if a date object does not define a custom `toJSON()` method, it is serialized as a [simplified ISO 8601 date-time string](http://es5.github.com/#x15.9.1.15).

**Several native `Date#toJSON()` implementations produce date time strings that do *not* conform to the grammar outlined in the spec**. For instance, all versions of Safari 4, as well as JSON 2, fail to serialize extended years correctly. Furthermore, JSON 2 and older implementations omit the milliseconds from the date-time string (optional in ES 5, but required in 5.1). Finally, in all versions of Safari 4 and 5, serializing an invalid date will produce the string `"Invalid Date"`, rather than `null`. Because these environments exhibit other serialization bugs, however, JSON 3 will override the native `stringify()` implementation.

Portions of the date serialization code are adapted from the [`date-shim`](https://github.com/Yaffle/date-shim) project.

## Usage

### Web Browsers

```
<script src="//cdnjs.cloudflare.com/ajax/libs/json3/3.3.2/json3.min.js"></script>
<script>
    JSON.stringify({"Hello":123});
    // => {'Hello': 123}
    JSON.parse("[[1, 2, 3], 1, 2, 3, 4]", function (key, value) {
        if (typeof value == "number") {
            value = value % 2 ? "Odd" : "Even";
        }
        return value;
    });
    // => [["Odd", "Even", "Odd"], "Odd", "Even", "Odd", "Even"]
</script>
```

**When used in a web browser**, JSON 3 exposes an additional `JSON3` object containing the `noConflict()` and `runInContext()` functions, as well as aliases to the `stringify()` and `parse()` functions.

### `noConflict` and `runInContext`

* `JSON3.noConflict()` restores the original value of the global `JSON` object and returns a reference to the `JSON3` object.
* `JSON3.runInContext([context, exports])` initializes JSON 3 using the given `context` object (e.g., `window`, `process` etc.).
`global`, etc.), or the global object if omitted. If an `exports` object is specified, the `stringify()`, `parse()`, and `runInContext()` functions will be attached to it instead of a new object.

### Asynchronous Module Loaders

JSON 3 is defined as an [anonymous module](https://github.com/amdjs/amdjs-api/wiki/AMD#define-function-) for compatibility with [RequireJS](http://requirejs.org/), [`curl.js`](https://github.com/cujojs/curl), and other asynchronous module loaders.

```html
<script src="https://cdnjs.cloudflare.com/ajax/libs/require.js/2.1.10/require.js"></script>
<script>
require({
  "paths": {
    "json3": "./path/to/json3"
  },
  "json3": function (JSON) {
    JSON.parse("[1, 2, 3]");
    // => [1, 2, 3]
  }
});
</script>
```

To avoid issues with third-party scripts, **JSON 3 is exported to the global scope even when used with a module loader**. If this behavior is undesired, `JSON3.noConflict()` can be used to restore the global `JSON` object to its original value.

**Note:** If you intend to use JSON3 alongside another module, **please do not simply concatenate these modules together**, as that would cause multiple `define` calls in one script, resulting in errors in AMD loaders. The `r.js` build optimizer can be used instead if you need a single compressed file for production.

## CommonJS Environments

```javascript
var JSON3 = require("./path/to/json3");
JSON3.parse([1, 2, 3]);
// => [1, 2, 3]
```

## JavaScript Engines

```javascript
load("path/to/json3.js");
JSON.stringify({"Hello": 123, "Good-bye": 456}, ["Hello"], "\t");
// => '{\n\t"Hello": 123\n}'
```

# Compatibility #

JSON 3 has been **tested** with the following web browsers, CommonJS environments, and JavaScript engines.

## Web Browsers

- Windows [Internet Explorer](http://windows.microsoft.com/en-us/internet-explorer/download-ie), version 6.0 and
higher
- Google [Chrome](http://www.google.com/chrome), version 19.0 and higher
- Mozilla [Firefox](https://www.mozilla.org/en-US/firefox/new/), version 2.0 and higher
- Apple [Safari](http://www.apple.com/safari/), version 3.0 and higher
- [Opera](http://www.opera.com/) 8.54 and higher
- [SeaMonkey](http://www.seamonkey-project.org/) 1.0 and higher

## CommonJS Environments

- [Node](http://nodejs.org/) 0.6.21 and higher
- [io.js](https://iojs.org/) 1.0.3 and higher
- [RingoJS](http://ringojs.org/) 0.9 and higher
- [Narwhal](https://github.com/280north/narwhal) 0.3.2

## JavaScript Engines

- Mozilla [Rhino](https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino) 1.7R3 and higher
- WebKit [JSC](https://trac.webkit.org/wiki/JSC)
- Google [V8](http://code.google.com/p/v8/)

## Known Incompatibilities

* Attempting to serialize the `arguments` object may produce inconsistent results across environments due to specification version differences. As a workaround, please convert the `arguments` object to an array first: `JSON.stringify([...].slice.call(arguments, 0))`.

## Required Native Methods

JSON 3 assumes that the following methods exist and function as described in the ECMAScript specification:

- The `Number`, `String`, `Array`, `Object`, `Date`, `SyntaxError`, and `TypeError` constructors.
- `String.fromCharCode`
- `Object#toString`
- `Object#hasOwnProperty`
- `Function#call`
- `Math.floor`
- `Number#toString`
- `Date#valueOf`
- `String.prototype`: `indexOf`, `charCodeAt`, `charAt`, `slice`, `replace`.
- `Array.prototype`: `push`, `pop`, `join`.

# Contribute #

Check out a working copy of the JSON 3 source code with [Git](http://git-scm.com):

```
$ git clone git://github.com/bestiejs/json3.git
$ cd json3
```
We ♥ bug reports, suggestions, questions, and pull requests! Please see our [contribution guidelines](https://bestiejs.github.io/json3/contribute.html) if you'd like to contribute.

JSON 3 is released under the [MIT License](http://kit.mit-license.org/).

Found in path(s):
* /opt/cola/permits/1127086146_1611766309.78/0/fis-components-json3-3-3-2-0-g3cb2f1b-tar-gz/fis-components-json3-2b3d145/README.md

### 1.2492 lodash.get 4.4.2

#### 1.2492.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====
Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2493 groovy-json 3.0.8

1.2493.1 Available under license:

No license file was found, but licenses were detected in source scan.

```
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.
#
org.apache.groovy.json.DefaultFastStringServiceFactory
```

Found in path(s):
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-jar/META-INF/services/org.apache.groovy.json.FastStringServiceFactory

No license file was found, but licenses were detected in source scan.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
```
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
*    http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-
  jar/org/apache/groovy/json/internal/ValueMap.java
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-
  jar/org/apache/groovy/json/internal/ValueMapImpl.java
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-
  jar/org/apache/groovy/json/internal/Sys.java
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-
  jar/org/apache/groovy/json/internal/JsonParserLax.java
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-
  jar/org/apache/groovy/json/internal/Dates.java
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-
  jar/org/apache/groovy/json/FastStringService.java
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-
  jar/org/apache/groovy/json/JsonTokenType.java
* /opt/cola/permits/1218308217_1634929155.36/0/groovy-json-3-0-8-sources-1-
  jar/org/apache/groovy/json/internal/CharScanner.java
1.2494 express 4.16.3 4.16.3

1.2494.1 Available under license:

(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2013-2014 Roman Shtylman <shtylman+expressjs@gmail.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2495 avatica-metrics 1.15.0
1.2495.1 Available under license:

Apache Calcite Avatica Metrics
Copyright 2012-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2496 tephra-hbase-compat-1.0 0.6.0

1.2496.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright © 2012-2014 Cask Data, Inc.
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1120497655_1609977972.89/0/tephra-hbase-compat-1-0-0-6-0-sources-1-jar/co/cask/tephra/hbase10/coprocessor/TransactionProcessor.java
* /opt/cola/permits/1120497655_1609977972.89/0/tephra-hbase-compat-1-0-0-6-0-sources-1-jar/co/cask/tephra/hbase10/Filters.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright © 2014 Cask Data, Inc.
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License. You may obtain a copy of
the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under
the License.
*/

Found in path(s):
* /opt/cola/permits/1120497655_1609977972.89/0/tephra-hbase-compatible-1.0.0.6-0-sources-1-
jar/co/cask/tephra/hbase10/TransactionAwareHTable.java
* /opt/cola/permits/1120497655_1609977972.89/0/tephra-hbase-compatible-1.0.0.6-0-sources-1-
jar/co/cask/tephra/hbase10/SecondaryIndexTable.java
* /opt/cola/permits/1120497655_1609977972.89/0/tephra-hbase-compatible-1.0.0.6-0-sources-1-
jar/co/cask/tephra/hbase10/coprocessor/TransactionVisibilityFilter.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright © 2015 Cask Data, Inc.
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License. You may obtain a copy of
the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under
the License.
*/

Found in path(s):
* /opt/cola/permits/1120497655_1609977972.89/0/tephra-hbase-compatible-1.0.0.6-0-sources-1-
jar/co/cask/tephra/hbase10/HBase10ConfigurationProvider.java

1.2497 groovyjsr 3.0.8
1.2497.1 Available under license:

No license file was found, but licenses were detected in source scan.

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

org.codehaus.groovy.jsr223.GroovyScriptEngineFactory

Found in path(s):
  * /opt/cola/permits/1218385734_1634925264.65/0/groovy-jsr223-3-0-8-sources-jar/META-INF/services/javax.script.ScriptEngineFactory

No license file was found, but licenses were detected in source scan.

/*
 * The initial contribution was derived from the reference implementation
 * developed by Sun in consultation with the Groovy community. The reference
 * implementation had the following license header:
 *
 * Copyright 2006 Sun Microsystems, Inc. All rights reserved.
 * Use is subject to license terms.
 * *
 * Redistribution and use in source and binary forms, with or without modification, are
 * permitted provided that the following conditions are met: Redistributions of source code
 * must retain the above copyright notice, this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright notice, this list of
 * conditions and the following disclaimer in the documentation and/or other materials
 * provided with the distribution. Neither the name of the Sun Microsystems nor the names of
 * contributors may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
 * ANY EXPRESS
 * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
 * MERCHANTABILITY
 * AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER
 */
* OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON
* ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGligence
* OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE
* POSSIBILITY OF SUCH DAMAGE.
*
* Subsequent modifications by the Groovy community have been done under the Apache License v2:
*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1218385734_1634925264.65/0/groovy-jsr223-3-0-8-sources-
jar/org/codehaus/groovy/jsr223/GroovyScriptEngineFactory.java
* /opt/cola/permits/1218385734_1634925264.65/0/groovy-jsr223-3-0-8-sources-
jar/org/codehaus/groovy/jsr223/GroovyCompiledScript.java
* /opt/cola/permits/1218385734_1634925264.65/0/groovy-jsr223-3-0-8-sources-
jar/org/codehaus/groovy/jsr223/GroovyScriptEngineImpl.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
1.2498 jsesc 1.3.0

1.2498.1 Available under license:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2499 ipaddr.js 1.0.5
1.2499.1 Available under license:

Copyright (C) 2011 Peter Zotov <whitequark@whitequark.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2500 @babel/highlight 7.0.0

1.2500.1 Available under license:

MIT License

Copyright (c) 2014-2018 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2501 guava 22.0-android

1.2501.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2015 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/hash/MacHashFunction.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/io/ReaderInputStream.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/util/concurrent/AggregateFutureState.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/util/concurrent/Platform.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/hash/LittleEndianByteArray.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/util/concurrent/CombinedFuture.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/collect/ConsumingQueueIterator.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *}
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll...er.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/RowSortedTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/ForwardingListMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/SortedMapDifference.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/WellBehavedMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/GwtTransient.java

No license file was found, but licenses were detected in source scan.

/*/ 
* Copyright (C) 2010 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/AbstractSequentialIterator.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/UnmodifiableListIterator.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/RowSortedTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/ForwardingListMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/coll/SortedMapDifference.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/MinMaxPriorityQueue.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingSortedSetMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingImmutableCollection.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingSetMultimap.java
No license file was found, but licenses were detected in source scan.

/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/AtomicDoubleArray.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/cache/Striped64.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/cache/LongAdder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/graph/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 The Guava Authors
* 

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/CharMatcher.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/Bytes.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/internal/Finalizer.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Converter.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/Longs.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/net/InetAddresses.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/Ints.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Stopwatch.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/Booleans.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/SerializingExecutor.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/FluentIterable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Joiner.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/escape/Escaper.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/net/TrieParser.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/thirdparty/publicsuffix/TrieParser.java
jar/com/google/common/io/MultiReader.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/Shorts.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/AbstractCheckedFuture.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/net/PercentEscaper.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/Floats.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/Chars.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of { @code Objects} only in web. Intended to be empty for regular
* version.
*/

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of
* the License at
* http://www.apache.org/licenses/LICENSE-2.0
 *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/collect/SortedMultisets.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/collect/SortedMultiset.java
No license file was found, but licenses were detected in source scan.

/
* Copyright (C) 2005 The Guava Authors
 *
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
 *
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
  jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

/
* Copyright (C) 2011 The Guava Authors
 *
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
 *
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
* either
* express or implied. See the License for the specific language governing permissions and
Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/Striped.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2013 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/MultimapBuilder.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This following method is a modified version of one found in
 * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
 * which contained the following notice:
 *
 * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
 * the public domain, as explained at http://creativecommons.org/publicdomain/zer/1.0/
* Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
*/

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/reflect/TypeVisitor.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Verify.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/VerifyException.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/AbstractTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/eventbus/SubscriberExceptionHandler.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/Runnables.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/FilteredMultimapValues.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/io/CharSequenceReader.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/hash/HashingInputStream.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/graph/AbstractGraph.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/graph/UndirectedGraphConnections.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/graph/ValueGraph.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
/*
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ImmutableMultisetGwtSerializationDependencies.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/graph/ConfigurableValueGraph.java
*/
jar/com/google/common/collect/TreeRangeMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/AbstractMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/DescendingImmutableSortedSet.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/RangeMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/DescendingMultiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/FilteredSetMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/FilteredMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/AllEqualOrdering.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/SortedMultisetBridge.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/RegularImmutableAsList.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2014 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 */
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/ImmutableNetwork.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/ImmutableGraph.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/Graph.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/MutableGraph.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/Network.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/TopKSelector.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/MutableNetwork.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/Graphs.java

No license file was found, but licenses were detected in source scan.

/

* Copyright (C) 2010 The Guava Authors

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and limitations under
* the License.
*

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/Atomics.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/Monitor.java
Open Source Used In DNAC Platform Antelope 10187

* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/net/package-info.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Equivalence.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Strings.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/annotations/package-info.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/package-info.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/annotations/Beta.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Ascii.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ContiguousSet.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 * 
 *
MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author hereby disclaims copyright to this source code.

No license file was found, but licenses were detected in source scan.

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Open Source Used In DNAC Platform Antelope 10189
jar/com/google/common/base/Preconditions.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/eventbus/package-info.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/io/ByteStreams.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/io/Closeables.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/base/Defaults.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/eventbus/EventBus.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/Interners.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/base/Functions.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/package-info.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/base/Suppliers.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2008 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/StandardTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/UnmodifiableIterator.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/ImmutableBiMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collect/StandardRowSortedTable.java

Open Source Used In DNAC Platform: Antelope 10190
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/EnumBiMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/MapDifference.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/ListMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/ForwardingSet.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/ForwardingMultiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/TreeMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/AbstractMapBasedMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/AbstractMapEntry.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/Multimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/ForwardingMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/ForwardingMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collection/ClassToInstanceMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
Open Source Used In DNAC Platform Antelope 10193
Open Source Used In DNAC Platform Antelope 10194

jar/com/google/common/collection/ImmutableList.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingIterator.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingQueue.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/Iterables.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/NullsLastOrdering.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/Lists.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/AbstractCollection.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/AbstractMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/AbstractMultiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingList.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ByFunctionOrdering.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/Sets.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/Maps.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/HashMultiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/Lists.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/AbstractBiMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/SortedSetMultimap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingObject.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/LinkedHashMultiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingMapEntry.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/MutableClassToInstanceMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/Multisets.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ImmutableMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/LinkedHashSet.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingSortedList.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingQueue.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/collection/ForwardingIterator.java
jar/com/google/common/collect/AbstractIterator.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/Multiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ConcurrentHashMultiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ReverseNaturalOrdering.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ForwardingConcurrentMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/SingletonImmutableSet.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/AbstractSetMultimap.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ImmutableRangeSet.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/reflect/MutableTypeToInstanceMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/io/ByteSink.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/math/PairedStatsAccumulator.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/RateLimiter.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):

*/

Open Source Used In DNAC Platform Antelope 10197
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ImmutableEnumSet.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/RegularImmutableList.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ImmutableTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/RegularImmutableSortedSet.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ImmutableSortedMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ForwardingTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/SingletonImmutableTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/EmptyImmutableSetMultimap.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2012 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims copyright to this source code.
 */

Found in path(s):
/*
 * Copyright (C) 2014 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/ListenerCallQueue.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/math/Quantiles.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/eventbus/Dispatcher.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/eventbus/Subscriber.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/MoreObjects.java

/*
 * Copyright (C) 2017 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under

Open Source Used In DNAC Platform Antelope 10189
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/ImmutableLongArray.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/ImmutableDoubleArray.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
/*
 * Outer class that exists solely to let us write { @code Partially.GwtIncompatible } instead of plain
 * { @code GwtIncompatible }. This is more accurate for { @link Futures#catching }, which is available
 * under GWT but with a slightly different signature.
 *
 * We can't use { @code PartiallyGwtIncompatible } because then the GWT compiler wouldn't recognize
 * it as a { @code GwtIncompatible } annotation. And for { @code Futures.catching }, we need the GWT
 * compiler to autostrip the normal server method in order to expose the special, inherited GWT
 * version.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2006 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 */
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
}/*

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/escape/CharEscaper.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/base/CaseFormat.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/ImmediateFuture.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/reflect/TypeToken.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/annotations/VisibleForTesting.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/io/PatternFilenameFilter.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/Futures.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-
jar/com/google/common/io/AppendableWriter.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/DenseImmutableTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/escape/Platform.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/SparseImmutableTable.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/net/UrlEscapers.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
*/
Open Source Used In DNAC Platform Antelope 10206

* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/cache/RemovalListener.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/cache/LoadingCache.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/cache/CacheLoader.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/cache/ForwardingCache.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/hash/AbstractStreamingHashFunction.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/net/MediaType.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Absent.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/net/HttpHeaders.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Optional.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/math/LongMath.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/reflect/TypeParameter.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/net/HostAndPort.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/math/BigIntegerMath.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/hash/Funnels.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/hash/AbstractStreamingHashFunction.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/cache/ForwardingCache.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/cache/CACHELoader.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/hash/BloomFilter.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/hash/HashingOutputStream.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/EmptyContiguousSet.java
Open Source Used In DNAC Platform Antelope 10207

* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/math/package-info.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Signing.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/UnsignedInts.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/cache/RemovalCause.java

Open Source Used In DNAC Platform Antelope 10207

* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/math/package-info.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/base/Signing.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/primitives/UnsignedInts.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/cache/RemovalCause.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2017 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/BaseGraph.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ObjectCountLinkedHashMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/ObjectCountHashMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/AbstractBaseGraph.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/graph/EnumCountHashMap.java
* /opt/ws_local/PERMITS_SQL/1069088313_1594481589.37/0/guava-22-0-android-sources-jar/com/google/common/collect/AbstractObjectCountMap.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 The Guava Authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

1.2502 swagger-core 1.5.0

1.2503 dropwizard-metrics-hadoop-metrics2-reporter 0.1.2

1.2503.1 Available under license:

Copyright 2016 Josh Elser

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2504 boolbase 1.0.0

1.2504.1 Available under license:

Dojo is available under *either* the terms of the modified BSD license *or* the Academic Free License version 2.1. As a recipient of Dojo, you may choose which license to receive this code under (except as noted in per-module LICENSE files). Some modules may not be the copyright of the Dojo Foundation. These modules contain explicit declarations of copyright in both the LICENSE files in
the directories in which they reside and in the code itself. No external contributions are allowed under licenses which are fundamentally incompatible with the AFL or BSD licenses that Dojo is distributed under.

The text of the AFL and BSD licenses is reproduced below.

-------------------------------------------------------------------------------

The "New" BSD License:
************************

Copyright (c) 2005-2015, The Dojo Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Dojo Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-------------------------------------------------------------------------------

The Academic Free License, v. 2.1:
*****************************************

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide,
royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

a) to reproduce the Original Work in copies;

b) to prepare derivative works ("Derivative Works") based upon the Original Work;

c) to distribute copies of the Original Work and Derivative Works to the public;

d) to perform the Original Work publicly; and

e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) This section intentionally omitted.
6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately proceeding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim,
against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys’ fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.
1.2505 spdx-exceptions 2.3.0

1.2505.1 Available under license:

No license file was found, but licenses were detected in source scan.

The package exports an array of strings. Each string is an identifier for a license exception under the [Software Package Data Exchange (SPDX)](https://spdx.org) software license metadata standard.

[SPDX]: https://spdx.org

## Copyright and Licensing

### SPDX

"SPDX" is a federally registered United States trademark of The Linux Foundation Corporation.

From version 2.0 of the [SPDX] specification:

> Copyright © 2010-2015 Linux Foundation and its Contributors. Licensed under the Creative Commons Attribution License 3.0 Unported. All other rights are expressly reserved.

The Linux Foundation and the SPDX working groups are good people. Only they decide what "SPDX" means, as a standard and otherwise. I respect their work and their rights. You should, too.

### This Package

> I created this package by copying exception identifiers out of the SPDX specification. That work was mechanical, routine, and required no creativity whatsoever. - Kyle Mitchell, package author

United States users concerned about intellectual property may wish to discuss the following Supreme Court decisions with their attorneys:

- _Baker v. Selden_, 101 U.S. 99 (1879)


Found in path(s):

* /opt/cola/permits/1136088994_1613675065.5/0/spdx-exceptions-2-3-0-5-tgz/package/README.md
1.2506 @babel/helper-module-imports 7.13.12
1.2506.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2507 ignite-core 2.11.0
1.2507.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

==============================================================================
Apache Ignite Subcomponents:

The Apache Ignite project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the following licenses.

For SnapTree:

This product bundles SnapTree, which is available under a "3-clause BSD" license. For details, see https://github.com/nbronson/snaptree/blob/master/LICENSE.

For JSR 166 classes in "org.jsr166" package

This product bundles JSR-166 classes which are donated to public domain. For details, see CC0 1.0 Universal (1.0), Public Domain Dedication, http://creativecommons.org/publicdomain/zero/1.0/

For books used for tests in "org.apache.ignite.internal.processors.hadoop.books"

This code bundles book text files used for testing purposes which contain the following header:

This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.org

// List of ignite-core module's dependencies provided as a part of this distribution.

For JSR107 API and SPI (https://github.com/jsr107/jsr107spec) javax.cache:cache-api:jar:1.0.0

This product bundles JSR107 API and SPI which is available under the following: JSR-000107 JCACHE 2.9 Public Review - Updated Specification License. For details, see https://raw.githubusercontent.com/jsr107/jsr107spec/master/LICENSE.txt.

For ignite-shmem (http://www.gridgain.com) org.gridgain:ignite-shmem:jar:1.0.0

This product bundles ignite-shmem which is available under the following: The Apache Software License, Version 2.0. For details, see http://www.apache.org/licenses/LICENSE-2.0.txt.

This product bundles JetBrains Java Annotations which is available under the following:
The Apache Software License, Version 2.0. For details, see http://www.apache.org/license/LICENSE-2.0.txt.
Apache Ignite
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software includes code from IntelliJ IDEA Community Edition
Copyright (C) JetBrains s.r.o.
https://www.jetbrains.com/idea/
Licensed under Apache License, Version 2.0.
http://search.maven.org/#artifactdetails%7Corg.jetbrains%7Cannotations%7C13.0%7Cjar

1.2508 babel-helper-regex 6.26.0 6.26.0
1.2508.1 Available under license :
   The MIT License (MIT)

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYright HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2509 formidable 1.2.2 1.2.2
1.2509.1 Available under license :
   The MIT License (MIT)

   Copyright (c) 2011-present Felix Geisendrfer, and contributors.

   Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2510 cache-api 1.0.0

1.2510.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1,
you must provide the Installation Information in the manner
specified by section 6 of the GNU GPL for conveying Corresponding
Source.)


You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
1.2511.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
1.2512 jctools-core 3.1.0

1.2512.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section)
patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and where such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2513 influx 5.6.3
1.2513.1 Available under license :
MIT License

Copyright (c) 2016 Connor Peet

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2514 ms 2.1.1
1.2514.1 Available under license :
The MIT License (MIT)

Copyright (c) 2016 Zeit, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2515 espree 3.5.4

1.2515.1 Available under license:

Espree
Copyright JS Foundation and other contributors, https://js.foundation

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2516 bouncy-castle 1.57

1.2516.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2517 array-unique 0.2.1

1.2517.1 Available under license : 
The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2518 string-prototype-trimend 1.0.4

1.2518.1 Available under license : 
MIT License

Copyright (c) 2017 Khaled Al-Ansari
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2519 istanbul-lib-hook 3.0.0

1.2519.1 Available under license:

Copyright 2012-2015 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
1.2520 split-on-first 1.1.0 1.1.0
1.2520.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2521 hk2-locator 2.5.0-b32
1.2521.1 Available under license:
CDDL + GPLv2 with classpath exception
https://glassfish.java.net/nonav/public/CDDL+GPL_1_1.html
A business-friendly OSS license

1.2522 inflight 1.0.6
1.2522.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2523 babel-helper-define-map 6.26.0 6.26.0

1.2523.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2524 util-deprecate 1.0.2

1.2524.1 Available under license:
(The MIT License)

Copyright (c) 2014 Nathan Rajlich <nathan@tootallnate.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
1.2525 is-number 3.0.0

1.2525.1 Available under license:
MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2526 string_decoder 1.0.3

1.2526.1 Available under license:
Node.js is licensed for use as follows:

----

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

==

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

==

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

==

1.2527 request-promise 4.2.6

1.2527.1 Available under license:

ISC License

Copyright (c) 2020, Nicolai Kamenzky, Ty Abonil, and contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2528 binary 0.3.0
1.2528.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2529 semver 7.0.0
1.2529.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2530 p-limit 2.3.0 2.3.0
1.2530.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2531 jsrender 1.0.4
1.2531.1 Available under license :

Copyright (c) 2019 Boris Moore https://github.com/BorisMoore/jsrender

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
1.2532 has-color 0.1.7 0.1.7

1.2532.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2533 setprototypeof 1.0.3 1.0.3

1.2533.1 Available under license:

Copyright (c) 2015, Wes Todd

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2534 jline 2.9
1.2535 jetty-servlet 9.4.34.v20201102

1.2535.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.
2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the
5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not pertain
to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

====================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html
* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html
OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2536 html-to-xlsx 2.0.3
1.2536.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jan Blaha

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2537 depd 1.1.2 1.1.2
1.2537.1 Available under license:
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2538 parseurl 1.3.3
1.2538.1 Available under license :

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2539 type 2.1.0
1.2539.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Alejandro Celaya

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2540 map-visit 1.0.0
1.2540.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HoldERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2541 regenerator-runtime 0.11.1

1.2541.1 Available under license:
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2014-present, Facebook, Inc.
 * 
 * This source code is licensed under the MIT license found in the
 * LICENSE file in the root directory of this source tree.
 */

Found in path(s):
* /opt/cola/permits/1110631596_1606854664.8/0/regenerator-runtime-0-11-1-3-tgz/package/path.js
* /opt/cola/permits/1110631596_1606854664.8/0/regenerator-runtime-0-11-1-3-tgz/package/runtime-module.js
* /opt/cola/permits/1110631596_1606854664.8/0/regenerator-runtime-0-11-1-3-tgz/package/runtime.js

1.2542 bonecp 0.8.0.RELEASE

1.2542.1 Available under license:
No license file was found, but licenses were detected in source scan.

/**
 * Copyright 2010 Wallace Wadge
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 *     http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1128404392_1611918002.64/0/bonecp-0-8-0-release-sources-4-jar/com/jolbox/bonecp/ConnectionHandle.java
* /opt/cola/permits/1128404392_1611918002.64/0/bonecp-0-8-0-release-sources-4-
1.2543 batik-ttf2svg 1.13

1.2543.1 Available under license:

Apache Batik
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the
Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for
Standardization for the definition of character entities used in the software's
documentation.

This product includes images from the Tango Desktop Project
(http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2544 error-ex 1.3.2

1.2544.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 JD Ballard

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2545 shebang-regex 1.0.0

1.2545.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2546 is-glob 4.0.1

1.2546.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2547 from 0.1.7 0.1.7

1.2547.1 Available under license :
The MIT License

Copyright (c) 2011 Dominic Tarr
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License, Version 2.0

Copyright (c) 2011 Dominic Tarr

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2548 ansi-regex 4.1.0

1.2548.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2549 ws 1.1.5

1.2549.1 Available under license:

No license file was found, but licenses were detected in source scan.

(The MIT License)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/README.md

No license file was found, but licenses were detected in source scan.

/*!
* ws: a node.js websocket client
* Copyright(c) 2011 Einar Otto Stangvik <einaros@gmail.com>
* MIT Licensed
*/

Found in path(s):
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/lib/ErrorCodes.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/lib/Sender.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/lib/Receiver.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/lib/BufferUtil.fallback.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/lib/BufferUtil.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/lib/BufferPool.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/lib/WebSocketServer.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/index.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/WebSocket.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/Validation.fallback.js
*/opt/cola/permits/1110639655_1606843085.09/0/ws-1-1-5-2-tgz/package/lib/Sender.hixie.js
1.2550 has-symbols 1.0.0

1.2550.1 Available under license:
MIT License

Copyright (c) 2016 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2551 estraverse 4.3.0 4.3.0

1.2551.1 Available under license:
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2552 range-parser 1.2.0 1.2.0
1.2552.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015-2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2553 drill-format-pcapng 1.19.0
1.2553.1 Available under license:

Drill : Contrib : Format : Pcap-NG
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
your Contributions, as defined in Section 1 above, shall be your
own copyright or shall be licenced to You under the terms of
this License.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2554 content-type 1.0.4
1.2554.1 Available under license :
(The MIT License)

Copyright (c) 2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2555 nth-check 1.0.2
1.2555.1 Available under license :

Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2556 node-promise 0.5.14

1.2556.1 Available under license:

No license file was found, but licenses were detected in source scan.

MIT License.

The node-promise project provides a complete promise implementation. Promises provide a clean separation of concerns between asynchronous behavior and the interface so asynchronous functions can be called without callbacks, and callback interaction can be done on the generic promise interface. The node-promise package provides just a promise implementation, however, https://github.com/kriszyp/promised-io is recommended for more complete promise-based IO functionality. The promised-io includes the promise implementation from node-promise, as well as wrappers around Node's filesystem and other system I/O APIs for consistent promise-based interaction.

The node-promise module features a promise implementation with:

* Chainable promises
* Promises throw errors if an error handler is not provided
* CommonJS promise proposal [1] compliant
* Immutable once fulfilled to reduce possible side-effects
* Promises can be used securely (as separate resolver/promise pairs in opac situations)
* Backwards compatibility where possible (addCallback, addErrback, emitSuccess, and emitError should still behave as expected)

Utility functions, including:

* when() - Normalization of sync (normal values) and async (promises)
* all() - Create a promise that accumulate multiple concurrent promises (failed promises resolve to Error objects)
* allOrNone() - Ditto, but the first promise to fail causes the composition to fail as well
* first() - Find the first promise to be fulfilled in a group of promises
* seq() - Sequentially execute a set of promise returning functions
* delay() - Returns a promise that is fulfilled after a given amount of time
* execute() - Executes a function that takes a callback and returns a
promise (thank you Benjamin Thomas for providing this)

Much of this is adapted from Tyler Close's ref_send and Kris Kowal's work on promises.

Some quick examples from test-promise.js (again, it is recommended that you use http://github.com/kriszyp/promised-io for file and other I/O interaction):

```javascript
util = require('util');
var fs = require('./fs-promise');

// open a file and read it
fs.open("fs-promise.js", process.O_RDONLY).then(function(fd){
  return fs.read(fd, 4096);
}).then(function(args){
  util.puts(args[0]); // print the contents of the file
});

// does the same thing
fs.readFile("fs-promise.js").addCallback(util.puts);
```

A default Promise constructor can be used to create a self-resolving deferred.promise:

```javascript
var Promise = require("node-promise").Promise;
var promise = new Promise();
asyncOperation(function(){
  promise.resolve("successful result");
});
promise -> given to the consumer
```

A consumer can use the promise:

```javascript
promise.then(function(result){
  ... when the action is complete this is executed ...
},
function(error){
  ... executed when the promise fails
});
```

Alternately, a provider can create a deferred and resolve it when it completes an action. The deferred object a promise object that provides a separation of consumer and producer to protect promises from being fulfilled by untrusted code.

```javascript
var defer = require("node-promise").defer;
var deferred = defer();
asyncOperation(function(){
  deferred.resolve("successful result");
});
deferred.promise -> given to the consumer
```
Another way that a consumer can use promises:

```javascript
var when = require("node-promise").when;
when(promise, function(result){
    ... when the action is complete this is executed ...
},
function(error){
    ... executed when the promise fails
});
```

More examples:

```javascript
function printFirstAndLast(itemsDeferred){
    findFirst(itemsDeferred).then(util.puts);
    findLast(itemsDeferred).then(util.puts);
}
function findFirst(itemsDeferred){
    return itemsDeferred.then(function(items){
        return items[0];
    });
}
function findLast(itemsDeferred){
    return itemsDeferred.then(function(items){
        return items[items.length];
    });
}
```

And now you can do:

```javascript
printFirstAndLast(someAsyncFunction());
```

The workhorse function of this library is the "when" function, which provides a means for normalizing interaction with values and functions that may be a normal synchronous value, or may be a promise (asynchronously fulfilled). The when() function takes a value that may be a promise or a normal value for the first function, and when the value is ready executes the function provided as the second argument (immediately in the case of a non-promise normal value). The value returned from when() is the result of the execution of the provided function, and returns a promise if provided a promise or synchronously returns a normal value if provided a non-promise value. This makes it easy to "chain" computations together. This allows us to write code that is agnostic to sync/async interfaces:

```javascript
var when = require("node-promise").when;
function printFirstAndLast(items){
    // print the first and last item
    when(findFirst(items), util.puts);
    when(findLast(items), util.puts);
}
function findFirst(items){
    // return the first item
```
function findLast(items){
    // return the last item
    return when(items, function(items){
        return items[items.length - 1];
    });
}

Now we can do:

> printFirstAndLast([1,2,3,4,5]);
1
5

And we can also provide asynchronous promise:

    var promise = new process.Promise();
    > printFirstAndLast(promise);

    (nothing printed yet)

    > promise.emitSuccess([2,4,6,8,10]);
2
10

The "all" function is intended to provide a means for waiting for the completion of an array of promises. The "all" function should be passed an array of promises, and it returns an promise that is fulfilled once all the promises in the array are fulfilled. The returned promise's resolved value will be an array with the resolved values of all of the promises in the passed in array.

The "first" function is intended to provide a means for waiting for the completion of the first promise in an array of promises to be fulfilled. The "first" function should be passed an array of promises, and it returns an promise that is fulfilled once the first promise in the array is fulfilled. The returned promise's resolved value will be the resolved value of the first fulfilled promise.

// // MIT License

No license file was found, but licenses were detected in source scan.
1.2557 is-typedarray 1.0.0 1.0.0

1.2557.1 Available under license :
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2558 to-array 0.1.4 0.1.4

1.2558.1 Available under license :
Copyright (c) 2012 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 1998-2008 Greg Roelofs. All rights reserved.

This software is provided "as is," without warranty of any kind, express or implied. In no event shall the author or contributors be held liable for any damages arising in any way from the use of this software.

The contents of this file are DUAL-LICENSED. You may modify and/or redistribute this software according to the terms of one of the following two licenses (at your option):

LICENSE 1 ("BSD-like with advertising clause"):

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, disclaimer, and this list of conditions.
2. Redistributions in binary form must reproduce the above copyright notice, disclaimer, and this list of conditions in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

   This product includes software developed by Greg Roelofs and contributors for the book, "PNG: The Definitive Guide," published by O'Reilly and Associates.

LICENSE 2 (GNU GPL v2 or later):

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

pnm2png / png2pnm --- conversion from PBM/PGM/PPM-file to PNG-file

copyright (C) 1999-2019 by Willem van Schaik <willem at schaik dot com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE

PNG Reference Library License version 2

* Copyright (c) 1995-2019 The PNG Reference Library Authors.
* Copyright (c) 2018-2019 Cosmin Truta.
* Copyright (c) 1996-1997 Andreas Dilger.
* Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or
other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

   Simon-Pierre Cadieux
   Eric S. Raymond
   Mans Rullgard
   Cosmin Truta
   Gilles Vollant
   James Yu
   Mandar Sahastrabuddhe
   Google Inc.
   Vadim Barkov

and with the following additions to the disclaimer:

   There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

   Some files in the "contrib" directory and some configure-generated
files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

Some files in the "scripts" directory have other copyright owners, but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of
the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

---

**1.2560 user-home 2.0.0**

**1.2560.1 Available under license:**

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software,

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2561 jackson-dataformat-csv 2.10.5

1.2561.1 Available under license:

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bnd-LastModified: 1595302282882
Build-Jdk: 1.8.0_162
Built-By: tatu
Bundle-Description: Support for reading and writing CSV-encoded data via Jackson abstractions.
Bundle-DocURL: https://github.com/FasterXML/jackson-dataformats-text
Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.txt
Bundle-ManifestVersion: 2
Bundle-Name: Jackson-dataformat-CSV
Bundle-SymbolicName: com.fasterxml.jackson.dataformat.jackson-dataformat-csv
Bundle-Vendor: FasterXML
Bundle-Version: 2.10.5
Created-By: Apache Maven Bundle Plugin
Implementation-Build-Date: 2020-07-21 03:31:04+0000
Implementation-Title: Jackson-dataformat-CSV
Implementation-Vendor: FasterXML
Implementation-Vendor-Id: com.fasterxml.jackson.dataformat
Implementation-Version: 2.10.5
Import-Package: com.fasterxml.jackson.core;version="[2.10,3)";com.fasterxml.jackson.core.base;version="[2.10,3)";com.fasterxml.jackson.core.format;version="[2.10,3)";com.fasterxml.jackson.core.io;version="[2.10,3)";com.fasterxml.jackson.core.type;version="[2.10,3)";com.fasterxml.jackson.core.util;version="[2.10,3)";com.fasterxml.jackson.databind;version="[2.10,3)";com.fasterxml.jackson.databind.cfg;version="[2.10,3)";com.fasterxml.jackson.databind.introspect;version="[2.10,3)";com.fasterxml.jackson.databind.util;version="[2.10,3)";com.fasterxml.jackson.dataformat.csv
1.2562 minimist 1.2.0 1.2.0

1.2562.1 Available under license:

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2563 is-finite 1.0.2 1.0.2

1.2563.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2564 defer-to-connect 1.1.3
1.2564.1 Available under license :
   MIT License

Copyright (c) 2018 Szymon Marczak

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2565 guice 5.0.1
1.2565.1 Available under license :
   Apache License
   Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2566 parquet-column 1.10.0

1.2566.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
This product depends on fastutil and includes it in this binary artifact.
Fastutil provides type-specific collection implementations.

Copyright: 2002-2014 Sebastiano Vigna
Home page: http://fasutil.di.unimi.it/
License: http://www.apache.org/licenses/LICENSE-2.0.html

1.2567 is-core-module 2.0.0
1.2567.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Dave Justice

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2568 process-nextick-args 2.0.1 2.0.1
1.2568.1 Available under license :
# Copyright (c) 2015 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

1.2569 moment-timezone 0.5.33 0.5.33

1.2569.1 Available under license:

The MIT License (MIT)

Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2570 hamcrest 1.3

1.2570.1 Available under license:

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<title>Common Public License - v 1.0</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body background="#FFFFFF" vlink="#800000">

<p align="center"><b>Common Public License - v 1.0</b></p>

<!-- License text goes here -->
```

Open Source Used In DNAC Platform Antelope 10305
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:

i) changes to the Program, and
ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.
implied warranties or conditions of merchantability and fitness for a particular purpose; </FONT></UL>

<UL><FONT SIZE="2">ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; </FONT></UL>

<UL><FONT SIZE="2">iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and </FONT></UL>

<UL><FONT SIZE="2">iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange. </FONT><FONT SIZE="2" COLOR="#FF0000"> </FONT><FONT SIZE="2" COLOR="#FF0000"></FONT></UL>

<UL><FONT SIZE="2"> When the Program is made available in source code form: </FONT></UL>

<UL><FONT SIZE="2">a) it must be made available under this Agreement; and </FONT></UL>

<UL><FONT SIZE="2">b) a copy of this Agreement must be included with each copy of the Program. </FONT></UL>

<P><STRIKE><STRIKE></STRIKE>Contributors may not remove or alter any copyright notices contained within the Program. </P>

<P>Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution. </P>

<P><B>4. COMMERCIAL DISTRIBUTION</B></P>

<P>Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.</P>
For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, and assumes all risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after
Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
Copyright (c) 2000-2003, jMock.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of jMock nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
EasyMock License (MIT License)

Copyright (c) 2001-2006 OFFIS, Tammo Freese.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.2571 commons-math 3.1.1

1.2571.1 Available under license:

Apache Commons Math
Copyright 2001-2012 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

===============================================================================

The BracketFinder (package org.apache.commons.math3.optimization.univariate) and PowellOptimizer (package org.apache.commons.math3.optimization.general) classes are based on the Python code in module "optimize.py" (version 0.5) developed by Travis E. Oliphant for the SciPy library (http://www.scipy.org/) Copyright © 2003-2009 SciPy Developers.

===============================================================================

The LinearConstraint, LinearObjectiveFunction, LinearOptimizer, RelationShip, SimplexSolver and SimplexTableau classes in package org.apache.commons.math3.optimization.linear include software developed by Benjamin McCann (http://www.benmccann.com) and distributed with the following copyright: Copyright 2009 Google Inc.

===============================================================================

This product includes software developed by the University of Chicago, as Operator of Argonne National Laboratory.
The LevenbergMarquardtOptimizer class in package org.apache.commons.math3.optimization.general includes software translated from the lmder, lmpr and qrsolv Fortran routines from the Minpack package
Minpack Copyright Notice (1999) University of Chicago. All rights reserved

===============================================================================

The GraggBulirschStoerIntegrator class in package org.apache.commons.math3.ode.nonstiff includes software translated from the odex Fortran routine developed by E. Hairer and G. Wanner.
Original source copyright:
Copyright (c) 2004, Ernst Hairer

===============================================================================

The EigenDecompositionImpl class in package org.apache.commons.math3.linear includes software translated from some LAPACK Fortran routines. Original source copyright:
Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.
The MersenneTwister class in package org.apache.commons.math3.random includes software translated from the 2002-01-26 version of the Mersenne-Twister generator written in C by Makoto Matsumoto and Takuji Nishimura. Original source copyright:
Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, All rights reserved

The LocalizedFormatsTest class in the unit tests is an adapted version of the OrekitMessagesTest class from the orekit library distributed under the terms of the Apache 2 licence. Original source copyright:
Copyright 2010 CS Systèmes d'Information

The HermiteInterpolator class and its corresponding test have been imported from the orekit library distributed under the terms of the Apache 2 licence. Original source copyright:
Copyright 2010-2012 CS Systèmes d'Information

The creation of the package "o.a.c.m.analysis.integration.gauss" was inspired by an original code donated by Sébastien Brisard.

The complete text of licenses and disclaimers associated with the original sources enumerated above at the time of code translation are in the LICENSE.txt file.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all...
other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE COMMONS MATH DERIVATIVE WORKS:

The Apache commons-math library includes a number of subcomponents whose implementation is derived from original sources written in C or Fortran. License terms of the original sources are reproduced below.
For the lmder, lmpar and qrsolv Fortran routine from minpack and translated in
the LevenbergMarquardtOptimizer class in package
org.apache.commons.math3.optimization.general

Original source copyright and license statement:

Minpack Copyright Notice (1999) University of Chicago. All rights reserved

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above
copyright notice, this list of conditions and the following
disclaimer.

2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials
provided with the distribution.

3. The end-user documentation included with the
redistribution, if any, must include the following
acknowledgment:

"This product includes software developed by the
University of Chicago, as Operator of Argonne National
Laboratory.

Alternately, this acknowledgment may appear in the software
itself, if and wherever such third-party acknowledgments
normally appear.

4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS"
WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE
UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND
THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE
OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY
OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR
USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF
THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4)
DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION
UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL
BE CORRECTED.
5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT
HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF
ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT,
INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF
ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF
PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER
SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT
(INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE,
EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE
POSSIBILITY OF SUCH LOSS OR DAMAGES.

===============================================================================

Copyright and license statement for the odex Fortran routine developed by
E. Hairer and G. Wanner and translated in GraggBulirschStoerIntegrator class
in package org.apache.commons.math3.ode.nonstiff:

Copyright (c) 2004, Ernst Hairer

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===============================================================================

Copyright and license statement for the original lapack fortran routines
translated in EigenDecompositionImpl class in package
org.apache.commons.math3.linear:

Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.

- Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright and license statement for the original Mersenne twister C routines translated in MersenneTwister class in package org.apache.commons.math3.random:

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===============================================================================

The class "org.apache.commons.math3.exception.util.LocalizedFormatsTest" is an adapted version of "OrekitMessagesTest" test class for the Orekit library. The "org.apache.commons.math3.analysis.interpolation.HermiteInterpolator" has been imported from the Orekit space flight dynamics library.

The Orekit library is described at:
https://www.orekit.org/forge/projects/orekit
The original files are distributed under the terms of the Apache 2 license which is: Copyright 2010 CS Communication & Systèmes

1.2572 jetty-xml 9.4.41.v20210516
1.2572.1 Available under license :
This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution.
No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses
to its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other
intellectual property rights of any other entity. Each Contributor
discerns any liability to Recipient for claims brought by any other
entity based on infringement of intellectual property rights or
otherwise. As a condition to exercising the rights and licenses granted
hereunder, each Recipient hereby assumes sole responsibility to secure
any other intellectual property rights needed, if any. For example, if a
third party patent license is required to allow Recipient to distribute
the Program, it is Recipient's responsibility to acquire that license
before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient
copyright rights in its Contribution, if any, to grant the copyright
license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under
its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties
      and conditions, express and implied, including warranties or
      conditions of title and non-infringement, and implied warranties or
      conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for
      damages, including direct, indirect, special, incidental and
      consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are
      offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such
      Contributor, and informs licensees how to obtain it in a reasonable
      manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained
within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.
4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in the
Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in
any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[[]]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message
The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2573 parseuri 0.0.5
1.2573.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2574 extglob 2.0.4
1.2574.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2575 yargs 13.2.4

1.2575.1 Available under license:

Copyright 2010 James Halliday (mail@substack.net)
Modified work Copyright 2014 Contributors (ben@npmjs.com)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2576 z-schema 3.25.1

1.2576.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Martin Zagora and other contributors
https://github.com/zaggino/z-schema/graphs/contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.2577 nss 3.28.4-0ubuntu0.16.04.10

1.2577.1 Available under license:

NSS is available under the Mozilla Public License, version 2, a copy of which is below.

Note on GPL Compatibility
-------------------------

The MPL 2, section 3.3, permits you to combine NSS with code under the GNU General Public License (GPL) version 2, or any later version of that license, to make a Larger Work, and distribute the result under the GPL. The only condition is that you must also make NSS, and any changes you have made to it, available to recipients under the terms of the MPL 2 also.

Anyone who receives the combined code from you does not have to continue to dual licence in this way, and may, if they wish, distribute under the terms of either of the two licences - either the MPL alone or the GPL alone. However, we discourage people from distributing copies of NSS under the GPL alone, because it means that any improvements they make cannot be reincorporated into the main version of NSS. There is never a need to do this for license compatibility reasons.

Note on LGPL Compatibility
--------------------------

The above also applies to combining MPLed code in a single library with code under the GNU Lesser General Public License (LGPL) version 2.1, or any later version of that license. If the LGPLed code and the MPLed code are not in the same library, then the copyleft coverage of the two licences does not overlap, so no issues arise.

Mozilla Public License Version 2.0
==================================

1. Definitions
--------------

1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version" means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution" means Covered Software of a particular Contributor.

1.4. "Covered Software" means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses" means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form" means any form of the work other than Source Code Form.

1.7. "Larger Work" means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License" means this document.

1.9. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications" means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License" means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form" means the form of the work preferred for making modifications.

1.14. "You" (or "Your") means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.
2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.
2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

-------------------

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients’ rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices
You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.
5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

**************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is" basis, *
*  without warranty of any kind, either expressed, implied, or statutory, *
*  including, without limitation, warranties that the Covered Software *
*  is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.
*                                                                      *
**************************************************************************
**************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*                                                                      *
*  Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the
8. Litigation

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. Any law or regulation which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section
10.3, no one other than the license steward has the right to modify or
publish new versions of this License. Each version will be given a
distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version
of the License under which You originally received the Covered Software,
or under the terms of any subsequent version published by the license
steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to
create a new license for such software, you may create and use a
modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice
---------------------------------------------

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice
---------------------------------------------------------

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Within this directory, each of the file listed below is licensed under
the terms given in the file LICENSE-MPL, also in this directory.

pi1k.txt
pi2k.txt
pi5k.txt

Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

This Source Code Form is subject to the terms of the Mozilla Public
License, v. 2.0. If a copy of the MPL was not distributed with this
file, You can obtain one at http://mozilla.org/MPL/2.0/.
Within this directory, each of the file listed below is licensed under
the terms given in the file LICENSE-MPL, also in this directory.

PRIMES
Within this directory, each of the file listed below is licensed under
the terms given in the file LICENSE-MPL, also in this directory.

basecvt.pod
gcd.pod
invmod.pod
isprime.pod
lap.pod
mpi-test.pod
prime.txt
prng.pod

# This file contains a list of people who've made non-trivial
# contribution to the Google C++ Testing Framework project. People
# who commit code to the project are encouraged to add their names
# here. Please keep the list sorted by first names.

Ajay Joshi <jaj@google.com>
Balzs Dn <balazs.dan@gmail.com>
Bharat Mediratta <bharat@menalto.com>
1.2578 is-fullwidth-code-point 2.0.0

1.2578.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2579 intl-relativeformat 2.1.0 2.1.0
1.2579.1 Available under license :

Copyright 2014 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---------------------------------------------------------------------
Pluralization rules built from
https://github.com/papandreou/node-cldr
which is licensed under the BSD license and has the following license:

Copyright (c) 2012, Andreas Lind Petersen
All rights reserved.
1.2580 babel-plugin-proposal-optional-chaining 7.13.12
1.2580.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2581 @babel/plugin-transform-dotall-regex
7.12.13
1.2581.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2582 jsreport-static-pdf 0.4.0
1.2582.1 Available under license:
MIT License

Copyright (c) 2019 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2583 is-binary-path 1.0.1
1.2583.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2584 apache-log4j 2.8.2

1.2584.1 Available under license:

Apache Log4j Core
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2585 listenablefuture 1.0
1.2585.1 Available under license :
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2007 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1012932365_1591900514.54/0/listenablefuture-1-0-sources-
jar/com/google/common/util/concurrent/ListenableFuture.java

1.2586 aspectj-tools 1.8.13

1.2587 ajv 6.10.0

1.2587.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
1.2588 verror 1.10.0

1.2588.1 Available under license:

Copyright (c) 2016, Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2589 esquery 1.3.1

1.2589.1 Available under license:

Copyright (c) 2013, Joel Feenstra

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the ESQuery nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JOEL FEESTRA BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2590 superagent 3.8.3 3.8.3

1.2590.1 Available under license:
(The MIT License)

Copyright (c) 2014-2016 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2591 babel-plugin-syntax-top-level-await

7.12.13

1.2591.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2592 isemail 1.2.0 1.2.0

1.2592.1 Available under license:

Copyright 2008-2011, Dominic Sayers
Copyright 2013-2014, GlobeSherpa
Copyright 2014-2015, Eli Skeggs

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Dominic Sayers nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2593 cookie 0.4.0
1.2593.1 Available under license :
(The MIT License)

Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2594 guava 25.0-jre
1.2594.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright-Only Dedication* (based on United States law)  
or Public Domain Certification
This license is acceptable for Free Cultural Works.
Creative Commons has retired this legal tool and does not recommend that it be applied to works.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any
copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

/*
 * This method was written by Doug Lea with assistance from members of JCP
 * JSR-166 Expert Group and released to the public domain, as explained at
 * http://creativecommons.org/licenses/publicdomain
 * As of 2010/06/11, this method is identical to the (package private) hash
 * method in OpenJDK 7's java.util.HashMap class.
 *
 */

/*
 * Copyright (C) 2010 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

1.2595 ono 4.0.11 4.0.11
1.2595.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 James Messinger

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2596 chalk 2.4.1
1.2596.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2597 jersey-media-json-jackson 2.25.1
1.2597.1 Available under license :
CDDL+GPL License
http://glassfish.java.net/public/CDDL+GPL_1_1.html
1.2598 electron-to-chromium 1.3.690

1.2598.1 Available under license:
Copyright 2018 Kilian Valkhof

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2599 @babel/helper-plugin-utils 7.13.0

1.2599.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2600 ipaddr.js 1.8.0 1.8.0

**1.2600.1 Available under license:**
Copyright (C) 2011-2017 whitequark <whitequark@whitequark.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2601 commander 4.1.1

**1.2601.1 Available under license:**
No license file was found, but licenses were detected in source scan.

# Permission is hereby granted, free of charge, to any person obtaining
# a copy of this software and associated documentation files (the
# "Software"), to deal in the Software without restriction, including
# without limitation the rights to use, copy, modify, merge, publish,
# distribute, sublicense, and/or sell copies of the Software, and to
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1163925133_1620933582.81/0/commander-4-1-1-gem/data-tar-gz/lib/commander.rb

No license file was found, but licenses were detected in source scan.

= Commander

The complete solution for Ruby command-line executables. Commander bridges the gap between other terminal related libraries you know and love (OptionParser, HighLine), while providing many new features, and an elegant API.
== Features

* Easier than baking cookies
* Parses options using OptionParser
* Auto-populates struct with options ( no more { |v| options[:recursive] = v } )
* Auto-generates help documentation via pluggable help formatters
* Optional default command when none is present
* Global / Command level options
* Packaged with two help formatters (Terminal, TerminalCompact)
* Imports the highline gem for interacting with the terminal
* Adds additional user interaction functionality
* Highly customizable progress bar with intuitive, simple usage
* Multi-word command name support such as 'drupal module install MOD', rather than 'drupal module_install MOD'
* Sexy paging for long bodies of text
* Support for MacOS text-to-speech
* Command aliasing (very powerful, as both switches and arguments can be used)
* Growl notification support for MacOS
* Use the 'commander' executable to initialize a commander driven program

== Installation

$ gem install commander

== Quick Start

To generate a quick template for a commander app, run:

$ commander init yourfile.rb

== Example

For more option examples view the Commander::Command#option method. Also an important feature to note is that action may be a class to instantiate, as well as an object, specifying a method to call, so view the RDoc for more information.

    require 'rubygems'
    require 'commander/import'

    # :name is optional, otherwise uses the basename of this executable
    program :name, 'Foo Bar'
    program :version, '1.0.0'
    program :description, 'Stupid command that prints foo or bar.'

    command :foo do [c]
        c.syntax = 'foobar foo'
        c.description = 'Displays foo'
    end
c.action do |args, options|
  say 'foo'
end
end

command :bar do |c|
c.syntax = 'foobar bar [options]' 
c.description = 'Display bar with optional prefix and suffix'
c.option '--prefix STRING', String, 'Adds a prefix to bar'
c.option '--suffix STRING', String, 'Adds a suffix to bar'
c.action do |args, options|
  options.default :prefix => '(', :suffix => ')
  say "#{options.prefix}bar#{options.suffix}"'
end
end

$ foobar bar
# => (bar)

$ foobar bar --suffix '{' --prefix '}'
# => {bar}

== HighLine

As mentioned above the highline gem is imported into 'global scope', below are some quick examples for how to utilize highline in your command(s):

# Ask for password masked with '*' character
ask("Password: ") { |q| q.echo = "*" }

# Ask for password
ask("Password: ") { |q| q.echo = false }

# Ask if the user agrees (yes or no)
agree("Do something?")

# Asks on a single line (note the space after ':')
ask("Name: ")

# Asks with new line after "Description:"  
ask("Description:")

# Calls Date#parse to parse the date string passed
ask("Birthday? ", Date)

# Ensures Integer is within the range specified
ask("Age? ", Integer) { |q| q.in = 0..105 }
# Asks for a list of strings, converts to array
ask("Fav colors?", Array)

== HighLine & Interaction Additions

In addition to highline's fantastic choice of methods, commander adds the following methods to simplify common tasks:

# Ask for password
password

# Ask for password with specific message and mask character
password "Enter your password please;", '-'

# Ask for CLASS, which may be any valid class responding to #parse. Date, Time, Array, etc
names = ask_for_array 'Names: '
bday = ask_for_date 'Birthday?: '

# Simple progress bar (Commander::UI::ProgressBar)
uris = %w[
  http://vision-media.ca
  http://google.com
  http://yahoo.com
]
progress uris do |uri|
  res = open uri
  # Do something with response
end

# 'Log' action to stdout
log "create", "path/to/file.rb"

# Enable paging of output after this point
enable_paging

# Ask editor for input (EDITOR environment variable or whichever is available: TextMate, vim, vi, emacs, nano, pico)
ask_editor

# Ask editor, supplying initial text
ask_editor 'previous data to update'

# Ask editor, preferring a specific editor
ask_editor 'previous data', 'vim'

# Choose from an array of elements
choice = choose("Favorite language?", :ruby, :perl, :js)
# Alter IO for the duration of the block
io new_input, new_output do
  new_input_contents = $stdin.read
  puts new_input_contents # outputs to new_output stream
end
# $stdin / $stdout reset back to original streams

# Speech synthesis
speak 'What is your favorite food?'
food = ask 'favorite food?:'
speak "Wow, I like #{food} too. We have so much in common."

# Execute arbitrary applescript
applescript 'foo'

# Converse with speech recognition server
case converse 'What is the best food?', :cookies => 'Cookies', :unknown => 'Nothing'
  when :cookies
    speak 'o.m.g. you are awesome!'
  else
    case converse 'That is lame, shall I convince you cookies are the best?', :yes => 'Ok', :no => 'No', :maybe =>
      'Maybe another time'
      when :yes
        speak 'Well you see, cookies are just fantastic, they melt in your mouth.'
      else
        speak 'Ok then, bye.'
      end
  end
end

== Growl Notifications

Commander provides methods for displaying Growl notifications. To use these methods you need to install http://github.com/visionmedia/growl which utilizes the growlnotify[http://growl.info/extras.php#growlnotify] executable. Note that growl is auto-imported by Commander when available, no need to require.

# Display a generic Growl notification
notify 'Something happened'

# Display an 'info' status notification
notify_info 'You have #{emails.length} new email(s)'

# Display an 'ok' status notification
notify_ok 'Gems updated'

# Display a 'warning' status notification
notify_warning '1 gem failed installation'
# Display an 'error' status notification
notify_error "Gem #{name} failed"

growlnotify can also be installed via homebrew[http://mxcl.github.com/homebrew/]:

$ brew install growlnotify

== Commander Goodies

### Option Defaults

The options struct passed to #action provides a #default method, allowing you to set defaults in a clean manner for options which have not been set.

```ruby
command :foo do |c|
  c.option '--interval SECONDS', Integer, 'Interval in seconds'
  c.option '--timeout SECONDS', Integer, 'Timeout in seconds'
  c.action do [args, options]
    options.default [:interval => 2, :timeout => 60]
  end
end
```

### Command Aliasing

Aliases can be created using the #alias_command method like below:

```ruby
command :'install gem' do |c|
  c.action { puts 'foo' }
end
alias_command :'gem install', :'install gem'
```

Or more complicated aliases can be made, passing any arguments as if it was invoked via the command line:

```ruby
command :'install gem' do |c|
  c.syntax = 'install gem <name> [options]'[
  c.option '--dest DIR', String, 'Destination directory'
  c.action { |args, options| puts "installing #{args.first} to #{options.dest}" }
end
alias_command :update, :'install gem', 'rubygems', '--dest', 'some_path'
```

$ foo update
# => installing rubygems to some_path

### Command Defaults
Although working with a command executable framework provides many benefits over a single command implementation, sometimes you still want the ability to create a terse syntax for your command. With that in mind we may use #default_command to help with this. Considering our previous :'install gem' example:

```ruby
default_command :update

$ foo
# => installing rubygems to some_path
```

Keeping in mind that commander searches for the longest possible match when considering a command, so if you were to pass arguments to foo like below, expecting them to be passed to :update, this would be incorrect, and would end up calling :'install gem', so be careful that the users do not need to use command names within the arguments.

```ruby
$ foo install gem
# => installing to
```

### Additional Global Help

Arbitrary help can be added using the following #program symbol:

```ruby
program :help, 'Author', 'TJ Holowaychuk <tj@vision-media.ca>'
```

Which will output the rest of the help doc, along with:

```
AUTHOR:

TJ Holowaychuk <tj@vision-media.ca>
```

### Global Options

Although most switches will be at the command level, several are available by default at the global level, such as --version, and --help. Using #global_option you can add additional global options:

```ruby
global_option('-c', '--config FILE', 'Load config data for your commands to use') { |file| ... }
```

This method accepts the same syntax as Commander::Command#option so check it out for documentation.

All global options regardless of providing a block are accessible at the command level. This means that instead of the following:

```ruby
global_option('--verbose') { $verbose = true }
...
```

```ruby
c.action do |args, options|

```
say 'foo' if $verbose
...

You may:

global_option '--verbose'
...
c.action do [args, options]
say 'foo' if options.verbose
...

=== Formatters

Two core formatters are currently available, the default Terminal formatter as well as TerminalCompact. To utilize a different formatter simply use :help_formatter like below:

program :help_formatter, Commander::HelpFormatter::TerminalCompact

Or utilize the help formatter aliases:

program :help_formatter, :compact

This abstraction could be utilized to generate HTML documentation for your executable.

== Tips

When adding a global or command option, OptionParser implicitly adds a small switch even when not explicitly created, for example -c will be the same as --config in both examples, however `-c' will only appear in the documentation when explicitly assigning it.

global_option '-c', '--config FILE'
global_option '--config FILE'

== ASCII Tables

For feature rich ASCII tables for your terminal app check out visionmedia's terminal-table gem at http://github.com/visionmedia/terminal-table

+----------+-------+----+--------+-----------------------+
| Terminal | Table | Is | Wicked | Awesome               |
+----------+-------+----+--------+-----------------------+
|          |       |    |        | get it while its hot! |
+----------+-------+----+--------+-----------------------+

== Running Specifications

$ rake spec
OR

$ spec --color spec

== Contrib

Feel free to fork and request a pull, or submit a ticket
http://github.com/visionmedia/commander/issues

== License

(The MIT License)

Copyright (c) 2008-2009 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):
* /opt/cola/permits/1163925133_1620933582.81/0/commander-4-1-1-gem/data-tar-gz/README.rdoc

1.2602 deep-eql 3.0.1

1.2602.1 Available under license:

Copyright (c) 2013 Jake Luer <jake@alogicalparadox.com> (http://alogicalparadox.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2603 jersey-guice 1.19

1.2604 raw-body 2.3.2 2.3.2
1.2604.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013-2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2605 drill-storage-http 1.19.0

1.2605.1 Available under license:

Drill : Contrib : Storage : HTTP
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same “printed page” as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2606 wrappy 1.0.2 1.0.2
1.2606.1 Available under license :

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2607 memo-parser 0.2.1
1.2607.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Ben Mosher
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2608 streamifier 0.1.1

1.2608.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Gabriel Llamas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2609 object.pick 1.3.0
1.2609.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2610 kerby-config 1.0.0
1.2610.1 Available under license:

Kerby Config
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.2611 yargs-unparser 1.6.0
1.2611.1 Available under license :
   The MIT License (MIT)

   Copyright (c) 2017 Made With MOXY Ltda <hello@moxy.studio>

   Permission is hereby granted, free of charge, to any person obtaining a copy
   of this software and associated documentation files (the "Software"), to deal
   in the Software without restriction, including without limitation the rights
   to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
   copies of the Software, and to permit persons to whom the Software is
   furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in
   all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
   IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
   FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
   AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
   LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
   OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
   THE SOFTWARE.

1.2612 drill-udfs 1.19.0
1.2612.1 Available under license :

   Drill : Contrib : UDFs
   Copyright 2021 The Apache Software Foundation

   This product includes software developed at
   The Apache Software Foundation (http://www.apache.org/).

   Apache License
   Version 2.0, January 2004
   http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.2613 call-bind 1.0.2 1.0.2

1.2613.1 Available under license:

MIT License

Copyright (c) 2020 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2614 debug 2.6.9 2.6.9

1.2614.1 Available under license:

(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, 
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN 
CONNECTION WITH THE 
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2615 is-regex 1.1.2

1.2615.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of 
this software and associated documentation files (the "Software"), to deal in 
the Software without restriction, including without limitation the rights to 
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of 
the Software, and to permit persons to whom the Software is furnished to do so, 
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all 
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR 
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS 
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR 
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER 
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN 
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2616 estraverse 4.3.0

1.2616.1 Available under license :
Redistribution and use in source and binary forms, with or without 
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright 
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright 
  notice, this list of conditions and the following disclaimer in the 
  documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" 
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE 
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE 
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY 
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2617 dont-sniff-mimetype 1.0.0 1.0.0

1.2617.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014, 2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2618 buffer 5.6.0

1.2618.1 Available under license :
The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh, and other contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2619 is-extendable 1.0.1

1.2619.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2620 snapdragon 0.8.2

1.2620.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2621 pango 1.38.1-1

1.2621.1 Available under license :

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General
Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation
and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

d) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the source code distributed need not include anything that is normally
distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:
a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY
15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
This package was debianized by Akira TAGOH <tagoh@debian.org> on Sun, 10 Mar 2002 22:30:33 +0900.

It was downloaded from:

Upstream Authors:
Owen Taylor <otaylor@redhat.com>

Robert Brady (Indic shapers), Hans Breuer (Windows backend), Matthias Clasen (Documentation), Sivaraj Doddannan (Tamil shaper), Dov Grobgeld (Hebrew shaper), Karl Koehler (Arabic shaper), Alex Larsson (FreeType and Windows backends), Tor Lillqvist (FreeType and Windows backends), Changwoo Ryu (Hangul shaper), Havoc Pennington, Roozbeh Pournader (Arabic shaper), Chookij Vanatham (Hebrew shaper).

Copyright:
This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Lesser General Public License for more details.
License along with this package; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

See /usr/share/common-licenses/LGPL-2 on your debian system.

The directory pango/mini-fribidi is copyright 1999,2000 Dov Grobgeld,
2001,2002 Behdad Esfahbod and is distributed under the following terms:

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License
along with this library, in a file named COPYING; if not, write to the
Free Software Foundation, Inc., 59 Temple Place, Suite 330,
Boston, MA 02111-1307, USA

On Debian systems the full text of the GNU Lesser General Public License can
be found in the `/usr/share/common-licenses/LGPL-2.1' file.

The file tests/GraphemeBreakTest.txt is copyright 1991-2014 Unicode, Inc and
is distributed under the following terms:

Copyright 1991-2014 Unicode, Inc. All rights reserved. Distributed under
the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy
of the Unicode data files and any associated documentation (the "Data Files")
or Unicode software and any associated documentation (the "Software") to deal
in the Data Files or Software without restriction, including without
limitation the rights to use, copy, modify, merge, publish, distribute,
and/or sell copies of the Data Files or Software, and to permit persons to
whom the Data Files or Software are furnished to do so, provided that (a) the
above copyright notice(s) and this permission notice appear with all copies
of the Data Files or Software, (b) both the above copyright notice(s) and
this permission notice appear in associated documentation, and (c) there is
clear notice in each modified Data File or in the Software as well as in the
documentation associated with the Data File(s) or Software that the data or
software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

### 1.2622 jackson-jaxrs 2.9.10

#### 1.2622.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2625 batik-i18n 1.13
1.2625.1 Available under license:

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1135872594_1613620755.73/0/batik-i18n-1-13-sources-1-jar/META-INF/maven/org.apache.xmlgraphics/batik-i18n/pom.xml
No license file was found, but licenses were detected in source scan.

/
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1135872594_1613620755.73/0/batik-i18n-1-13-sources-1.jar/org/apache/batik/i18n/Localizable.java
* /opt/cola/permits/1135872594_1613620755.73/0/batik-i18n-1-13-sources-1.jar/org/apache/batik/i18n/LocalizableSupport.java
* /opt/cola/permits/1135872594_1613620755.73/0/batik-i18n-1-13-sources-1.jar/org/apache/batik/i18n/LocaleGroup.java
* /opt/cola/permits/1135872594_1613620755.73/0/batik-i18n-1-13-sources-1.jar/org/apache/batik/i18n/ExtendedLocalizable.java

1.2626 metrics-jvm 4.1.0

1.2627 okio 2.8.0
1.2627.1 Available under license :

No license file was found, but licenses were detected in source scan.

/
* Copyright (C) 2018 Square, Inc.
* 
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
Opened Source Used In DNAC Platform Antelope 10420

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/ByteString.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/-Utf8.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/-Util.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/-Platform.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/-Platform.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/ByteString.kt

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/RealBufferedSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/SegmentedByteString.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/internal/RealBufferedSource.kt

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Source.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Buffer.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Okio.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/BufferedSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/BufferedSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Sink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Internal/Buffer.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Timeout.kt

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/PackSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedOkio.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedUpgrade.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Throttler.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/-DeprecatedUtf8.kt

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingTimeout.kt
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 Square Inc.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/ByteString.kt
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2017 Square, Inc.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Utf8.kt
No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 * *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/-Base64.kt
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2016 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/HashingSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/MemorySink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/HashingSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Options.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Pipe.kt

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2014 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License”);
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/GzipSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/InflaterSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/AsyncTimeout.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Source.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/SegmentPool.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/SegmentPool.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/BufferedSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/commonMain/okio/Segment.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Sink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/JvmOkio.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Buffer.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Timeout.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/ForwardingSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/DeflaterSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/Deflater.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/GzipSink.kt
* /opt/cola/permits/1096821398_1601900245.47/0/okio-2-8-0-sources-jar/jvmMain/okio/BufferedSink.kt
1.2628 *es6-weak-map 2.0.3*

1.2628.1 Available under license:

**ISC License**

Copyright (c) 2013-2018, Mariusz Nowak, @medikoo, medikoo.com

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2629 *@babel/cli 7.13.14*

1.2629.1 Available under license:

**MIT License**

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2630 @babel/plugin-transform-destructuring 7.13.0

1.2630.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2631 babel-plugin-transform-exponentiation-operator 6.24.1

1.2631.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights
from parties other than Ecma International), including patent rights, and no licenses under such third party rights
are granted under this license even if the third party concerned is a member of Ecma International. SEE THE
ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.
MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2632 hosted-git-info 2.8.8
1.2632.1 Available under license:

Copyright (c) 2015, Rebecca Turner

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

1.2633 json-schema-traverse 0.4.1
1.2633.1 Available under license :
MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2634 minimist 1.2.5
1.2634.1 Available under license :
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2635 babel-plugin-syntax-exponentiation-operator 6.13.0

1.2635.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2636 v8flags 2.1.1
1.2636.1 Available under license :

Copyright (c) 2014 Tyler Kellen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2637 inflight 1.0.6 1.0.6
1.2637.1 Available under license :

The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.2639.1 Available under license:

---

Open Source Used In DNAC Platform Antelope 10436

---
1.2640 type-check 0.3.2 0.3.2

1.2640.1 Available under license:

Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2641 underscore 1.13.1 1.13.1

1.2641.1 Available under license:

Copyright (c) 2009-2021 Jeremy Ashkenas, Julian Gonggrijp, and DocumentCloud and Investigative Reporters & Editors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
1.2642 json-schema 0.2.3

1.2642.1 Available under license:

No license file was found, but licenses were detected in source scan.

/**
 * JSONSchema Validator - Validates JavaScript objects using JSON Schemas
 *(http://www.json.com/json-schema-proposal/)
 *
 * Copyright (c) 2007 Kris Zyp SitePen (www.sitepen.com)
 * Licensed under the MIT (MIT-LICENSE.txt) license.
 *
 * To use the validator call the validate function with an instance object and an optional schema object.
 * If a schema is provided, it will be used to validate. If the instance object refers to a schema (self-validating),
 * that schema will be used to validate and the schema parameter is not necessary (if both exist,
 * both validations will occur).
 * The validate method will return an array of validation errors. If there are no errors, then an
 * empty list will be returned. A validation error will have two properties:
 * "property" which indicates which property had the error
 * "message" which indicates what the error was
 */

Found in path(s):
* /opt/cola/permits/1173677558_1623725378.39/0/kriszyp-json-schema-v0-2-3-0-g07ae2c6-1-tar-gz/kriszyp-json-schema-07ae2c6/lib/validate.js

No license file was found, but licenses were detected in source scan.

/**
 * JSON Schema link handler
 * Copyright (c) 2007 Kris Zyp SitePen (www.sitepen.com)
 * Licensed under the MIT (MIT-LICENSE.txt) license.
 */

Found in path(s):
* /opt/cola/permits/1173677558_1623725378.39/0/kriszyp-json-schema-v0-2-3-0-g07ae2c6-1-tar-gz/kriszyp-json-schema-07ae2c6/lib/links.js

No license file was found, but licenses were detected in source scan.

JSON Schema is a repository for the JSON Schema specification, reference schemas and a CommonJS
implementation of JSON Schema (not the only JavaScript implementation of JSON Schema, JSV is another
elegant JavaScript validator).
Code is licensed under the AFL or BSD license as part of the Persevere
project which is administered under the Dojo foundation,
and all contributions require a Dojo CLA.

Found in path(s):
* /opt/cola/permits/1173677558_1623725378.39/0/kriszyp-json-schema-v0-2-3-0-g07ae2c6-1-tar-gz/kriszyp-json-
schema-07ae2c6/README.md

1.2643 @babel/helper-optimise-call-expression 7.12.13

1.2643.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2644 javax-ws-rs-api 2.0.1

1.2644.1 Available under license :

Found license 'General Public License 2.0' in
* Copyright (c) 2011-2013 Oracle and/or its affiliates. All rights
reserved. * The contents of this file are subject to the terms of either the GNU * General Public License Version 2
only ("GPL") or the Common Development * and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can * Oracle designates this particular file as
subject to the "Classpath" * exception as provided by Oracle in the GPL Version 2 section of the License’
subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License. Found license 'General Public License 2.0' in * Copyright (c) 2012 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License. Found license 'General Public License 2.0' in * Copyright (c) 2011-2014 Oracle and/or its affiliates. All rights reserved. * The contents of this file are subject to the terms of the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can * Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License.

1.2645 optional-require 1.0.3 1.0.3

1.2645.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation
or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by
combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

a. You must give any other recipients of the Work or Derivative Works a copy of this License; and

b. You must cause any modified files to carry prominent notices stating that You changed the files; and

c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any
Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
1.2646 jersey 2.5.1
1.2646.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of
the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under
common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause
the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty
percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims,
the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use,
reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or
without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use,
practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or
otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the
Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the
combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims,
each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce,
modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions
thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger
Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor
either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell,
offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions
thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or
portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or
otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has
deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the
Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.
6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License
is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

----------
NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to
make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish),
that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new
free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to
surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the
software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all
the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show
them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you
legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no
warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to
know that what they have is not the original, so that any problems introduced by others will not reflect on the
original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that
redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To
prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying
it may be distributed under the terms of this General Public License. The "Program", below, refers to any such
program or work, and a "work based on the Program" means either the Program or any derivative work under
copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications
and/or translated into another language. (Hereinafter, translation is included without limitation in the term
"modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its
scope. The act of running the Program is not restricted, and the output from the Program is covered only if its
contents constitute a work based on the Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium,
provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and
disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and
give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty
protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted
interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.
To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL
Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

/*
 * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
 *
 * Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
 *
 * The contents of this file are subject to the terms of either the GNU
 * General Public License Version 2 only ("GPL") or the Common Development
 * and Distribution License("CDDL") (collectively, the "License"). You
 * may not use this file except in compliance with the License. You can
 * obtain a copy of the License at
 * http://glassfish.java.net/public/CDDL+GPL_1_1.html
 * or packager/legal/LICENSE.txt. See the License for the specific
 * language governing permissions and limitations under the License.
 *
 * When distributing the software, include this License Header Notice in each
 * file and include the License file at packager/legal/LICENSE.txt.
 *
 * GPL Classpath Exception:
 * Oracle designates this particular file as subject to the "Classpath"
 * exception as provided by Oracle in the GPL Version 2 section of the License
 * file that accompanied this code.
 *
 * Modifications:
 * If applicable, add the following below the License Header, with the fields
 * enclosed by brackets [] replaced by your own identifying information:
 * "Portions Copyright [year] [name of copyright owner]"
 *
 * Contributor(s):
 * If you wish your version of this file to be governed by only the CDDL or
 * only the GPL Version 2, indicate your decision by adding "[Contributor]
 * elects to include this software in this distribution under the [CDDL or GPL
 * Version 2] license." If you don't indicate a single choice of license, a
 * recipient has the option to distribute your version of this file under
 * either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to such option by the copyright
* holder.
*/

DO NOT TRANSLATE OR LOCALIZE.

%% The following software may be included in this product: ASM
Use of any of this software is governed by the terms of the license below:

Copyright (c) 2000-2005 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

%% The following software may be included in this product: Jettison
Use of any of this software is governed by the terms of the license below:

Copyright 2006 Envoi Solutions LLC
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

------------------------------------------------------------------------

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of
fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source
code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form,
including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as
indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix
below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the
Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole,
an original work of authorship. For the purposes of this License, Derivative Works shall not include works that
remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works
thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and

2. You must cause any modified files to carry prominent notices stating that You changed the files; and

3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute,
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you
may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
limitations under the License.

%%% The following software may be included in this product: Jackson
Use of any of this software is governed by the terms of the license below:

Jackson is dual-licensed under two alternative popular Open Source licenses: Apache (AL 2.0) and Gnu Lesser GPL
(LGPL 2.1). You choose one or the other, as necessary (if you want to redistribute the code for use, you do not need
license), and abide by the license rules as defined by the respective license agreement (and only that one).

-----------------------------------------------

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or
indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of
fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source
code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form,
including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as
indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix
below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and

2. You must cause any modified files to carry prominent notices stating that You changed the files; and

3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a
NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2647 should-type-adaptors 1.1.0 1.1.0

1.2647.1 Available under license :
MIT License

Copyright (c) 2016 Should.js assertion library

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2648 @babel/preset-env 7.13.12

1.2648.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2649 estraverse 5.2.0
1.2649.1 Available under license :
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2650 should-format 3.0.3 3.0.3
1.2650.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2651 es6-promisify 5.0.0

1.2651.1 Available under license :

No license file was found, but licenses were detected in source scan.

Published under the [MIT License](http://opensource.org/licenses/MIT).

Found in path(s):
* /opt/cola/permits/1110637349_1612225538.25/0/digitaldesignlabs-es6-promisify-v5-0-0-0-g7eb2f5e-1-targ-gz/digitaldesignlabs-es6-promisify-6c15842/README.md

1.2652 lm-sensors 3.4.0-2

1.2652.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.
The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.
In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a “work that uses the Library” with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable “work that
uses the Library”, as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE libraries TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Here is a list of the main contributors to lm-sensors version 3.

* Frodo Looijaard
  Original author of libsensors, sensors-detect, sensors and isadump.
* Merlin Hughes
  Original author of sensord.
* Bob Schlaermann
  Dynamic chip feature detection (a.k.a. generic chip support) in libsensors and sensors.
* Mark M. Hoffman
  Many improvements to the libsensors configuration file scanner.
* Jean Delvare
  New libsensors API, and migration of sensors and sensord thereto.
  Many optimizations in libsensors and sensors.
  Configuration file converter.
  Rewrite of sensors-detect.
  Support for multiple configuration files in libsensors.

1.2653 helmet 2.3.0
1.2653.1 Available under license :
(The MIT License)

Copyright (c) 2012-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2654 jsreport-client 1.2.0
1.2654.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2655 babel-plugin-transform-es2015-spread
6.22.0
1.2655.1 Available under license :
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")]>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
   from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2656 orc-core 1.5.1

1.2656.1 Available under license:

ORC Core
Copyright 2013-2018 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2657 joda-time 2.9.1
1.2657.1 Available under license :
=============================================================================
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=============================================================================

This product includes software developed by
Joda.org (http://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2658 is-number 4.0.0

1.2658.1 Available under license :

   The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2659 junit 4.10
1.2659.1 Available under license :

Common Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.
"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient’s responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.
Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2660 hadoop-mapreduce-client-jobclient

3.2.2

1.2660.1 Available under license :

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:

* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of Gson 2.2.4, which has the following notices:

The Netty Project
=================
Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

-------------------------------------------------------------------------------
This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
  * HOMEPAGE:
    * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
  * HOMEPAGE:
    * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
  * HOMEPAGE:
    * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
  * HOMEPAGE:
    * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
  * HOMEPAGE:
    * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:
This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:
This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
  * HOMEPAGE:
    * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===============================================================================

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.

Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:

* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices:

* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
  Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
  TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of "Java Concurrency in Practice" book annotations 1.0, which has the following notices:

* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
  Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.
The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,
which has the following notices:

* ==============================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
* ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle
The following artifacts are CDDL + GPLv2 with classpath exception. https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions. http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.
org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache PureJavaCrc32C from apache-hadoop-common http://hadoop.apache.org/ (Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by "GCC RUNtime Library Exception" http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
== Contributors ==
* Tatu Saloranta
  * Providing benchmark suite
* Alec Wysoker
  * Performance and memory usage improvement

The binary distribution of this product bundles binaries of Xerces2 Java Parser 2.9.1,
which has the following notices:
* ==============================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xerces Java distribution. ==
 ==============================================================

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
  - voluntary contributions made by Paul Eng on behalf of the
    Apache Software Foundation that were originally developed at iClick, Inc.,
    software copyright (c) 1999.

The binary distribution of this product bundles binaries of Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0

--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting
it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was
relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of
Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library distributed under the terms of the Apache 2 license. Original source copyright:

Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5 (https://pypi.python.org/pypi/python-daemon/). Original source copyright:

# Copyright 2008-2015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 2007-2008 Robert Niederreiter, Jens Klein
# Copyright 2004-2005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrer
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of

Plexus Cipher: encryption/decryption Component 1.4,
which has the following notices:

* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

  The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
  which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of

software.amazon.ion:ion-java 1.0.1,
which has the following notices:

* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of

joda-time:joda-time:2.9.9
which has the following notices:

* = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

This product includes software developed by
Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of

Ehcache 3.3.1,
which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64,
LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
* Copyright (c) 2005, European Commission project OneLab under contract 034819 (http://www.one-lab.org)
* All rights reserved.
* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of the University Catholique de Louvain - UCL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor the names of its contributors may be used to endorse or promote
products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/
/*++
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 * Abstract: The main routine
 * --*/

For src/main/native/src/org/apache/hadoop/io/compress/lz4/{lz4.h,lz4.c,lz4hc.h,lz4hc.c},

/*
 LZ4 - Fast LZ compression algorithm
 Header File
 Copyright (C) 2011-2014, Yann Collet.
 BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at:
- LZ4 source repository: http://code.google.com/p/lz4/
- LZ4 public forum: https://groups.google.com/forum/#!forum/lz4c
*/

For hadoop-common-project/hadoop-common/src/main/native/gtest
---------------------------------------------------------------------

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The binary distribution of this product bundles these dependencies under the following license:

re2j 1.1

---------------------------------------------------------------------

(GO license)
This is a work derived from Russ Cox’s RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

---------------------------------------------------------------------

Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni
(https://github.com/fusesource/leveldbjni), which is available under the
following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of FuseSource Corp. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

============

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the
distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

--------------------------------------------------------------------------------

Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

--------------------------------------------------------------------------------

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

For:
    hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

---------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:
    hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
    hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
    hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
    hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
    bootstrap v3.3.6
    broccoli-asset-rev v2.4.2
    broccoli-funnel v1.0.1
    datatables v1.10.8
    em-helpers v0.5.13
    em-table v0.1.6
    ember v2.2.0
    ember-array-contains-helper v1.0.2
    ember-bootstrap v0.5.1
    ember-cli v1.13.13
    ember-cli-app-version v1.0.0
    ember-cli-babel v5.1.6
    ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

--------------------------------------------------------------------------------

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js

--------------------------------------------------------------------------------

The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js

The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery

Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
o! Algorithms - version 43.0

--------------------------------------------------------------------------------

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

For:

hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js

--------------------------------------------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

--------------------------------------------------------------------------------

D3 is available under a 3-clause BSD license. For details, see:

hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
HSQldb Database 2.3.4

(HSQL License)

"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ""AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQldb.ORG,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the
Hypersonic SQL Group."

The binary distribution of this product bundles these dependencies under the
following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
1. Definitions.

1.1 Contributor means each individual or entity
that creates or contributes to the creation of
Modifications.

1.2 Contributor Version means the combination of the
Original Software, prior Modifications used by a Contributor (if any), and the
Modifications made by that particular Contributor.

1.3 Covered Software means (a) the Original Software, or (b) Modifications, or (c) the
combination of files containing Original Software with files containing
Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.
2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer.
and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a)rename the license and remove any references to the name of the license steward (except to note that the
license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE,
SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its
utilization of rights under this License and You agree to work with Initial
Developer and Contributors to distribute such responsibility on an equitable
basis. Nothing herein is intended or shall be deemed to constitute any admission
of liability.

The binary distribution of this product bundles these dependencies under the
following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
gizzly-framework 2.2.21
gizzly-http 2.2.21
gizzly-http-server 2.2.21
gizzly-http-servlet 2.2.21
gizzly-rcm 2.2.21

-----------------------------------------------------------------------------

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or
contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software,
prior Modifications used by a Contributor (if any), and the Modifications made
by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b)
Modifications, or (c) the combination of files containing Original Software with
files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source
Code.
1.5. Initial Developer means the individual or entity that first makes
Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions
thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent
possible, whether at the time of the initial grant or subsequently acquired, any
and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the
following:
A. Any file that results from an addition to, deletion from or modification of
the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous
Modification; or
C. Any new file that is contributed or otherwise made available under the terms
of this License.

1.10. Original Software means the Source Code and Executable form of
computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter
acquired, including without limitation, method, process, and apparatus claims,
in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in
which modifications are made and (b) associated documentation included in or
with such code.
1.13. You (or Your) means an individual or a legal entity exercising
rights under, and complying with all of the terms of, this License. For legal
entities, You includes any entity which controls, is controlled by, or is
under common control with You. For purposes of this definition, control
means (a) the power, direct or indirect, to cause the direction or management of
such entity, whether by contract or otherwise, or (b) ownership of more than
fifty percent (50%) of the outstanding shares or beneficial ownership of such
entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to
third party intellectual property claims, the Initial Developer hereby grants
You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual
property rights (other than patent or trademark) Licensable by Initial
Developer, to use, reproduce, modify, display, perform, sublicense and
distribute the Original Software (or portions thereof), with or without
Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims
infringed by the making, using or selling of Original Software, to make, have
made, use, practice, sell, and offer for sale, and/or otherwise dispose of the
Original Software (or portions thereof).
(c) The licenses granted in Sections
2.1(a) and (b) are effective on the date Initial Developer first distributes or
otherwise makes the Original Software available to a third party under the terms
of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is
granted: (1) for code that You delete from the Original Software, or (2) for
infringements caused by: (i) the modification of the Original Software, or (ii)
the combination of the Original Software with other software or devices.
2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
   (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.
3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under
which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within
such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.  

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.  

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.  

7. LIMITATION OF LIABILITY.  

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.  

8. U.S. GOVERNMENT END USERS.  

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.  

9. MISCELLANEOUS.  

This License represents the complete agreement concerning
subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:

Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04
-------------------------------------------------------------------------------
By obtaining, using and/or copying this work, you (the licensee) agree that you
have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation,
with or without modification, for any purpose and without fee or royalty is
hereby granted, provided that you include the following on ALL copies of the
software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the
  redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and
  conditions. If none exist, the W3C Software Short Notice should be included
  (hypertext is preferred, text is permitted) within the body of any redistributed
  or derivative code.
- Notice of any changes or modifications to the files, including the date changes
  were made. (We recommend you provide URIs to the location from which the code is
derived.)

The binary distribution of this product bundles these dependencies under the
following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;
where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form.
This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.
When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.
Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.
4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively “Losses”) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY
CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.
This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

--------------------------------------------------------------------------------
(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

--------------------------------------------------------------------------------
(2-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Open Source Used In DNAC Platform Antelope 10554
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0
--------------------------------------------------------------------------------
(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast,
transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange).
royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.
5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable. No term or provision of this License shall be deemed waived and no breach
consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

jamon-runtime 2.4.1

--------------------------------------------------------------------------------
(MPL 2.0)
Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive
license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its
Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such
3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.
5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous
This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.
Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

________________________________________________________________________
/*--
Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the following license:

Hbase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions
a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography.
a work of drawing, painting, architecture, sculpture, engraving or
lithography; a photographic work to which are assimilated works
expressed by a process analogous to photography; a work of applied
art; an illustration, map, plan, sketch or three-dimensional work
relative to geography, topography, architecture or science; a
performance; a broadcast; a phonogram; a compilation of data to the
extent it is protected as a copyrightable work; or a work performed by
a variety or circus performer to the extent it is not otherwise
considered a literary or artistic work.
g. "You" means an individual or entity exercising rights under this
License who has not previously violated the terms of this License with
respect to the Work, or who has received express permission from the
Licensor to exercise rights under this License despite a previous
violation.
h. "Publicly Perform" means to perform public recitations of the Work and
to communicate to the public those public recitations, by any means or
process, including by wire or wireless means or public digital
performances; to make available to the public Works in such a way that
members of the public may access these Works from a place and at a
place individually chosen by them; to perform the Work to the public
by any means or process and the communication to the public of the
performances of the Work, including by public digital performance; to
broadcast and rebroadcast the Work by any means including signs,
sounds or images.
i. "Reproduce" means to make copies of the Work by any means including
without limitation by sound or visual recordings and the right of
fixation and reproducing fixations of the Work, including storage of a
protected performance or phonogram in digital form or other electronic
medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce,
limit, or restrict any uses free from copyright or rights arising from
limitations or exceptions that are provided for in connection with the
copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License,
Licensor hereby grants You a worldwide, royalty-free, non-exclusive,
perpetual (for the duration of the applicable copyright) license to
exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more
Collections, and to Reproduce the Work as incorporated in the
Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation,
including any translation in any medium, takes reasonable steps to
clearly label, demarcate or otherwise identify that changes were made
to the original Work. For example, a translation could be marked "The
original work was translated from English to Spanish," or a
modification could indicate "The original work has been modified.”;
c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You
must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the
applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in
compliance with Creative Commons' then-current trademark usage
guidelines, as may be published on its website or otherwise made
available upon request from time to time. For the avoidance of doubt,
this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.

For: hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs
/server/datanode/checker/AbstractFuture.java and
hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs
/server/datanode/checker/TimeoutFuture.java

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License. You may obtain a copy of
the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under
the License.

1.2661 strip-ansi 6.0.0 6.0.0

1.2661.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.2662 mkdirp 0.3.0

1.2662.1 Available under license:

Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2663 engine.io-parser 1.3.2 1.3.2

1.2663.1 Available under license:

(The MIT License)

Copyright (c) 2016 Guillermo Rauch (@rauchg)

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND.
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2664 es-to-primitive 1.2.1
1.2664.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2665 metrics-health-checks 4.1.0

1.2666 commons-math 2.2
1.2666.1 Available under license:
Apache Commons Math
Copyright 2001-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
The BracketFinder (package org.apache.commons.math.optimization.univariate) and PowellOptimizer (package org.apache.commons.math.optimization.general) classes are based on the Python code in module "optimize.py" (version 0.5) developed by Travis E. Oliphant for the SciPy library (http://www.scipy.org/)

The LinearConstraint, LinearObjectiveFunction, LinearOptimizer, RelationShip, SimplexSolver and SimplexTableau classes in package org.apache.commons.math.optimization.linear include software developed by Benjamin McCann (http://www.benmccann.com) and distributed with the following copyright: Copyright 2009 Google Inc.

This product includes software developed by the University of Chicago, as Operator of Argonne National Laboratory.
The LevenbergMarquardtOptimizer class in package org.apache.commons.math.optimization.general includes software translated from the lmder, lmpar and qrsolv Fortran routines from the Minpack package Minpack Copyright Notice (1999) University of Chicago. All rights reserved

The GraggBulirschStoerIntegrator class in package org.apache.commons.math.ode.nonstiff includes software translated from the odex Fortran routine developed by E. Hairer and G. Wanner.
Original source copyright:
Copyright (c) 2004, Ernst Hairer

The EigenDecompositionImpl class in package org.apache.commons.math.linear includes software translated from some LAPACK Fortran routines. Original source copyright:
Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

The MersenneTwister class in package org.apache.commons.math.random includes software translated from the 2002-01-26 version of the Mersenne-Twister generator written in C by Makoto Matsumoto and Takuji Nishimura. Original source copyright:
Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, All rights reserved

The complete text of licenses and disclaimers associated with the the original sources enumerated above at the time of code translation are in the LICENSE.txt
 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE COMMONS MATH DERIVATIVE WORKS:

The Apache commons-math library includes a number of subcomponents
whose implementation is derived from original sources written
in C or Fortran. License terms of the original sources
are reproduced below.

===============================================================================
For the lmder, lmpar and qrsolv Fortran routine from minpack and translated in
the LevenbergMarquardtOptimizer class in package
org.apache.commons.math.optimization.general
Original source copyright and license statement:

Minpack Copyright Notice (1999) University of Chicago. All rights reserved

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above
copyright notice, this list of conditions and the following
disclaimer.

2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials
provided with the distribution.

3. The end-user documentation included with the
redistribution, if any, must include the following
acknowledgment:

"This product includes software developed by the
University of Chicago, as Operator of Argonne National
Laboratory."
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.

5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE, EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGES.

===============================================================================
Copyright and license statement for the odex Fortran routine developed by E. Hairer and G. Wanner and translated in GraggBulirschStoerIntegrator class in package org.apache.commons.math.ode.nonstiff:

Copyright (c) 2004, Ernst Hairer

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===============================================================================

Copyright and license statement for the original lapack fortran routines translated in EigenDecompositionImpl class in package org.apache.commons.math.linear:

Copyright (c) 1992-2008 The University of Tennessee. All rights reserved.

$COPYRIGHT$

Additional copyrights may follow

$HEADERS$

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.

- Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
Copyright and license statement for the original Mersenne twister C
routines translated in MersenneTwister class in package
org.apache.commons.math.random:

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. The names of its contributors may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2667 jopt-simple 5.0.4
1.2667.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
The MIT License

Copyright (c) 2004-2015 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/AlternativeLongOptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/OptionParserState.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/UnconfiguredOptionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/AbstractOptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/OptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/internal/MethodInvokingValueConverter.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/OptionalArgumentOptionSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/util/DateConverter.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/util/RegexMatcher.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0-4-sources-8-jar/joptsimple/internal/ConstructorInvokingValueConverter.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/OptionArgumentConversionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/internal/Classes.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/UnavailableOptionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/OptionDescriptor.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/ValueConversionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/OptionParser.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/internal/Reflection.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/UnrecognizedOptionException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/internal/Messages.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/ArgumentList.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/NonOptionArgumentSpec.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/internal/AbbreviationMap.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/IllegalOptionSpecificationException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/internal/Columns.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/OptionSpecBuilder.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/OptionSpecMissingRequiredOptionsException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/internal/KeyValuePair.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/optionMissingRequiredArgumentException.java
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5.0.4-sources-8-jar/joptsimple/OptionSet.java

---

**Open Source Used In DNAC Platform Antelope** 10587
No license file was found, but licenses were detected in source scan.

/*
The MIT License

Copyright (c) 2004-2014 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1177945013_1625575335.76/0/jopt-simple-5-0-4-sources-8-jar/joptsimple/util/EnumConverter.java

1.2668 zero-allocation-hashing 0.9
1.2668.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 Higher Frequency Trading http://www.higherfrequencytrading.com
 * Licensed under the Apache License, Version 2.0 (the "License);
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/XxHash.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2014 Higher Frequency Trading http://www.higherfrequencytrading.com
~ Licensed under the Apache License, Version 2.0 (the "License);
~ you may not use this file except in compliance with the License.
~ You may obtain a copy of the License at
~ http://www.apache.org/licenses/LICENSE-2.0
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS,
~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
~ See the License for the specific language governing permissions and
~ limitations under the License.

Found in path(s):

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 Higher Frequency Trading http://www.higherfrequencytrading.com
 * Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/Access.java
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/Primitives.java
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/StringHash.java
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/ModernHotSpotStringHash.java
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/LongHashFunction.java
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/UnsafeAccess.java
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/CharSequenceAccess.java
* /opt/cola/permits/1202446137_1631568422.37/0/zero-allocation-hashing-0-9-sources-jar/net/openhft/hashing/ByteBufferAccess.java

1.2669 core-js 3.9.1
1.2669.1 Available under license:

Copyright (c) 2014-2021 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2670 netty-transport 4.1.42.Final

1.2670.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/MaxBytesRecvByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/DefaultMaxMessagesRecvByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/pool/FixedChannelPool.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/pool/ChannelPool.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/pool/ChannelPoolHandler.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/pool/AbstractChannelPoolHandler.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/pool/AbstractChannelPoolMap.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/pool/ChannelPoolMap.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/pool/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/pool/DefaultMaxBytesRecvByteBufAllocator.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/pool/SimpleChannelPool.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/pool/ChannelHealthChecker.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/bootstrap/BootstrapConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/group/VoidChannelGroupFuture.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/bootstrap/AbstractBootstrapConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/SelectStrategy.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/ChannelOutboundInvoker.java
* Copyright 2017 The Netty Project

* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
  "License"); you may not use this file except in compliance with the License. You may obtain a
  copy of the License at:
  *
  * http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software distributed under the License
  is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
  express
  or implied. See the License for the specific language governing permissions and limitations under
  the License.

*/

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/AbstractCoalescingBufferQueue.java

No license file was found, but licenses were detected in source scan.

*/

* Copyright 2017 The Netty Project

* The Netty Project licenses this file to you under the Apache License,
  version 2.0 (the "License"); you may not use this file except in compliance
  with the License. You may obtain a copy of the License at:
  *
  * http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/PendingBytesTracker.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/socket/ChannelOutputShutdownException.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/socket/ChannelOutputShutdownEvent.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/bootstrap/FailedChannel.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/nio/SelectedSelectionKeySetSelector.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/socket/oio/OioDatagramChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/DelegatingChannelPromiseNotifier.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/socket/ChannelInputShutdownReadComplete.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/internal/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/InternalChannelUtils.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/socket/oio/DefaultOioDatagramChannelConfig.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* Handles an I/O event or intercepts an I/O operation, and forwards it to its next handler in
* its { @link ChannelPipeline}.
Sub-types

{@link ChannelHandler} itself does not provide many methods, but you usually have to implement one of its subtypes:

- {@link ChannelInboundHandler} to handle inbound I/O events, and
- {@link ChannelOutboundHandler} to handle outbound I/O operations.

Alternatively, the following adapter classes are provided for your convenience:

- {@link ChannelInboundHandlerAdapter} to handle inbound I/O events,
- {@link ChannelOutboundHandlerAdapter} to handle outbound I/O operations, and
- {@link ChannelDuplexHandler} to handle both inbound and outbound events.

For more information, please refer to the documentation of each subtype.

The context object

A {@link ChannelHandlerContext} object. A {@link ChannelHandlerContext} is supposed to interact with the
{@link ChannelPipeline} it belongs to via a context object. Using the
context object, the {@link ChannelHandlerContext} can pass events upstream or
downstream, modify the pipeline dynamically, or store the information
(using {@link AttributeKey}s) which is specific to the handler.

State management

A {@link ChannelHandlerContext} often needs to store some stateful information.
The simplest and recommended approach is to use member variables:

```java
public interface Message {
    // your methods here
}

public class DataServerHandler extends SimpleChannelInboundHandler<Message> {
    private boolean loggedIn;

    @Override
    public void channelRead0(@link ChannelHandlerContext ctx, Message message) {
        if (message instanceof LoginMessage) {
            authenticate((LoginMessage) message);
        }
    }
}
```
```java
loggedIn = true;
}
else (message instanceof GetDataMessage) {
    if (loggedIn)
        ctx.writeAndFlush(fetchSecret((GetDataMessage) message));
    } else {
        fail();
    }
}
...
}
</pre>
* Because the handler instance has a state variable which is dedicated to
* one connection, you have to create a new handler instance for each new
* channel to avoid a race condition where a unauthenticated client can get
* the confidential information:
* 
* // Create a new handler instance per channel.
* // See [@link ChannelInitializer#initChannel(Channel)].
* public class DataServerInitializer extends [@link ChannelInitializer] {@link Channel}{
*     {@code @Override}
*     public void initChannel({@link Channel} channel) {
*         channel.pipeline().addLast("handler", new DataServerHandler());
*     }
* }
* </pre>
* 
* <h4>Using [@link AttributeKey]s</h4>
* Although it's recommended to use member variables to store the state of a
* handler, for some reason you might not want to create many handler instances.
* In such a case, you can use [@link AttributeKey]s which is provided by
* [{@link ChannelHandlerContext}]:
* 
* public interface Message {
*     // your methods here
* }
* 
* [{@code @Sharable}]
* public class DataServerHandler extends [@link SimpleChannelInboundHandler] Message {
*     private final [@link AttributeKey] Boolean auth =
*         AttributeKey.valueOf("auth");
*     
*     [{@code @Override}]
*     public void channelRead({@link ChannelHandlerContext} ctx, Message message) {
*         {[@link Attribute] Boolean attr = ctx.attr(auth);
*         if (message instanceof LoginMessage) {
```
authenticate((LoginMessage) o);
<b>attr.set(true)</b>;
} else (message instanceof GetDataMessage) {
  if (<b>Boolean.TRUE.equals(attr.get())</b>) {
    ctx.writeAndFlush(fetchSecret((GetDataMessage) o));
  } else {
    fail();
  }
}
...

Now that the state of the handler is attached to the @link ChannelHandlerContext, you can add the same handler instance to different pipelines:

    public class DataServerInitializer extends @link ChannelInitializer<@link Channel> {

        private static final DataServerHandler <b>SHARED</b> = new DataServerHandler();

        @Override
        public void initChannel(@link Channel channel) {
            channel.pipeline().addLast("handler", <b>SHARED</b>);
        }
    }

The @code @Sharable annotation

In the example above which used an @link AttributeKey, you might have noticed the @code @Sharable annotation.

If a @link ChannelHandler is annotated with the @code @Sharable annotation, it means you can create an instance of the handler just once and add it to one or more @link ChannelPipeline)s multiple times without a race condition.

If this annotation is not specified, you have to create a new handler instance every time you add it to a pipeline because it has unshared state such as member variables.

This annotation is provided for documentation purpose, just like the JCIP annotations.

Additional resources worth reading

Please refer to the @link ChannelHandler, and
* { @link ChannelPipeline } to find out more about inbound and outbound operations,
* what fundamental differences they have, how they flow in a pipeline, and how to handle
* the operation in your application.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/ChannelHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2018 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/socket/nio/NioChannelOption.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/SimpleUserEventChannelHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

/*
 * Open Source Used In DNAC Platform Antelope 10599
 */
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 * /

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/socket/io/OioServerSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/socket/io/DefaultOioSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/DefaultAddressedEnvelope.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/ChannelProgressiveFutureListener.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/ChannelId.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/DefaultChannelProgressivePromise.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/ChannelProgressivePromise.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/group/ChannelMatcher.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/ChannelHandlerAdapter.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/MessageSizeEstimator.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/socket/io/DefaultOioServerSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/socket/io/DefaultOioSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/socket/io/OioSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/ChannelId.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/ChannelProgressivePromise.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
jar/io.netty/channel/CoalescingBufferQueue.java
No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/ExtendedClosedChannelException.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/EventLoopTaskQueueFactory.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/ChannelHandlerMask.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/oio/OioByteStreamChannel.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2012 The Netty Project
* *
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/DefaultEventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/socket/SocketChannel.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/socket/nio/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/ChannelException.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/ServerChannel.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/nio/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/local/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-
  jar/io/netty/channel/nio/package-info.java
Open Source Used In DNAC Platform Antelope 10604

* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/socket/DatagramPacket.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/ChannelFutureListener.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/ChannelPromise.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/socket/DefaultDatagramChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/socket/DefaultServerSocketChannelConfig.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/oio/package-info.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/EventLoopGroup.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1.jar/io/netty/channel/local/LocalServerChannel.java
Open Source Used In DNAC Platform Antelope 10605
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/PendingWriteQueue.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/AbstractEventLoop.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/ChannelFactory.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/ReflectiveChannelFactory.java
* /opt/ws_local/PERMITS_SQL/1072871146_1595214703.66/0/netty-transport-4-1-42-final-sources-1-jar/io/netty/channel/embedded/EmbeddedChannelId.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project

~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0

~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):

1.2671 braces 2.3.2
1.2671.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2018, Jon Schlinkert.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2672 is-fullwidth-code-point 3.0.0

1.2672.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2673 glob-base 0.3.0

1.2673.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2674 color-convert 1.9.3

1.2674.1 Available under license:

Copyright (c) 2011-2016 Heather Arthur <fayearthur@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2675 next-tick 1.0.0
1.2675.1 Available under license:
The MIT License

Copyright (C) 2012-2016 Mariusz Nowak

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2676 traverse 0.3.9
1.2676.1 Available under license:

Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:
http://www.opensource.org/licenses/mit-license.php

Copyright 2010 James Halliday (mail@substack.net)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2677 require-main-filename 2.0.0

1.2677.1 Available under license:

Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2678 mixin-deep 1.3.2

1.2678.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2015, 2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2679 engine.io 1.8.5
1.2679.1 Available under license:
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2680 hadoop-mapreduce-client-common
3.2.2
1.2680.1 Available under license:
This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:
* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:
This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of
Gson 2.2.4,
which has the following notices:

    The Netty Project
    ==================================

Please visit the Netty web site for more information:

    * http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

--------------------------------------------------------------------------------------------------

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

    * LICENSE:
    * license/LICENSE.jsr166y.txt (Public Domain)
    * HOMEPAGE:
    * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
    * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

    * LICENSE:
* license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh Bloch of Google, Inc:

* LICENSE:
  * license/LICENSE.deque.txt (Public Domain)

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
* HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:
This product contains a modified portion of ‘jfastlz’, a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:  
* license/LICENSE.jfastlz.txt (MIT License)  
* HOMEPAGE:  
* https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on ‘Protocol Buffers’, Google’s data interchange format, which can be obtained at:

* LICENSE:  
* license/LICENSE.protobuf.txt (New BSD License)  
* HOMEPAGE:  
* http://code.google.com/p/protobuf/

This product optionally depends on ‘Bouncy Castle Crypto APIs’ to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:  
* license/LICENSE.bouncycastle.txt (MIT License)  
* HOMEPAGE:  
* http://www.bouncycastle.org/

This product optionally depends on ‘Snappy’, a compression library produced by Google Inc, which can be obtained at:

* LICENSE:  
* license/LICENSE.snappy.txt (New BSD License)  
* HOMEPAGE:  
* http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:  
* license/COPYING (Apache License 2.0)  
* HOMEPAGE:  
* https://github.com/google/guava

This product optionally depends on ‘JBoss Marshalling’, an alternative Java serialization API, which can be obtained at:
This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fastexxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:
This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

The content of package org.apache.commons.codec.language.bm has been translated from the original php source code available at http://stevemorse.org/phoneticinfo.htm with permission from the original authors.

Original source copyright: Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:

* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices:

* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
  Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
* TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of "Java Concurrency in Practice" book annotations 1.0, which has the following notices:

* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
  Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.
The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,
which has the following notices:

* ==============================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
  ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

  * The Apache 2.0 License
    http://www.apache.org/licenses/LICENSE-2.0.html

and

  * The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------

Eclipse

The following artifacts are EPL.
  * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
  * org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
  * org.eclipse.jetty.orbit:javax.mail.glassfish

------

Oracle
The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL/GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.
org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:

* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common http://hadoop.apache.org/ (Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by "GCC RUntime Library Exception"
== Contributors ==
* Tatu Saloranta
  * Providing benchmark suite
* Alec Wysoker
  * Performance and memory usage improvement

The binary distribution of this product bundles binaries of Xerces2 Java Parser 2.9.1, which has the following notices:
* =========================================================================
  ==  NOTICE file corresponding to section 4(d) of the Apache License, ==
  == Version 2.0, in this case for the Apache Xerces Java distribution. ==
=========================================================================

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
  - voluntary contributions made by Paul Eng on behalf of the
    Apache Software Foundation that were originally developed at iClick, Inc.,
    software copyright (c) 1999.

The binary distribution of this product bundles binaries of Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation
This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--
This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0

--
Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

--
This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--
This product includes portions of the Guava project v14, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of
Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Interset Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5 (https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of Plexus Cipher: encryption/decryption Component 1.4, which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java
  which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java
  which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of software.amazon.ion:ion-java 1.0.1, which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of joda-time:joda-time:2.9.9 which has the following notices:
* =--------------------------------------------------------------------------------------------=
  = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
  =--------------------------------------------------------------------------------------------=

This product includes software developed by Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1, which has the following notices:

* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml),
which has the following notices:

* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core),
which has the following notices:

* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4
which has the following notices:

* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, 
publicly display, publicly perform, sublicense, and distribute the 
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of 
this License, each Contributor hereby grants to You a perpetual, 
worldwide, non-exclusive, no-charge, royalty-free, irrevocable 
(except as stated in this section) patent license to make, have made, 
use, offer to sell, sell, import, and otherwise transfer the Work, 
where such license applies only to those patent claims licensable 
by such Contributor that are necessarily infringed by their 
Contribution(s) alone or by combination of their Contribution(s) 
with the Work to which such Contribution(s) was submitted. If You 
institute patent litigation against any entity (including a 
cross-claim or counterclaim in a lawsuit) alleging that the Work 
or a Contribution incorporated within the Work constitutes direct 
or contributory patent infringement, then any patent licenses 
granted to You under this License for that Work shall terminate 
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the 
Work or Derivative Works thereof in any medium, with or without 
modifications, and in Source or Object form, provided that You 
meet the following conditions:

(a) You must give any other recipients of the Work or 
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices 
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works 
that You distribute, all copyright, patent, trademark, and 
attraction notices from the Source form of the Work, 
excluding those notices that do not pertain to any part of 
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its 
distribution, then any Derivative Works that You distribute must 
include a readable copy of the attribution notices contained 
within such NOTICE file, excluding those notices that do not 
pertain to any part of the Derivative Works, in at least one 
of the following places: within a NOTICE text file distributed 
as part of the Derivative Works; within the Source form or 
documentation, if provided along with the Derivative Works; or, 
within a display generated by the Derivative Works, if and 
wherever such third-party notices normally appear. The contents 
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

For the org.apache.hadoop.util.bloom.* classes:
/**
 * Copyright (c) 2005, European Commission project OneLab under contract 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 * - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

For portions of the native implementation of slicing-by-8 CRC calculation in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor
the names of its contributors may be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*++
 *
 * Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
 *
 * This software program is licensed subject to the BSD License,
 * available at http://www.opensource.org/licenses/bsd-license.html
 *
 * Abstract: The main routine
 *
 --*/

For src/main/native/src/org/apache/hadoop/io/compress/lz4/{lz4.h,lz4.c,lz4hc.h,lz4hc.c},

/*
 LZ4 - Fast LZ compression algorithm
 Header File
 Copyright (C) 2011-2014, Yann Collet.
 BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright
 notice, this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above
 copyright notice, this list of conditions and the following disclaimer
 in the documentation and/or other materials provided with the
 distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT 
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR 
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT 
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, 
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT 
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, 
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY 
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT 
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE 
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :
- LZ4 source repository : http://code.google.com/p/lz4/ 
- LZ4 public forum : https://groups.google.com/forum/#!forum/lz4c 
*/

For hadoop-common-project/hadoop-common/src/main/native/gtest
---------------------------------------------------------------------
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without 
modification, are permitted provided that the following conditions are 
met:

* Redistributions of source code must retain the above copyright 
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above 
  copyright notice, this list of conditions and the following disclaimer 
in the documentation and/or other materials provided with the 
distribution.
* Neither the name of Google Inc. nor the names of its 
  contributors may be used to endorse or promote products derived from 
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT 
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR 
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT 
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, 
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT 
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, 
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY 
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT 
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1

---------------------------------------------------------------------
(GO license)
This is a work derived from Russ Cox's RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h

---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbdjni (https://github.com/fusesource/leveldbdjni), which is available under the following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of FuseSource Corp. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=============

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer.
in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/
--------------------------------------------------------------------------------
Copyright (C) 2008-2016, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js
--------------------------------------------------------------------------------
Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the
following license:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/bootstrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
ewm-helpers v0.5.13
ewm-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodeings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js

The MIT License (MIT)
Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE
OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js
--------------------------------------------------------------------------------
The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery
Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7
--------------------------------------------------------------------------------
MIT License

Copyright (c) 2003-2017 Optimatika
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
oj! Algorithms - version 43.0

--------------------------------------------------------------------------------


This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

For:
hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js
--------------------------------------------------------------------------------
Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js
--------------------------------------------------------------------------------
D3 is available under a 3-clause BSD license. For details, see:
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the
following license:
HSQldb Database 2.3.4

(HSQL License)
"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ""AS IS""
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL HSQldb Development Group, HSQldb.org,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.

The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions
thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.
2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such
combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a)rename the license and remove
any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER
NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is
responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
gizzly-framework 2.2.21
gizzly-http 2.2.21
gizzly-http-server 2.2.21
gizzly-http-servlet 2.2.21
gizzly-rcm 2.2.21

(CDDL 1.1)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii)
2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted:

(1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have
sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the
Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and
automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04
--------------------------------------------------------------------------------

By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the
following license:
JUnit 4.11
Eclipse JDT Core 3.1.1

--------------------------------------------------------------------------------
(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
i) changes to the Program, and
ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the
Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program. Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the
4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively ”Losses”) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.
This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

(2-clause BSD)
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction,
abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect,
whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients’ exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., “French translation of the Work by Original Author,” or “Screenplay based on original Work by Original Author”). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner
at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

jamon-runtime 2.4.1

--------------------------------------------------------------------------------
(MPL 2.0)
Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.
1.7. Larger Work
means a work that combines Covered Software with other material, in a separate
file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the
time of the initial grant or subsequently, any and all of the rights conveyed by
this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or
modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and
apparatus claims, in any patent Licensable by such Contributor that would be
infringed, but for the grant of the License, by the making, using, selling,
offering for sale, having made, import, or transfer of either its Contributions
or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General
Public License, Version 2.1, the GNU Affero General Public License, Version 3.0,
or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For
legal entities, You includes any entity that controls, is controlled by,
or is under common control with You. For purposes of this definition,
control means (a) the power, direct or indirect, to cause the direction or
management of such entity, whether by contract or otherwise, or (b) ownership of
more than fifty percent (50%) of the outstanding shares or beneficial ownership
of such entity.

2. License Grants and Conditions

2.1. Grants
Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are
its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further
distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.
5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.
9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.
You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

/*--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:
"This product includes software developed by the JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ""AS IS"" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

/*

The binary distribution of this product bundles these dependencies under the following license:
Hbase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.

Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN “AS-IS” BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE (“CCPL” OR “LICENSE”). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.
1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are
assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The
original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this
License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URL if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert,
as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same
terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of
the remainder of the terms of this License, and without further action
by the parties to this agreement, such provision shall be reformed to
the minimum extent necessary to make such provision valid and
enforceable.
d. No term or provision of this License shall be deemed waived and no
breach consented to unless such waiver or consent shall be in writing
and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with
respect to the Work licensed here. There are no understandings,
agreements or representations with respect to the Work not specified
here. Licensor shall not be bound by any additional provisions that
may appear in any communication from You. This License may not be
modified without the mutual written agreement of the Licensor and You.
f. The rights granted under, and the subject matter referenced, in this
License were drafted utilizing the terminology of the Berne Convention
for the Protection of Literary and Artistic Works (as amended on
September 28, 1979), the Rome Convention of 1961, the WIPO Copyright
Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996
and the Universal Copyright Convention (as revised on July 24, 1971).
These rights and subject matter take effect in the relevant
jurisdiction in which the License terms are sought to be enforced
according to the corresponding provisions of the implementation of
those treaty provisions in the applicable national law. If the
standard suite of rights granted under applicable copyright law
includes additional rights not granted under this License, such
additional rights are deemed to be included in the License; this
License is not intended to restrict the license of any rights under
applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty
whatsoever in connection with the Work. Creative Commons will not be
liable to You or any party on any legal theory for any damages
whatsoever, including without limitation any general, special,
incidental or consequential damages arising in connection to this
license. Notwithstanding the foregoing two (2) sentences, if Creative
Commons has expressly identified itself as the Licensor hereunder, it
shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the
Work is licensed under the CCPL, Creative Commons does not authorize
the use by either party of the trademark "Creative Commons" or any
related trademark or logo of Creative Commons without the prior
written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.

--------------------------------------------------------------------------------

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2681 hide-powered-by 1.0.0

1.2681.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014, 2015 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
OPEN SOURCE USED IN DNAC PLATFORM ANTELope

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2682 mesa 18.0.5-0ubuntu0~16.04.1

1.2682.1 Available under license:
Legal characters in GLSL are:

* Identifier characters:
  * Letters a-z
  * Letters A-Z
  * Underscore
  * Numbers 0-9
* Punctuation:
*  
*  Period, plus, dash, slash, asterisk, percent, angled brackets,
*  square brackets, parentheses, braces, caret, vertical bar,
*  ampersand, tilde, equals, exclamation point, colon, semicolon,
*  comma, and question mark
*  
* Special:
*  
*  Number sign (as used in preprocessor)
*  
*  Backslash just before newline as line continuation
*  
* White space:
*  
*  Space, horizontal tab, vertical tab, form feed, carriage-return,
*  and line-feed.
*  
* [GLSL Language Specification 4.30.6, section 3.1]
*  
* In this file, we test each of these in turn as follows:
*  
*  Identifier characters: All pass through unchanged
*  Punctuation: All pass through unchanged
*  Special: Empty directive replaced with blank line
*  Line continuation merges two lines, then a blank line
*  Whitespace: 4 horizontal space characters each replaced with space
*  2 newline characters each replaced with a newline
*  */

defghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

 0123456789
-+
-
/
*
%
<
>
[
]
(
)
{
}
# (C) Copyright IBM Corporation 2004
# All Rights Reserved.
#
# Permission is hereby granted, free of charge, to any person obtaining a
# copy of this software and associated documentation files (the "Software"),
# to deal in the Software without restriction, including without limitation
# on the rights to use, copy, modify, merge, publish, distribute, sub
# license, and/or sell copies of the Software, and to permit persons to whom
# the Software is furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice (including the next
# paragraph) shall be included in all copies or substantial portions of the
# Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
# IBM AND/OR ITS SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
# IN THE SOFTWARE.
#
# Authors:
#   Ian Romanick <idr@us.ibm.com>

bsd_license_template = """"%s
All Rights Reserved."""
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL %s, AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Please do not refer to the library as <em>MesaGL</em> (for legal reasons). It's just <em>Mesa</em> or <em>The Mesa 3-D graphics library</em>.  


<h1>License / Copyright Information</h1>

The Mesa distribution consists of several components. Different copyrights and licenses apply to different components. For example, the GLX client code uses the SGI Free Software License B, and some of the Mesa device drivers are copyrighted by their authors. See below for a list of Mesa's main components and the license for each.

The core Mesa library is licensed according to the terms of the MIT license. This allows integration with the XFree86, Xorg and DRI projects.

The default Mesa license is as follows:

Copyright (C) 1999-2007  Brian Paul   All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
Attention, Contributors

When contributing to the Mesa project you must agree to the licensing terms of the component to which you're contributing. The following section lists the primary components of the Mesa distribution and their respective licenses.

Mesa Component Licenses

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Mesa code</td>
<td>src/mesa/</td>
<td>MIT</td>
</tr>
<tr>
<td>Device drivers</td>
<td>src/mesa/drivers/*</td>
<td>MIT, generally</td>
</tr>
<tr>
<td>Gallium code</td>
<td>src/gallium/</td>
<td>MIT</td>
</tr>
<tr>
<td>Ext headers</td>
<td>include/GL/glext.h</td>
<td>Khronos</td>
</tr>
<tr>
<td>GLX client code</td>
<td>src/glx/</td>
<td>SGI Free Software License B</td>
</tr>
<tr>
<td>C11 thread</td>
<td>include/c11/threads*.h</td>
<td>Boost (permissive)</td>
</tr>
</tbody>
</table>

In general, consult the source files for license terms.
1.2683 braces 1.8.5
1.2683.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2684 babel-plugin-transform-exponentiation-operator 6.24.1 6.24.1
1.2684.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2685 update-notifier 4.1.3

1.2685.1 Available under license:

Copyright Google

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2686 opentracing-noop 0.31.0

1.2687 component-emitter 1.2.1

1.2687.1 Available under license:

(The MIT License)

Copyright (c) 2014 Component contributors <dev@component.io>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2688 caching-transform 4.0.0

1.2688.1 Available under license:

The MIT License (MIT)

Copyright (c) James Talmage <james@talmage.io> (github.com/jamestalmage)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2689 jaeger-core 1.5.0

1.2689.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2017, Uber Technologies, Inc
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/baggage/HttpBaggageRestrictionManagerProxy.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/baggage/BaggageRestrictionResponse.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/propagation/HexCodec.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/baggage/BaggageSetter.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/Reference.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/utils/Java6CompatibleThreadLocalRandom.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/baggage/Restriction.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/baggage/DefaultBaggageRestrictionManager.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/BaggageRestrictionManager.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/exceptions/BaggageRestrictionManagerException.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/baggage/RemoteBaggageRestrictionManager.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/BaggageRestrictionManagerProxy.java

No license file was found, but licenses were detected in source scan.


*/
* Copyright (c) 2019, The Jaeger Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1.5.0-sources-
  jar/io/jaegertracing/internal/propagation/BinaryCodec.java
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2018, The Jaeger Authors
 * Copyright (c) 2017, Uber Technologies, Inc
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1.5.0-sources-
  jar/io/jaegertracing/internal/propagation/B3TextMapCodec.java
No license file was found, but licenses were detected in source scan.

/**
 * Copyright (c) 2016-2017, Uber Technologies, Inc
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
Found in path(s):
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1.5.0-sources-jar/io/jaegertracing/internal/reporters/RemoteReporter.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2018, The Jaeger Authors
 * Copyright (c) 2016, Uber Technologies, Inc
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1.5.0-sources-jar/io/jaegertracing/internal/JaegerSpan.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1.5.0-sources-jar/io/jaegertracing/internal/JaegerTracer.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1.5.0-sources-jar/io/jaegertracing/internal/propagation/TextMapCodec.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1.5.0-sources-jar/io/jaegertracing/Configuration.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1.5.0-sources-jar/io/jaegertracing/internal/JaegerSpanContext.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2018, The Jaeger Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2016, Uber Technologies, Inc
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/samplers/RateLimitingSampler.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/samplers/ConstSampler.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/utils/Http.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/exceptions/UnsupportedFormatException.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/samplers/SamplingStatus.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/exceptions/EmptyIpException.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/samplers/NoopSender.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/exceptions/TraceIdOutOfBoundException.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/JaegerObjectFactory.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/PropagationRegistry.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/SenderFactory.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/package-info.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/NoopSenderFactory.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/NoopSender.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/SenderResolver.java

jar/io/jaegertracing/internal/reporters/LoggingReporter.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/Constants.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/samplers/http/RateLimitingSamplingStrategy.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/Sender.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/metrics/Metrics.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/clock/SystemClock.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/samplers/ProbabilisticSampler.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/clock/Clock.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/exceptions/EmptyTracerStateStringException.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/exceptions/SenderException.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/exceptions/NotFourOctetsException.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2017, The Jaeger Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/Codec.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/metrics/InMemoryMetricsFactory.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/metrics/NoopMetricsFactory.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/MetricsFactory.java
*/opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/spi/MetricsFactory.java
jar/io/jaegertracing/internal/propagation/CompositeCodec.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2020, The Jaeger Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
 * in compliance with the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
 * or implied. See the License for the specific language governing permissions and limitations under
 * the License.
 */

Found in path(s):
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/clock/MillisAccurateClock.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/MDCScopeManager.java
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/clock/MicrosAccurateClock.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2020, OpenTelemetry Authors
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1152086972_1618069762.04/0/jaeger-core-1-5-0-sources-jar/io/jaegertracing/internal/propagation/TraceContextCodec.java
1.2690 lodash-union 4.6.0

1.2690.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2691 parquet-encoding 1.12.0
1.2691.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This project includes code from Daniel Lemire's JavaFastPFOR project in this binary artifact. The “Lemire” bit packing classes produced by parquet-generator are derived from the JavaFastPFOR project.
1.2692 websocket-common 9.4.17.v20190418

1.2692.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (“AGREEMENT”). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT’S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor’s behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.
"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for
damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the
Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as
reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "["]
replaced with your own identifying information. (Don't include
the brackets!)  The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

=====================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=====================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both
* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
  * org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
  * org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
  * org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
  https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

  * javax.servlet:javax.servlet-api
  * javax.annotation:javax.annotation-api
  * javax.transaction:javax.transaction-api
  * javax.websocket:javax.websocket-api

-----

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

  * java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2693 browser-stdout 1.3.1
1.2693.1 Available under license:
Copyright 2018 kumavis

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2694 core-js 2.6.12 2.6.12
1.2694.1 Available under license:
Copyright (c) 2014-2020 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2695 find-up 2.1.0
1.2695.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2696 chokidar 3.5.1
1.2696.1 Available under license :
The MIT License (MIT)

Copyright (c) 2012-2019 Paul Miller (https://paulmillr.com), Elan Shanker

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the Software), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2697 avro 1.7.7

1.2697.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the
Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an
addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2698 wrap-ansi 2.1.0
1.2698.1 Available under license : The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2699 annotations 16.0.3

1.2699.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2006 Sascha Weinreuter
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 *    http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-jar/org/intellij/lang/annotations/PrintFormat.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-jar/org/intellij/lang/annotations/Identifier.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-jar/org/intellij/lang/annotations/Subst.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
jar/org/intellij/lang/annotations/Pattern.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
jar/org/intellij/lang/annotations/Language.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
jar/org/intellij/lang/annotations/RegExp.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2000-2016 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
jar/org/jetbrains/annotations/Contract.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2000-2012 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
jar/org/intellij/lang/annotations/JdkConstants.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
jar/org/jetbrains/annotations/NotNull.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2000-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
  jar/org/jetbrains/annotations/Nls.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
  jar/org/intellij/lang/annotations/Flow.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
  jar/org/jetbrains/annotations/TestOnly.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2000-2014 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-
  jar/org/jetbrains/annotations/Nullable.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2000-2009 JetBrains s.r.o.
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-jar/org/jetbrains/annotations/NonNls.java
* /opt/ws_local/PERMITS_SQL/1066932998_1593721013.75/0/annotations-16-0-3-sources-jar/org/jetbrains/annotations/PropertyKey.java

1.2700 tough-cookie 2.5.0 2.5.0

1.2700.1 Available under license:

Copyright (c) 2015, Salesforce.com, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of Salesforce.com nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2701 os-tmpdir 1.0.2 1.0.2
1.2701.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2702 xerces-j 2.9.1
1.2702.1 Available under license :
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You
distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages of loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2703 antlr-runtime 4.9
1.2703.1 Available under license:

No license file was found, but licenses were detected in source scan.
Open Source Used In DNAC Platform Antelope 10736

* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime-4.9-sources-jar/org/antlr/v4/runtime/misc/Pair.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime-4.9-sources-jar/org/antlr/v4/runtime/tree/xpath/XPath.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/FailedPredicateException.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/ConsoleErrorListener.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/VocabularyImpl.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/xpath/XPathLexerErrorListener.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/UnbufferedCharStream.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/atn/PlusBlockStartState.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/atn/RuleTransition.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/atn/SingletonPredictionContext.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/atn/SetTransition.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/atn/DecisionInfo.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/xpath/XPathLexerErrorListener.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/UnbufferedCharStream.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/xpath/XPathLexerErrorListener.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/UnbufferedTokenStream.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/pattern/Chunk.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/misc/Utils.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/misc/IntegerList.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/xpath/XPathRuleAnywhereElement.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/XPathRuleAnywhereElement.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/xpath/XPathRuleAnywhereElement.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/xpath/XPathRuleAnywhereElement.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/xpath/XPathRuleAnywhereElement.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4.9-sources-jar/antlr4-runtime/tree/xpath/XPathRuleAnywhereElement.java
jar/org/antlr/v4/runtime/aten/RangeTransition.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4-9-sources-jar/org/antlr/v4/runtime/misc/Triple.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4-9-sources-jar/org/antlr/v4/runtime/aten/ATNType.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4-9-sources-jar/org/antlr/v4/runtime/tree/ErrorNodeImpl.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4-9-sources-jar/org/antlr/v4/runtime/tree/pattern/TokenTagToken.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4-9-sources-jar/org/antlr/v4/runtime/TokenStreamRewriter.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4-9-sources-jar/org/antlr/v4/runtime/aten/AbstractPredicateTransition.java
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4-9-sources-jar/org/antlr/v4/runtime/aten/EmptyPredictionContext.java

No license file was found, but licenses were detected in source scan.

~ Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.

Found in path(s):
* /opt/cola/permits/1154413561_1618841561.55/0/antlr4-runtime-4-9-sources-jar/META-INF/maven/org.antlr/antlr4-runtime/pom.xml

1.2704 jsreport-sample-template 2.6.1
1.2704.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.
An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
1.2705 regenerate 1.4.0 1.4.0

1.2705.1 Available under license:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2706 json-schema-ref-parser 6.1.0 6.1.0

1.2706.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 James Messinger

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2707 babel-plugin-transform-es2015-arrow-functions 6.22.0

1.2707.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2708 parquet-avro 1.12.0

1.2708.1 Available under license:

Apache Parquet MR (Incubating)
Copyright 2014-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

---------------------------------------------------------------------

This product includes code from Apache Avro, which includes the following in its NOTICE file:

Apache Avro
Copyright 2010-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This product includes code from Apache Avro.

Copyright: 2014 The Apache Software Foundation.
Home page: https://avro.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

1.2709 jetty-xml 9.4.34.v20201102

1.2709.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

“Contribution” means:
a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
   a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
   b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
   c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or
otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain any resulting litigation.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==============================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL/GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from
following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2710 indexof 0.0.1

1.2710.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2711 isorelax 20090621

1.2711.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * @(#)Id: IslandSchemaReader.java,v 1.4 2001/11/01 00:11:45 kkawa Exp $
 *
 * Copyright 2001 KAWAGUCHI Kohsuke
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be included
 * in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 * IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
 * CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
 * TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
 * SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 */

Found in path(s):
* /opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-jar/org/iso_relax/dispatcher/IslandSchemaReader.java

No license file was found, but licenses were detected in source scan.

/*
 * @(#)Id: IgnoredSchema.java,v 1.5 2003/05/30 23:46:32 kkawa Exp $
 *
 * Copyright 2001 Kohsuke KAWAGUCHI
 */
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*/

Found in path(s):
*/opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-
jar/org/iso_relax/dispatcher/impl/IgnoredSchema.java
No license file was found, but licenses were detected in source scan.

/*@(#)Id: Dispatcher.java,v 1.6 2003/05/30 23:46:32 kkawa Exp $
* Copyright 2001 MURATA Makoto, KAWAGUCHI Kohsuke
* Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Found in path(s):
*/

/opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-
jar/org/iso_relax/dispatcher/Dispatcher.java

No license file was found, but licenses were detected in source scan.

*/

* @(#)Id: IslandVerifier.java,v 1.6 2003/05/30 23:46:32 kkawa Exp $
*
* Copyright 2001 MURATA Makoto, KAWAGUCHI Kohsuke
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
*/

/opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-
jar/org/iso_relax/dispatcher/IslandVerifier.java

No license file was found, but licenses were detected in source scan.

*/

* @(#)Id: AttributesDecl.java,v 1.3 2001/11/01 00:11:45 kkawa Exp $
*
* Copyright 2001 Kohsuke KAWAGUCHI
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-
  jar/org/iso_relax/dispatcher/AttributesDecl.java

No license file was found, but licenses were detected in source scan.

/ *
* @(#)Id: SchemaProvider.java,v 1.6 2003/05/30 23:46:32 kkawa Exp $ *
* Copyright 2001 Kohsuke KAWAGUCHI *
* Permission is hereby granted, free of charge, to any person obtaining *
* a copy of this software and associated documentation files (the *
* "Software"), to deal in the Software without restriction, including *
* without limitation the rights to use, copy, modify, merge, publish, *
* distribute, sublicense, and/or sell copies of the Software, and to *
* permit persons to whom the Software is furnished to do so, subject to *
* the following conditions:
* The above copyright notice and this permission notice shall be included *
* in all copies or substantial portions of the Software. *
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF *
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY *
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE *
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-
  jar/org/iso_relax/dispatcher/SchemaProvider.java
No license file was found, but licenses were detected in source scan.

/*
 * @(#)Id: ElementDecl.java,v 1.3 2001/11/01 00:11:45 kkawa Exp $
 * Copyright 2001 Kohsuke KAWAGUCHI 
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 * 
 * The above copyright notice and this permission notice shall be included
 * in all copies or substantial portions of the Software.
 * 
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 * IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
 * CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
 * TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
 * SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 */

Found in path(s):
* /opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-
  jar/org/iso_relax/dispatcher/ElementDecl.java

No license file was found, but licenses were detected in source scan.

/*
 * @(#)Id: IslandSchema.java,v 1.5 2003/05/30 23:46:32 kkawa Exp $
 * Copyright 2001 KAWAGUCHI Kohsuke 
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 * 
 * The above copyright notice and this permission notice shall be included
 * in all copies or substantial portions of the Software.
 * 

* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-jar/org/iso_relax/dispatcher/IslandSchema.java

No license file was found, but licenses were detected in source scan.

/*
 * @(#)Id: DispatcherImpl.java,v 1.5 2003/05/30 23:46:32 kkawa Exp $
 *
 * Copyright 2001 Kohsuke KAWAGUCHI
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *  
 * The above copyright notice and this permission notice shall be included
 * in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 * IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
 * CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
 * TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
 * SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/1135864262_1613618257.27/0/isorelax-20090621-sources-1-jar/org/iso_relax/dispatcher/impl/DispatcherImpl.java

1.2712 metrics 2.2.0
1.2.712.1 Available under license:

Apache HBase - Metrics Implementation
Copyright 2007-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object...
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2713 jheaps 0.9

1.2713.1 Available under license:

No license file was found, but licenses were detected in source scan.

  <modelVersion>4.0.0</modelVersion>
  <groupId>org.jheaps</groupId>
  <artifactId>jheaps</artifactId>
  <version>0.9</version>
  <packaging>jar</packaging>
  <name>JHeaps</name>
  <description>A free, production-ready, efficient Java library containing a collection of heap data-structures.</description>
  <url>http://www.jheaps.org</url>

  <licenses>
    <license>
      <name>Apache License, Version 2.0</name>
      <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
      <distribution>repo</distribution>
    </license>
  </licenses>

  <properties>
    <java.version>1.6</java.version>
  </properties>
</project>
<distributionManagement>
  <snapshotRepository>
    <id>ossrh</id>
    <url>https://oss.sonatype.org/content/repositories/snapshots/</url>
  </snapshotRepository>
  <repository>
    <id>ossrh</id>
    <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
  </repository>
</distributionManagement>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>${maven-compiler-plugin.version}</version>
      <configuration>
        <source>${java.version}</source>
        <target>${java.version}</target>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>${maven-jar-plugin.version}</version>
      <configuration>
        <excludes>
          <exclude>logback.xml</exclude>
        </excludes>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-release-plugin</artifactId>
      <version>${maven-release-plugin.version}</version>
      <configuration>
        <mavenExecutorId>forked-path</mavenExecutorId>
      </configuration>
    </plugin>
  </plugins>
</build>
<execution>
  <id>default-check</id>
  <goals>
    <goal>check</goal>
  </goals>
  <configuration>
    <rules>
      <rule>
        <element>BUNDLE</element>
        <limits>
          <limit>
            <counter>INSTRUCTION</counter>
            <value>COVEREDRATIO</value>
            <minimum>0.75</minimum>
          </limit>
          <limit>
            <counter>CLASS</counter>
            <value>COVEREDRATIO</value>
            <minimum>0.75</minimum>
          </limit>
          <limit>
            <counter>METHOD</counter>
            <value>COVEREDRATIO</value>
            <minimum>0.75</minimum>
          </limit>
          <limit>
            <counter>BRANCH</counter>
            <value>COVEREDRATIO</value>
            <minimum>0.75</minimum>
          </limit>
          <limit>
            <counter>COMPLEXITY</counter>
            <value>COVEREDRATIO</value>
            <minimum>0.75</minimum>
          </limit>
          <limit>
            <counter>LINE</counter>
            <value>COVEREDRATIO</value>
            <minimum>0.75</minimum>
          </limit>
        </limits>
      </rule>
    </rules>
  </configuration>
</executions>
</plugin>
<build>

<reporting>
<plugins>
<plugin>
<groupId>org.jacoco</groupId>
<artifactId>jacoco-maven-plugin</artifactId>
<reportSets>
<reportSet>
<reports>
<report>report</report>
</reports>
</reportSet>
</reportSets>
</plugin>
</plugins>
</reporting>

<profiles>
<profile>
<id>release-sign-artifacts</id>
<activation>
<property>
<name>performRelease</name>
/value>true</value>
</property>
</activation>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-gpg-plugin</artifactId>
<version>1.5</version>
<executions>
<execution>
<id>sign-artifacts</id>
<phase>verify</phase>
<goals>
<goal>sign</goal>
</goals>
</execution>
</executions>
</plugin>
</plugins>
</build>
</profile>
</profiles>
<show>public</show>
<windowtitle>JHeaps</windowtitle>
</configuration>
<executions>
<execution>
<id>attach-javadocs</id>
<goals>
<goal>jar</goal>
</goals>
</execution>
<executions>
</plugin>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-source-plugin</artifactId>
<executions>
<execution>
<id>attach-sources</id>
<goals>
<goal>jar-no-fork</goal>
</goals>
</execution>
<executions>
</plugin>
</build>
</profile>
<profile>
<id>checkstyle</id>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-checkstyle-plugin</artifactId>
<version>${maven-checkstyle-plugin.version}</version>
<dependencies>
<dependency>
<groupId>com.puppycrawl.tools</groupId>
<artifactId>checkstyle</artifactId>
<version>${checkstyle.version}</version>
</dependency>
<dependency>
<groupId>com.puppycrawl.tools</groupId>
<artifactId>checkstyle</artifactId>
<version>${checkstyle.version}</version>
</dependency>
<executions>
<execution>
<id>checkstyle</id>
<phase>validate</phase>
<goals>
<goal>check</goal>
</goals>
</execution>
<executions>
</plugin>
</build>
</profile>
</profile>
</profile>
</profile>
</configuration>
1.2714 commons-text 1.6

1.2714.1 Available under license:

Apache Commons Text
Copyright 2014-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2715 drill-opentsdb-storage 1.19.0
1.2715.1 Available under license :

Drill : Contrib : Storage : OpenTSDB
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS.
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2716 metrics-core 2.2.0
1.2716.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Written by Doug Lea with assistance from members of JCP JSR-166
 * Expert Group and released to the public domain, as explained at
 * http://creativecommons.org/publicdomain/zero/1.0/
 * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/main/java/util/concurrent/ThreadLocalRandom.java?view=markup
 */

Found in path(s):
* /opt/cola/permits/1145440743_1616178883.36/0/metrics-core-2-2-0-sources-5-jar/com/yammer/metrics/stats/ThreadLocalRandom.java

1.2717 de.huxhorn.sulky.codec 0.9.17
1.2717.1 Available under license:
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.
As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An ”Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A ”Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the ”Linked Version”.

The ”Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The ”Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.2718 libxcursor 1.1.14-1ubuntu0.16.04.2

1.2718.1 Available under license :

Copyright 2002 Keith Packard

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Keith Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Keith Packard makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

KEITH PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL KEITH PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2719 escape-string-regexp 1.0.5 1.0.5

1.2719.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2720 sqlite-jdbc 3.30.1

1.2720.1 Available under license:

Copyright (c) 2006, David Crawshaw. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced
with your own identifying information. (Don't include the
brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
1.2721 prop-types 15.7.2 15.7.2

1.2721.1 Available under license:
MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2722 es6-symbol 3.1.1

1.2722.1 Available under license:
Copyright (C) 2013-2015 Mariusz Nowak (www.medikoo.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.2723 graphite 1.3.10-0ubuntu0.16.04.1
1.2723.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations
below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.
For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.
For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

**GNU LESSER GENERAL PUBLIC LICENSE**

**TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)
"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has
a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least
three years, to give the same user the materials specified in
Subsection 6a, above, for a charge no more than the cost of
performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above,

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms
of the ordinary General Public License).

To apply these terms, attach the following notices to the library.
It is safest to attach them to the start of each source file to most
effectively convey the exclusion of warranty; and each file should
have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
/* GRAPHITE2 LICENSING

Copyright 2010, SIL International
All rights reserved.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

Lesser General Public License for more details.

You should also have received a copy of the GNU Lesser General Public License along with this library in the file named "LICENSE". If not, write to the Free Software Foundation, 51 Franklin Street, Suite 500, Boston, MA 02110-1335, USA or visit their web page on the internet at http://www.fsf.org/licenses/lgpl.html.

Alternatively, you may use this library under the terms of the Mozilla Public License (http://mozilla.org/MPL) or under the GNU General Public License, as published by the Free Software Foundation; either version 2 of the license or (at your option) any later version.

/*
Format: http://dep.debian.net/deps/dep5
Upstream-Name: graphite2
Source: http://sf.net/projects/silgraphite

Files: *
Copyright: (C) 2010-2011, SIL International
License: LGPL-2+
   This library is free software; you can redistribute it and/or
   modify it under the terms of the GNU Library General Public License
   as published by the Free Software Foundation; either version 2 of
   the License, or (at your option) any later version.

   This library is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
   Library General Public License for more details.

   On Debian systems you should have a local copy of the GNU Library
   General Public License in /usr/share/common-licenses/LGPL-2. If you
downloaded the source you will find it in the file LICENSE; if not,
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Files: contrib/perl/*
Copyright: Copyright (C) 2011 Simon Cozens <simon@cpan.org>
License: GPL1+, Artistic
   This program is free software; you can redistribute it and/or modify
   it under the terms of either:

   a) the GNU General Public License as published by the Free Software
      Foundation; either version 1, or (at your option) any later
      version, or

   b) the "Artistic License" which comes with Perl.
On Debian GNU/Linux systems, the complete text of the GNU General
Public License can be found in `/usr/share/common-licenses/GPL` and
the Artistic Licence in `/usr/share/common-licenses/Artistic`.

Files: contrib/perl/ppport.h
Copyright: 2004-2010, Marcus Holland-Moritz <mhx-cpan@gmx.net>
2001, Paul Marquess <pmqs@cpan.org> (Version 2.x)
1999, Kenneth Albanowski <kjahds@kjahds.com> (Version 1.x)
License: Artistic or GPL-1+

Files: debian-src/*
Copyright: 2010 Martin Hosken <martin_hosken@sil.org>
License: LGPL-2+
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Library General Public License for more details.

On Debian systems you should have a local copy of the GNU Library
General Public License in /usr/share/common-licenses/LGPL-2. If you
downloaded the source you will find it in the file LICENSE; if not,
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Files: debian*/*
Copyright: 2010 Martin Hosken <martin_hosken@sil.org>
2011 Rene Engelhard <renee@debian.org>
License: LGPL-2+
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Library General Public License for more details.

On Debian systems you should have a local copy of the GNU Library
General Public License in /usr/share/common-licenses/LGPL-2. If you
downloaded the source you will find it in the file LICENSE; if not,
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Files: tests/comparerenderer/icule/*
Copyright: Copyright (c) 1995-2006 International Business Machines Corporation and others
License: other
Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies of
the Software and that both the above copyright notice(s) and this
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale, use
or other dealings in this Software without prior written authorization
of the copyright holder.

Files: gr2fonttest/gr2FontTest.cpp
Copyright: Copyright (C) 2005 www.thanlwinsoft.org
License: LGPL | other

Files: tests/comparerenderer/GrUtfTextSrc.*
Copyright: Copyright (C) 2004 Keith Stribley
License: MPL-1.1 | GPL-2 | LGPL-2.1

Files: tests/fonts/*
License: other

-----------------------------------------------------------
SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007
-----------------------------------------------------------

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide
development of collaborative font projects, to support the font creation
efforts of academic and linguistic communities, and to provide a free and
open framework in which fonts may be shared and improved in partnership
with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS
"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS
Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding
Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Format: http://dep.debian.net/deps/dep5
Upstream-Name: graphite2
Source: http://sf.net/projects/silgraphite

Files: *
Copyright: (C) 2010-2011, SIL International
License: LGPL-2+
This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

On Debian systems you should have a local copy of the GNU Library General Public License in /usr/share/common-licenses/LGPL-2. If you
downloaded the source you will find it in the file LICENSE; if not,
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301 USA

Files: contrib/perl/*
Copyright: Copyright (C) 2011 Simon Cozens <simon@cpan.org>
License: GPL1+ | Artistic
This program is free software; you can redistribute it and/or modify
it under the terms of either:

a) the GNU General Public License as published by the Free Software
   Foundation; either version 1, or (at your option) any later
   version, or

b) the "Artistic License" which comes with Perl.

On Debian GNU/Linux systems, the complete text of the GNU General
Public License can be found in `/usr/share/common-licenses/GPL' and
the Artistic Licence in `/usr/share/common-licenses/Artistic'.

Files: contrib/perl/ppport.h
Copyright: 2004-2010, Marcus Holland-Moritz <mhx-cpan@gmx.net>
2001, Paul Marquess <pmqs@cpan.org> (Version 2.x)
1999, Kenneth Albanowski <kjahds@kjahds.com> (Version 1.x)
License: Artistic or GPL-1+

Files: debian-src/*
Copyright: 2010 Martin Hosken <martin_hosken@sil.org>
License: LGPL-2+
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Library General Public License for more details.

On Debian systems you should have a local copy of the GNU Library
General Public License in /usr/share/common-licenses/LGPL-2. If you
downloaded the source you will find it in the file LICENSE; if not,
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301 USA

Files: debian/*/*
Copyright: 2010 Martin Hosken <martin_hosken@sil.org>
2011 Rene Engelhard <rene@debian.org>
License: LGPL-2+

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

On Debian systems you should have a local copy of the GNU Library General Public License in /usr/share/common-licenses/LGPL-2. If you downloaded the source you will find it in the file LICENSE; if not, along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Files: tests/comparerenderer/icule/*
Copyright: Copyright (c) 1995-2006 International Business Machines Corporation and others
License: other
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Files: gr2fonttest/gr2FontTest.cpp
Copyright: Copyright (C) 2005 www.thanlwinsoft.org
License: LGPL | other
Files: tests/comparerenderer/GrUtfTextSrc.*
Copyright: Copyright (C) 2004 Keith Stribley
License: MPL-1.1 | GPL-2 | LGPL-2.1

Files: tests/fonts/*
License: other

-----------------------------------------------------------
SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007
-----------------------------------------------------------

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide
development of collaborative font projects, to support the font creation
efforts of academic and linguistic communities, and to provide a free and
open framework in which fonts may be shared and improved in partnership
with others.

The OFL allows the licensed fonts to be used, studied, modified and
redistributed freely as long as they are not sold by themselves. The
fonts, including any derivative works, can be bundled, embedded,
redistributed and/or sold with any software provided that any reserved
names are not used by derivative works. The fonts and derivatives,
however, cannot be released under any other type of license. The
requirement for fonts to remain under this license does not apply
to any document created using the fonts or their derivatives.

DEFINITIONS
"Font Software" refers to the set of files released by the Copyright
Holder(s) under this license and clearly marked as such. This may
include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the
copyright statement(s).

"Original Version" refers to the collection of Font Software components as
distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting,
or substituting -- in part or in whole -- any of the components of the
Original Version, by changing formats or by porting the Font Software to a
new environment.

"Author" refers to any designer, engineer, programmer, technical
writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS
Permission is hereby granted, free of charge, to any person obtaining
a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.
1.2724 ant-launcher 1.10.9

1.2724.1 Available under license:

/*
 * Apache License
 * Version 2.0, January 2004
 * http://www.apache.org/licenses/
 *
 * TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
 *
 * 1. Definitions.
 *
 * "License" shall mean the terms and conditions for use, reproduction,
 * and distribution as defined by Sections 1 through 9 of this document.
 *
 * "Licensor" shall mean the copyright owner or entity authorized by
 * the copyright owner that is granting the License.
 *
 * "Legal Entity" shall mean the union of the acting entity and all
 * other entities that control, are controlled by, or are under common
 * control with that entity. For the purposes of this definition,
 * "control" means (i) the power, direct or indirect, to cause the
 * direction or management of such entity, whether by contract or
 * otherwise, or (ii) ownership of fifty percent (50%) or more of the
 * outstanding shares, or (iii) beneficial ownership of such entity.
 *
 * "You" (or "Your") shall mean an individual or Legal Entity
 * exercising permissions granted by this License.
 *
 * "Source" form shall mean the preferred form for making modifications,
 * including but not limited to software source code, documentation
 * source, and configuration files.
 *
 * "Object" form shall mean any form resulting from mechanical
 * transformation or translation of a Source form, including but
 * not limited to compiled object code, generated documentation,
 * and conversions to other media types.
 *
 * "Work" shall mean the work of authorship, whether in Source or
 * Object form, made available under the License, as indicated by a
 * copyright notice that is included in or attached to the work
 * (an example is provided in the Appendix below).
 *
 * "Derivative Works" shall mean any work, whether in Source or Object
 * form, that is based on (or derived from) the Work and for which the
 * editorial revisions, annotations, elaborations, or other modifications
 * represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
* except as required for reasonable and customary use in describing the
* origin of the Work and reproducing the content of the NOTICE file.
*
* 7. Disclaimer of Warranty. Unless required by applicable law or
* agreed to in writing, Licensor provides the Work (and each
* Contributor provides its Contributions) on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied, including, without limitation, any warranties or conditions
* of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
* PARTICULAR PURPOSE. You are solely responsible for determining the
* appropriateness of using or redistributing the Work and assume any
* risks associated with Your exercise of permissions under this License.
*
* 8. Limitation of Liability. In no event and under no legal theory,
* whether in tort (including negligence), contract, or otherwise,
* unless required by applicable law (such as deliberate and grossly
* negligent acts) or agreed to in writing, shall any Contributor be
* liable to You for damages, including any direct, indirect, special,
* incidental, or consequential damages of any character arising as a
* result of this License or out of the use or inability to use the
* Work (including but not limited to damages for loss of goodwill,
* work stoppage, computer failure or malfunction, or any and all
* other commercial damages or losses), even if such Contributor
* has been advised of the possibility of such damages.
*
* 9. Accepting Warranty or Additional Liability. While redistributing
* the Work or Derivative Works thereof, You may choose to offer,
* and charge a fee for, acceptance of support, warranty, indemnity,
* or other liability obligations and/or rights consistent with this
* License. However, in accepting such obligations, You may act only
* on Your own behalf and on Your sole responsibility, not on behalf
* of any other Contributor, and only if You agree to indemnify,
* defend, and hold each Contributor harmless for any liability
* incurred by, or claims asserted against, such Contributor by reason
* of Your accepting any such warranty or additional liability.
*
* END OF TERMS AND CONDITIONS
*
* APPENDIX: How to apply the Apache License to your work.
*
* To apply the Apache License to your work, attach the following
* boilerplate notice, with the fields enclosed by brackets "[]"
* replaced with your own identifying information. (Don't include
* the brackets!) The text should be enclosed in the appropriate
* comment syntax for the file format. We also recommend that a
* file or class name and description of purpose be included on the
* same "printed page" as the copyright notice for easier
* identification within third-party archives.
* Copyright [yyyy] [name of copyright owner]
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* https://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*
W3C SOFTWARE NOTICE AND LICENSE
https://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.
The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered under http://saxproject.org/?selected=pd

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05
Apache Ant
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.
1.2725 camelize 1.0.0

1.2725.1 Available under license:
This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2726 ieee754 1.1.13

1.2726.1 Available under license:
Copyright 2008 Fair Oaks Labs, Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2727 asm 7.3.1

1.2727.1 Available under license:
No license file was found, but licenses were detected in source scan.

// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
// 3. Neither the name of the copyright holders nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/FieldVisitor.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/Edge.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/Handler.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/Constants.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/RecordComponentWriter.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/RecordComponentVisitor.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/ClassReader.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/Type.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/ByteVector.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/TypeReference.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/ClassTooLargeException.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/TypePath.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/FieldWriter.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/SymbolTable.java
* /opt/cola/permits/1135907460_1613628459.33/0/asm-7-3-1-sources-2-jar/org/objectweb/asm/AnnotationWriter.java
No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
  modification, are permitted provided that the following conditions
  are met:
  1. Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.
  2. Redistributions in binary form must reproduce the above copyright
     notice, this list of conditions and the following disclaimer in the
     documentation and/or other materials provided with the distribution.
  3. Neither the name of the copyright holders nor the names of its
     contributors may be used to endorse or promote products derived from
     this software without specific prior written permission.
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
  ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
  LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
  CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
  SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
  INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
  CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
  ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
  THE POSSIBILITY OF SUCH DAMAGE.
1.2728 ws 6.2.2

1.2728.1 Available under license:
(The MIT License)

Copyright (c) 2017-2019 Kamil Myliwiec <http://kamilmysliwiec.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2729 base64-js 1.3.1

1.2729.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Jameson Little

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2730 secure-keys 1.0.0

1.2730.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2731 avro-ipc-jetty 1.9.1

1.2731.1 Available under license:

Apache Avro IPC Jetty
Copyright 2009-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2732 side-channel 1.0.4 1.0.4
1.2732.1 Available under license:
MIT License

Copyright (c) 2019 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2733 concat-stream 2.0.0
1.2733.1 Available under license:
The MIT License

Copyright (c) 2013 Max Ogden

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2734 commons-configuration 2.1.1

1.2734.1 Available under license :

Apache Commons Configuration
Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2735 metrics-jmx 4.0.2

1.2736 to-object-path 0.3.0

1.2736.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2737 collection-visit 1.0.0

1.2737.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2738 chainsaw 0.1.0

1.2738.1 Available under license:

Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT license:
http://www.opensource.org/licenses/mit-license.php
1.2739 natty 0.13

1.2739.1 Available under license:
No license file was found, but licenses were detected in source scan.

<name>MIT License</name>

Found in path(s):
* /opt/cola/permits/1136702688_1613864332.59/0/natty-0-13-sources-3-jar/META-INF/maven/com.joestelmach/natty/pom.xml

1.2740 is-fullwidth-code-point 1.0.0

1.2740.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2741 kind-of 5.1.0

1.2741.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2742 hbase-zookeeper 2.4.2

1.2742.1 Available under license:

Apache HBase - Zookeeper
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced
with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2743 @babel/helpers 7.13.10
1.2743.1 Available under license :
    MIT License

    Copyright (c) 2014-present Sebastian McKenzie and other contributors

    Permission is hereby granted, free of charge, to any person obtaining
    a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2744 es6-error 4.1.1

1.2744.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Ben Youngblood

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2745 is-posix-bracket 0.1.1
1.2745.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2746 pump 3.0.0

1.2746.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2747 mylib 0.0.0
1.2747.1 Available under license:
MIT License

Copyright (c) 2012 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2748 babel-helper-function-name 6.24.1
1.2748.1 Available under license:
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

MIT License
Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2749 jsreport-chrome-pdf 1.10.0

1.2749.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.
0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.2750 gson 2.8.2

1.2751 jetty-util 9.4.34.v20201102

1.2751.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own
“Contributor” means any person or entity that distributes the Program.

“Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

“Program” means the Contributions distributed in accordance with this Agreement.

“Recipient” means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient’s responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS
A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient’s rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of MERCHANTABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----

Oracle OpenJDK
If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

-----

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

-----

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

-----

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2752 archy 1.0.0

1.2752.1 Available under license:

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the
License for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

Open Source Used In DNAC Platform Antelope 10868
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: HOW TO APPLY THE APACHE LICENSE TO YOUR WORK

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
1.2754 user-home 1.1.1
1.2754.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2755 platform 1.3.1 1.3.1
1.2755.1 Available under license :

Copyright 2014-2016 Benjamin Tan <https://d10.github.io/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
1.2756 extend 3.0.2

1.2756.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Stefan Thomas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2757 jline-builtins 3.12.1

1.2757.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2016, the original author or authors.
 *
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opensource.org/licenses/BSD-3-Clause
 */
Open Source Used In DNAC Platform Antelope 10872

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2016, the original author or authors.
 * 
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 *
 * https://opensource.org/licenses/BSD-3-Clause
 */

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/**
 * Based on http://antony.lesuisse.org/software/ajaxterm/
 * Public Domain License
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2018, the original author or authors.
 * 
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 * 
 */
* https://opensource.org/licenses/BSD-3-Clause
*/

Found in path(s):
* /opt/cola/permits/1114600151_1607533999.81/0/jline-builtins-3-12-1-sources-jar/org/jline/builtins/NfaMatcher.java
* /opt/cola/permits/1114600151_1607533999.81/0/jline-builtins-3-12-1-sources-jar/org/jline/builtins/Completers.java
* /opt/cola/permits/1114600151_1607533999.81/0/jline-builtins-3-12-1-sources-jar/org/jline/builtins/InputRC.java
* /opt/cola/permits/1114600151_1607533999.81/0/jline-builtins-3-12-1-sources-jar/org/jline/builtins/Less.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2017, the original author or authors.
 * 
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 * 
 * https://opensource.org/licenses/BSD-3-Clause
 */

Found in path(s):
* /opt/cola/permits/1114600151_1607533999.81/0/jline-builtins-3-12-1-sources-jar/org/jline/builtins/Nano.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2002-2018, the original author or authors.
 * 
 * This software is distributable under the BSD license. See the terms of the
 * BSD license in the documentation provided with this software.
 * 
 * https://opensource.org/licenses/BSD-3-Clause
 */

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1114600151_1607533999.81/0/jline-builtins-3-12-1-sources-jar/org/jline/builtins/Options.java
No license file was found, but licenses were detected in source scan.

2016, the original author or authors.

This software is distributable under the BSD license. See the terms of the BSD license in the documentation provided with this software.

https://opensource.org/licenses/BSD

Found in path(s):
* /opt/cola/permits/1114600151_1607533999.81/0/jline-builtins-3-12-1-sources-jar/META-INF/maven/org.jline/jline-builtins/pom.xml

1.2758 hive-standalone-metastore 3.1.2

1.2758.1 Available under license:

Hive Standalone Metastore
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2759 avro-mapred 1.9.1
1.2759.1 Available under license :
Apache Avro
Copyright 2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

Based upon the representations of upstream licensors, it is understood that portions of the mapreduce API included in the Java implementation are licensed from various contributors under one or more contributor license agreements to Odiago, Inc. and were then contributed by Odiago to Apache Avro, which has now made them available under the Apache 2.0 license. The original file header text is:

| Licensed to Odiago, Inc. under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. Odiago, Inc. licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at |
| https://www.apache.org/licenses/LICENSE-2.0 |
| Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. |

The Odiago NOTICE at the time of the contribution:

| This product includes software developed by Odiago, Inc. (https://www.wibidata.com). |

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2760 node-modules-regexp 1.0.0

1.2760.1 Available under license :

The MIT License (MIT)

Copyright (c) James Talmage <james@talmage.io> (github.com/jamestalmage)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2761 jetty-http 9.4.34.v20201102
1.2761.1 Available under license:

This program and the accompanying materials are made available under the
terms of the Eclipse Public License 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0, or the Apache Software License
2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation
distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution 'originates'
from a Contributor if it was added to the Program by such Contributor
itself or anyone acting on such Contributor's behalf. Contributions do not
include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.
3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property
infringement. In order to qualify, an Indemnified Contributor must:
a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the
Commercial Contributor in, the defense and any related settlement
negotiations. The Indemnified Contributor may participate in any such claim at
its own expense.

For example, a Contributor might include the Program in a commercial product
offering, Product X. That Contributor is then a Commercial Contributor. If
that Commercial Contributor then makes performance claims, or offers
warranties related to Product X, those performance claims and warranties are
such Commercial Contributor's responsibility alone. Under this section, the
Commercial Contributor would have to defend claims against the other
Contributors related to those performance claims and warranties, and if a
court requires any other Contributor to pay any damages as a result, the
Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR
IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE,
NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each
Recipient is solely responsible for determining the appropriateness of using
and distributing the Program and assumes all risks associated with its
exercise of rights under this Agreement, including but not limited to the
risks and costs of program errors, compliance with applicable laws, damage to
or loss of data, programs or equipment, and unavailability or interruption of
operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY
CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL,
sPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION
LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of the
remainder of the terms of this Agreement, and without further action by the
parties hereto, such provision shall be reformed to the minimum extent
necessary to make such provision valid and enforceable.
If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

==============================================================================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
==============================================================================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api
Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2762 batik-svgbrowser 1.13
1.2762.1 Available under license:
The following files are from the Tango Desktop Project (http://tango.freedesktop.org):

document-open.png
document-open.png
document-print.png
et-find.png
go-next.png
go-previous.png
icon-general.png(originally preferences-desktop.png)
icon-language.png(originally preferences-desktop-locale.png)
icon-network.png(originally applications-internet.png)
icon-security.png(originally dialog-warning.png)
icon-stylesheet.png(originally applications-graphics.png)
media-playback-pause.png
media-playback-start.png
process-stop.png
redo.png(modified version of edit-redo.png)
The following files are from the Pasodoble Icon Theme (http://www.jesusda.com/projects/pasodoble):

<table>
<thead>
<tr>
<th>File</th>
<th>Original filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>zoom-in.png</td>
<td>viewmag+.png</td>
</tr>
<tr>
<td>zoom-out.png</td>
<td>viewmag-.png</td>
</tr>
</tbody>
</table>

All of these image files are licensed under the Creative Commons Attribution Share-Alike 2.5 License (http://creativecommons.org/licenses/by-sa/2.5/), which is in the LICENSE.icons.txt file.

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
b. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
c. "Licensor" means the individual or entity that offers the Work under the terms of this License.
d. "Original Author" means the individual or entity who created the Work.
e. "Work" means the copyrightable work of authorship offered under the terms of this License.
f. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
g. "License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.

2. Fair Use Rights.

Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant.

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works; 
b. to create and reproduce Derivative Works; 
c. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works; 
d. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.
e. For the avoidance of doubt, where the work is a musical composition: 
   i. Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.
   ii. Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights society or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).
f. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right
to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients’ exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties.
You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(c), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(c), as requested.

b. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as this License (e.g. Attribution-ShareAlike 2.5 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients’ exercise of the rights granted hereunder, and You must keep intact all notices that refer to
this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.

c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, you must keep intact all copyright notices for the Work and provide, reasonable to the medium or means you are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., ”French translation of the Work by Original Author,” or ”Screenplay based on original Work by Original Author”). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE MATERIALS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.


EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination
a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Apache Batik
Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the
This software contains code from the International Organisation for Standardization for the definition of character entities used in the software's documentation.

This product includes images from the Tango Desktop Project (http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme (http://www.jesusda.com/projects/pasodoble).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same “printed page” as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2763 jsreport-freeze 2.0.0

1.2763.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2764 methods 1.1.2 1.1.2

1.2764.1 Available under license :

(The MIT License)

Copyright (c) 2013-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015-2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2765 debug 2.2.0 2.2.0

1.2765.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
1.2766 socket.io-parser 2.3.1

1.2766.1 Available under license:

(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2767 maxmind-db 1.4.0

1.2767.1 Available under license:

No license file was found, but licenses were detected in source scan.

<name>Apache License 2.0</name>
<url>http://www.apache.org/licenses/LICENSE-2.0.html</url>

Found in path(s):
* /opt/cola/permits/1140689161_1614870595.69/0/maxmind-db-1-4-0-sources-1-jar/META-INF/maven/com.maxmind.db/maxmind-db/pom.xml
Open Source Used In DNAC Platform Antelope 10908

1.2768 orc-shims 1.5.6
1.2768.1 Available under license:

ORC Shims
Copyright 2013-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2769 body-parser 1.18.2 1.18.2

1.2769.1 Available under license:
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2770 is-callable 1.1.4

1.2770.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2771 mime 1.3.4 1.3.4

1.2771.1 Available under license:
Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1.2772 yargs 15.4.1

1.2772.1 Available under license :
MIT License

Copyright 2010 James Halliday (mail@substack.net); Modified work Copyright 2014 Contributors (ben@npmjs.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2773 antlr 3.1.3

1.2773.1 Available under license :
[The "BSD licence"]

Copyright (c) 2008 Erik van Bilsen
Copyright (c) 2005-2007 Kunle Odutola
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code MUST RETAIN the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form MUST REPRODUCE the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior WRITTEN permission.

4. Unless explicitly state otherwise, any contribution intentionally submitted for inclusion in this work to the copyright owner shall be under the terms and conditions of this license, without any additional terms or conditions.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ANTLR 3 License

[The BSD License]
Copyright (c) 2010 Terence Parr
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

[The "BSD licence"]
Copyright (c) 2003-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

[The "BSD licence"]
Copyright (c) 2005-2007 Kunle Odutola
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code MUST RETAIN the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form MUST REPRODUCE the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the
distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior WRITTEN permission.
4. Unless explicitly state otherwise, any contribution intentionally
   submitted for inclusion in this work to the copyright owner shall be
   under the terms and conditions of this license, without any
   additional terms or conditions.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// [The "BSD licence"]
// Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
// http://www.temporal-wave.com
// http://www.linkedin.com/in/jimidle
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

[The "BSD licence"]
Copyright (c) 2007-2008 Leon, Jen-Yuan Su
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*
Copyright (c) 2003-2008 Terence Parr. All rights reserved.
Code licensed under the BSD License:
http://www.antlr.org/license.html

Some parts of the ANTLR class:
Copyright (c) 2008, Yahoo! Inc. All rights reserved.
Code licensed under the BSD License:
http://developer.yahoo.net/yui/license.txt
*/

1.2774 send 0.17.1
1.2774.1 Available under license:
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2775 jackson-databind 2.11.1

1.2775.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Open Source Used In DNAC Platform Antelope 10919
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   You must give any other recipients of the Work or Derivative Works a copy of this License; and
   You must cause any modified files to carry prominent notices stating that You changed the files; and
   You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
   If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2776 append-transform 2.0.0

1.2776.1 Available under license:

The MIT License (MIT)

Copyright (c) James Talmage <james@talmage.io> (github.com/jamestalmage)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2777 hive-serde 3.1.2
1.2777.1 Available under license:

Hive Serde
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2778 xmllsec-java 2.2.1
1.2778.1 Available under license:

xml-commons/java/external/README.dom.txt $Id: README.dom.txt 477038 2006-11-20 04:40:36Z mrglavas $

HEAR YE, HEAR YE!

All of the .java software and associated documentation about
the DOM in this repository are distributed under the license
from the W3C, which is provided herein.

LICENSE.dom-software.txt covers all software from the W3C
including the following items in the xml-commons project:

xml-commons/java/external/src/org/w3c
   and all subdirectories
(Note: SAC (The Simple API for CSS) has been published under an older version of the W3C license. The original license file is LICENSE.sac.html.)

LICENSE.dom-documentation.txt covers all documentation from the W3C including the following items in the xml-commons project:

xml-commons/java/external/xdocs/dom
and all subdirectories

The actual DOM Java Language Binding classes in xml-commons came from:
http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/java-binding.html

The specification of DOM Level 3’s various parts is at:
http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/
http://www.w3.org/TR/2004/REC-DOM-Level-3-LS-20040407/
http://www.w3.org/TR/2004/NOTE-DOM-Level-3-XPath-20040226/

The specification of DOM Level 2’s various parts is at:
http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113/
http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113/
http://www.w3.org/TR/2000/REC-DOM-Level-2-Traversal-Range-20001113/
http://www.w3.org/TR/2000/REC-DOM-Level-2-Views-20001113/

The specification of DOM Level 1’s various parts is at:

Links to all available W3C DOM Java Bindings can be found at:
http://www.w3.org/DOM/DOMTR

The actual classes of The Simple API for CSS (SAC) came from:
http://www.w3.org/Style/CSS/SAC/
http://www.w3.org/2002/06/sacjava-1.3.zip

The actual DOM Java Language Binding classes for SMIL came from:
http://dev.w3.org/cvsweb/java/classes/org/w3c/dom/smil/
(both ElementTimeControl.java and TimeEvent.java were taken at revision 1.1)

The actual DOM Java Language Binding classes for SVG 1.1 came from:
http://www.w3.org/TR/SVG11/java.html

=========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xerces distribution. ==
=========================================================================

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

========================================================================
==  NOTICE file corresponding to section 4(d) of the Apache License,   ==
==  Version 2.0, in this case for the Apache Xalan serializer          ==
==  distribution.                                                      ==
========================================================================

This product includes software developed by IBM Corporation (http://www.ibm.com)
and The Apache Software Foundation (http://www.apache.org/).

Portions of this software was originally based on the following:
This license came from: http://www.w3.org/Consortium/Legal/copyright-software-20021231

W3C SOFTWARE NOTICE AND LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at
This formulation of W3C’s notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". Otherwise, this version is the same as the previous version and is written so as to preserve the Free Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle $Date: 2005-06-03 18:49:13 -0400 (Fri, 03 Jun 2005) $

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

// ------------------------------------------------------------------
// NOTICE file corresponding to the section 4d of The Apache License,
// Version 2.0, in this case for Commons Logging
// ------------------------------------------------------------------

Commons Logging
Copyright 2001-2007 The Apache Software Foundation

This product includes/uses software(s) developed by 'an unknown organization'
- Unnamed - avalon-framework:avalon-framework:jar:4.1.3
- Unnamed - log4j:log4j:jar:1.2.12
- Unnamed - logkit:logkit:jar:1.0.1
Apache Santuario - XML Security for Java
Copyright 1999-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
It was originally based on software copyright (c) 2001, Institute for Data Communications Systems, <http://www.nue.et-inf.uni-siegen.de/>.

The development of this software was partly funded by the European Commission in the <WebSig> project in the ISIS Programme.


HEAR YE, HEAR YE!

All of the .java software and associated documentation about SAX in this repository are distributed freely in the public domain.

LICENSE.sax.txt covers all software and documentation from the megginson.com including the following in the xml-commons project:

  xml-commons/java/external/src/org/xml/sax
  and all subdirectories
  xml-commons/java/external/xdocs/sax
  and all subdirectories

The actual SAX classes in xml-commons came from:
  http://www.megginson.com/Software/index.html
The original versions are tagged 'SAX-2.0-r2-prerelease'

========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache xml-commons xml-apis ==
== distribution. ==
========================================================================

Apache XML Commons XML APIs
Copyright 2006 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

===========================================================================
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xalan distribution. ==
===========================================================================

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Portions of this software was originally based on the following:
- software copyright (c) 1999-2002, Lotus Development Corporation.,
- software copyright (c) 2001-2002, Sun Microsystems.,
- voluntary contributions made by Ovidiu Predescu <ovidiu@cup.hp.com> on
  behalf of the Apache Software Foundation that was originally developed
  at Hewlett Packard Company.

The binary distribution package (ie. jars, samples and documentation) of
this product includes software developed by the following:

- The Apache Software Foundation
  - Xerces Java - see xercesImpl.README.txt
  - JAXP 1.3 APIs - see xml-apis.README.txt
  - Bytecode Engineering Library - see BCEL.README.txt
  - Regular Expression - see regexp.README.txt

- Scott Hudson, Frank Flannery, C. Scott Ananian
  - CUP Parser Generator runtime (javacup)runtime) - see runtime.README.txt

The source distribution package (ie. all source and tools required to build
Xalan Java) of this product includes software developed by the following:

- The Apache Software Foundation
  - Xerces Java - see xercesImpl.README.txt
  - JAXP 1.3 APIs - see xml-apis.README.txt
  - Bytecode Engineering Library - see BCEL.README.txt
  - Regular Expression - see regexp.README.txt
  - Ant - see ant.README.txt
  - Stylebook doc tool - see stylebook-1.0-b3_xalan-2.README.txt
Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.


This license came from: http://www.w3.org/Consortium/Legal/copyright-documents-20021231

W3C DOCUMENT LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231

Public documents on the W3C site are provided by the copyright holders under
the following license. By using and/or copying this document, or the W3C
document from which this statement is linked, you (the licensee) agree that
you have read, understood, and will comply with the following terms and
conditions:

Permission to copy, and distribute the contents of this document, or the W3C
document from which this statement is linked, in any medium for any purpose
and without fee or royalty is hereby granted, provided that you include the
following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it
doesn't exist, a notice (hypertext is preferred, but a textual
representation is permitted) of the form: "Copyright [Sdate-of-document]
World Wide Web Consortium, (Massachusetts Institute of Technology,
European Research Consortium for Informatics and Mathematics, Keio
University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

========================================================================

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle $Date: 2005-06-03 18:49:13 -0400 (Fri, 03 Jun 2005) $

========================================================================

== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache xml-commons xml-apis ==
This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered under http://saxproject.org/?selected=pd

This page is now out of date -- see the new SAX site at http://www.saxproject.org/ for more up-to-date releases and other information. Please change your bookmarks.

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
1.2779.1 Available under license:
The MIT License (MIT)

Copyright (c) 2017 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2780 jnr-ffi 2.1.5

1.2781 findup-sync 3.0.0
1.2781.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013-2018 Ben Alman <cowboy@rj3.net>, Blaine Bublitz <blaine.bublitz@gmail.com>, and Eric Schoffstall <yo@contra.io>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

### 1.2782 scala 2.11.0

#### 1.2782.1 Available under license:

No license file was found, but licenses were detected in source scan.

```scala
/**
 * Copyright (C) 2012 FuseSource, Inc.
 * http://fusesource.com
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *    http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Found in path(s):
- `/opt/cola/permits/1104749649_1611239463.38/0/hawtdispatch-scala-1-21-sources.jar/org/fusesource/hawtdispatch.scala`
- `/opt/cola/permits/1104749649_1611239463.38/0/hawtdispatch-scala-1-21-sources.jar/org/fusesource/hawtdispatch/ScalaDispatchHelpers.scala`
- `/opt/cola/permits/1104749649_1611239463.38/0/hawtdispatch-scala-1-21-sources.jar/org/fusesource/hawtdispatch/ListEventAggregator.scala`
- `/opt/cola/permits/1104749649_1611239463.38/0/hawtdispatch-scala-1-21-sources.jar/org/fusesource/hawtdispatch/TaskTracker.scala`
- `/opt/cola/permits/1104749649_1611239463.38/0/hawtdispatch-scala-1-21-sources.jar/org/fusesource/hawtdispatch/Future.scala`

### 1.2783 j2objc-annotations 1.3
ICU License - ICU 1.8.1 and later

Copyright (c) 1995-2015 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be
used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

<p></p>

<hr style="color:gray;background-color:gray">

<p><small>All trademarks and registered trademarks mentioned herein are the property of their respective owners.</small></p>

<hr style="height:3px;color:black;background-color:black">

<h2>Third-Party Software Licenses</h2>

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

<h3>1. Unicode Data Files and Software</h3>

<pre>COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2015 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) this copyright and permission notice appear with all copies of the Data Files or Software, (b) this copyright and permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.</p>

<h3>2. Chinese/Japanese Word Break Dictionary Data (cjdict.txt)</h3>

The Google Chrome software developed by Google is licensed under the BSD license. Other software included in this distribution is provided under other licenses, as set forth below.

# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
# All rights reserved.

# Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
# Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
# Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# The word list in cjdict.txt are generated by combining three word lists listed below with further processing for compound word breaking. The frequency is generated with an iterative training against Google web corpora.
# Its license terms and conditions are shown below.
# IPADIC (Japanese)
# - http://chasen.aist-nara.ac.jp/chasen/distribution.html
# - Its license terms and conditions are shown below.
#
#--------COPYING.libtabe ---- BEGIN--------------
#
#/ *
#/ * Copyright (c) 1999 TaBE Project.
#/ * Copyright (c) 1999 Pai-Hsiang Hsiao.
#/ * All rights reserved.
#/ *
#/ * Redistribution and use in source and binary forms, with or without
#/ * modification, are permitted provided that the following conditions
#/ * are met:
#/ *
#/ * . Redistributions of source code must retain the above copyright
#/ * notice, this list of conditions and the following disclaimer.
#/ * . Redistributions in binary form must reproduce the above copyright
#/ * notice, this list of conditions and the following disclaimer in
#/ * the documentation and/or other materials provided with the
#/ * distribution.
#/ * Neither the name of the TaBE Project nor the names of its
#/ * contributors may be used to endorse or promote products derived
#/ * from this software without specific prior written permission.
#/ *
#/ * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#/ * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#/ * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#/ * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#/ * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#/ * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#/ * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#/ * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#/ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#/ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#/ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
#/ * OF THE POSSIBILITY OF SUCH DAMAGE.
#/ */
#
#/*
#/ *
#/ * Copyright (c) 1999 Computer Systems and Communication Lab,
#/ * Institute of Information Science, Academia Sinica.
#/ * All rights reserved.
#/ *
#/ * Redistribution and use in source and binary forms, with or without
#/ * modification, are permitted provided that the following conditions
#/ * are met:
 Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 Neither the name of the Computer Systems and Communication Lab nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

 Copyright 1996 Chih-Hao Tsai @ Beckman Institute, University of Illinois
c-tsai4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsai4


 Use, reproduction, and distribution of this software is permitted.
 Any copy of this software, whether in its original form or modified, must include both the above copyright notice and the following paragraphs.

 Nara Institute of Science and Technology (NAIST), the copyright holders, disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall NAIST be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

# NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.
In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

---------------COPYING.ipadic-----END------------------------------------

<h3>3. Lao Word Break Dictionary Data (laodict.txt)</h3>
<pre>
# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
# (copied below)
#
# This file is derived from the above dictionary, with slight modifications.
# -----------------------------------------------
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without modification,
# are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in binary
# form must reproduce the above copyright notice, this list of conditions and
# the following disclaimer in the documentation and/or other materials
# provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
# ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
# ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
# LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
# (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# -----------------------------------------------
</pre>
4. Burmese Word Break Dictionary Data (burmesedict.txt)

COPYRIGHT (c) 2014 INTERNATIONAL BUSINESS MACHINES CORPORATION

This list is part of a project hosted at:

github.com/kanyawtech/myanmar-karen-word-lists

COPYRIGHT (c) 2013, LEROY BENJAMIN SHARON

ALL RIGHTS RESERVED.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

Neither the name Myanmar Karen Word Lists, nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.
The TZ database itself is not an IETF Contribution or an IETF document. Rather it is a pre-existing and regularly updated work that is in the public domain, and is intended to remain in the public domain. Therefore, BCPs 78 [<a href="http://tools.ietf.org/html/rfc5378">RFC5378</a>] and 79 [<a href="http://tools.ietf.org/html/rfc3979">RFC3979</a>] do not apply to the TZ Database or contributions that individuals make to it. Should any claims be made and substantiated against the TZ Database, the organization that is providing the IANA Considerations defined in this RFC, under the memorandum of understanding with the IETF, currently ICANN, may act in accordance with all competent court orders. No ownership claims will be made by ICANN or the IETF Trust on the database or the code. Any person making a contribution to the database or code waives all rights to future claims in that contribution or in the TZ Database.

</pre>
</body>
</html>

## -*-makefile-*-
## BSD-specific setup (FreeBSD, OpenBSD, NetBSD, *BSD)
## Copyright (c) 1999-2013, International Business Machines Corporation and others. All Rights Reserved.

## Commands to generate dependency files
GEN_DEPS.c:=$(CC) -E -MM $(DEFS) $(CPPFLAGS)
GEN_DEPS.cc:=$(CXX) -E -MM $(DEFS) $(CPPFLAGS)

## Flags for position independent code
SHAREDLIBCFLAGS = -fPIC
SHAREDLIBCXXFLAGS = -fPIC
SHAREDLIBCPPFLAGS = -DPIC

## Additional flags when building libraries and with threads
THREADSCPPFLAGS = -D_REENTRANT
LIBCPPFLAGS =

## Compiler switch to embed a runtime search path
LD_RPATH=
LD_RPATH_PRE=-Wl,.-rpath,

## Compiler switch to embed a library name
LD_SONAME = -Wl,-soname -Wl,$(notdir $(MIDDLE_SO_TARGET))

## Shared library options
LD_SOOPTIONS= -Wl,-Bsymbolic
## Shared object suffix
SO = so

## Non-shared intermediate object suffix
STATIC_O = ao

## Compilation rules
%.$(STATIC_O): $(srcreldir)/%.c  
$(COMPILE.c) $(STATICCPPFLAGS) $(STATICCFLAGS) -o $@ $<  
%.o: $(srcreldir)/%.c  
$(COMPILE.c) $(DYNAMICCPPFLAGS) $(DYNAMICCFLAGS) -o $@ $<  
%.$(STATIC_O): $(srcreldir)/%.cpp  
$(COMPILE.cc) $(STATICCPPFLAGS) $(STATICCXXFLAGS) -o $@ $<  
%.o: $(srcreldir)/%.cpp  
$(COMPILE.cc) $(DYNAMICCPPFLAGS) $(DYNAMICCXXFLAGS) -o $@ $<

## Dependency rules
%.d: $(srcreldir)/%.c  
@echo "generating dependency information for $<"  
@s$(SHELL) -ec "$(GEN_DEPS.c) $< \  
| sed \"s/(\%\*\*)\(.\)\([0-9]*\)\(.*\)/\1\$@ : /g\" > $@; \  
| -s $@ ] || rm -f $@"

%.d: $(srcreldir)/%.cpp  
@echo "generating dependency information for $<"  
@s$(SHELL) -ec "$(GEN_DEPS.cc) $< \  
| sed \"s/(\%\*\*)\(.\)\([0-9]*\)\(.*\)/\1\$@ : /g\" > $@; \  
| -s $@ ] || rm -f $@"

## Versioned libraries rules
%.$(SO).$(SO_TARGET_VERSION_MAJOR): %.$(SO).$(SO_TARGET_VERSION)  
$(RM) $@ && ln -s ${<F} $@

%.$(SO): %.$(SO).$(SO_TARGET_VERSION_MAJOR)  
$(RM) $@ && ln -s ${<F} $(SO).$(SO_TARGET_VERSION) $@

## Bind internal references

# LDflags that pkgdata will use  
BIR_LDFLAGS= -Wl,-Bsymb

# Dependencies [i.e. map files] for the final library  
BIR_DEPS=

## Remove shared library 's'
STATIC_PREFIX_WHEN_USED =
1.2785 is-installed-globally 0.3.2

1.2785.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2786 interpret 1.4.0

1.2786.1 Available under license:

Copyright (c) 2014-2018 Tyler Kellen <tyler@sleekcode.net>, Blaine Bublitz <blaine.bublitz@gmail.com>, and Eric Schoffstall <yo@contra.io>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

1.2787 jersey 2.32

1.2787.1 Available under license :

# Notice for Jersey Json Jackson module
This content is produced and maintained by the Eclipse Jersey project.

* https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright
All content is the property of the respective authors or their employers. For
more information regarding authorship of content, please consult the listed
source code repository logs.

## Declared Project Licenses
This program and the accompanying materials are made available under the terms
of the Eclipse Public License v. 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made
available under the following Secondary Licenses when the conditions for such
availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU
General Public License, version 2 with the GNU Classpath Exception which is

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content
Jackson JAX-RS Providers version 2.10.1
* License: Apache License, 2.0
* Project: https://github.com/FasterXML/jackson-jaxrs-providers
* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.
/*
 * Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
 *
 * This program and the accompanying materials are made available under the
 * terms of the Eclipse Public License v. 2.0, which is available at
 *
 * This Source Code may also be made available under the following Secondary
 * Licenses when the conditions for such availability set forth in the
 * Eclipse Public License v. 2.0 are satisfied: GNU General Public License,
 * version 2 with the GNU Classpath Exception, which is available at
 *
 * SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0
 */

## Notice for Jersey
This content is produced and maintained by the Eclipse Jersey project.

* Project home: https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright
All content is the property of the respective authors or their employers. For
more information regarding authorship of content, please consult the listed
source code repository logs.

## Declared Project Licenses
This program and the accompanying materials are made available under the terms
of the Eclipse Public License v. 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made
available under the following Secondary Licenses when the conditions for such
availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU
General Public License, version 2 with the GNU Classpath Exception which is

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content
Angular JS, v1.6.6
* License: MIT (http://www.opensource.org/licenses/mit-license.php)
* Project: http://angularjs.org
* Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1
* License: all the source code provided by AOP Alliance is Public Domain.
* Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2
* License: Apache License, 2.0
* Project: http://beanvalidation.org/1.1/
* Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
  * by the @authors tag.

Hibernate Validator CDI, 6.1.2.Final
* License: Apache License, 2.0
* Project: https://beanvalidation.org/
* Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

Bootstrap v3.3.7
* License: MIT license (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* Project: http://getbootstrap.com
* Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0
* License: Apache License, 2.0
* Copyright (C) 2009 The Guava Authors

javax.inject Version: 1
* License: Apache License, 2.0
* Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA
* License: Apache License, 2.0
* Project: http://www.javassist.org/
* Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.10.1
* License: Apache License, 2.0
* Project: https://github.com/FasterXML/jackson-jaxrs-providers
* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4
* License: jquery.org/license
* Project: jquery.org
* Copyright: (c) jQuery Foundation
jQuery Barcode plugin 0.3
* Project: http://www.pasella.it/projects/jQuery/barcode
* Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266
* License: CC0
* No copyright
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

KineticJS, v4.7.1
* License: MIT license (http://www.opensource.org/licenses/mit-license.php)
* Copyright: Eric Rowell

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0
* License: Apache License, 2.0
* Copyright (c) OSGi Alliance (2005, 2008). All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents
* License: W3C License
* Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/
/*
* Copyright (c) YYYY Oracle and/or its affiliates. All rights reserved.
*
* This program and the accompanying materials are made available under the
* terms of the Eclipse Distribution License v. 1.0, which is available at
*
* SPDX-License-Identifier: BSD-3-Clause
*/
# Notice for Jersey Core Server module
This content is produced and maintained by the Eclipse Jersey project.
## Trademarks

Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright: (c) 2000-2011 INRIA, France Telecom. All rights reserved.

W3.org documents
* License: W3C License
* Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/

# Notice for Jersey Core Common module
This content is produced and maintained by the Eclipse Jersey project.
* https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content

Google Guava Version 18.0
* License: Apache License, 2.0
* Copyright: (C) 2009 The Guava Authors

JSR-166 Extension - JEP 266
* License: Creative Commons 1.0 (CC0)
* No copyright
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
* Notice for Jersey
This content is produced and maintained by the Eclipse Jersey project.

* Project home: https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.
## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0 which is available at https://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:

- https://github.com/eclipse-ee4j/jersey/examples

## Third-party Content

Angular JS, v1.6.6
- License MIT (http://www.opensource.org/licenses/mit-license.php)
- Project: http://angularjs.org
- Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1
- License: all the source code provided by AOP Alliance is Public Domain.
- Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2
- License: Apache License, 2.0
- Project: http://beanvalidation.org/1.1/
- Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
  - by the @authors tag.

Bootstrap v3.3.7
- License: MIT license (https://github.com/twbs/bootstrap/blob/master/LICENSE)
- Project: http://getbootstrap.com
- Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0
- License: Apache License, 2.0
- Copyright (C) 2009 The Guava Authors

javax.inject Version: 1
- License: Apache License, 2.0
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
   Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified
Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the
d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

   iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial
Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.
5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses
granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program
proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed
only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (which makes passes at compilers) written by
James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications
with the library. If this is what you want to do, use the GNU Library
General Public License instead of this License.

---

## CLASSPATH EXCEPTION
Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Notice for Jersey Bean Validation module
This content is produced and maintained by the Eclipse Jersey project.
* https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright
All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses
This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content
Hibernate Validator CDI, 6.1.2.Final
* License: Apache License, 2.0
* Project: https://beanvalidation.org/
* Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

1.2788 lucene-queryparser 7.4.0

1.2788.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

/*
 * Copyright 2001-2004 Unicode, Inc.
 *
 * Disclaimer
 *
 * This source code is provided as is by Unicode, Inc. No claims are
 * made as to fitness for any particular purpose. No warranties of any
 * kind are expressed or implied. The recipient agrees to determine
 * applicability of information provided. If this file has been
 * purchased on magnetic or optical media from Unicode, Inc., the
 * sole remedy for any claim will be exchange of defective media
 * within 90 days of receipt.
 *
 * Limitations on Rights to Redistribute This Code
 *
 * Unicode, Inc. hereby grants the right to freely use the information
 * supplied in this file in the creation of products supporting the
 * Unicode Standard, and to make copies of this file in any form
 * for internal or external distribution as long as this notice
 * remains attached.
 */

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was derived from Brics automaton sources available at www.brics.dk/automaton/. Here is the copyright from those sources:
The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton were automatically generated with the moman/finenight FSA package. Here is the copyright for those sources:

# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
 * Copyright (C) 1999-2010, International Business Machines
 * Corporation and others. All Rights Reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, and/or sell copies of the
 * Software, and to permit persons to whom the Software is furnished to do so,
 * provided that the above copyright notice(s) and this permission notice appear
 * in all copies of the Software and that both the above copyright notice(s) and
 * this permission notice appear in supporting documentation.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
 * IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
 * LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
 * ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
 * IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
 * OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * Except as contained in this notice, the name of a copyright holder shall not
 * be used in advertising or otherwise to promote the sale, use or other
 * dealings in this Software without prior written authorization of the
 * copyright holder.
 */

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice,
  * this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  * notice, this list of conditions and the following disclaimer in the
  * documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
  * may be used to endorse or promote products derived from this software
  * without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and
"University of Massachusetts" must not be used to endorse or promote products
   derived from this software without prior written permission. To obtain
   permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---

The dictionary comes from Morfologik project. Morfologik uses data from
Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and
is licenced on the terms of (inter alia) LGPL and Creative Commons
ShareAlike. The part-of-speech tags were added in Morfologik project and
are not found in the data from sjp.pl. The tagset is similar to IPI PAN
tagset.

---
The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski, Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects, including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces
ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
The German, Spanish, Finnish, French, Hungarian, Italian, Portuguese, Russian and Swedish light stemmers (common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:

- analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
- analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil, and Edmond Nolan.

The Polish analyzer (stempel) comes with a default stopword list that is BSD-licensed created by the Carrot2 project. The file resides in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt. See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/) is derived from Unicode data such as the Unicode Character Database. See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary (http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP) (http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original source code for this can be found at http://www.eclipse.org/jetty/downloads.php.
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

mecab-ipadic-2.7.0-20070801 Notice

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

===========================================================================
Nori Korean Morphological Analyzer - Apache Lucene Integration
===========================================================================

This software includes a binary and/or source version of data from

mecab-ko-dic-2.0.3-20170922

which can be obtained from

https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.0.3-20170922.tar.gz
1.2789 http-errors 1.3.1
1.2789.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2790 iconv-lite 0.4.19 0.4.19
1.2790.1 Available under license:

Copyright (c) 2011 Alexander Shtuchkin

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
1.2791 accepts 1.3.5 1.3.5

1.2791.1 Available under license:
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2792 regenerator-runtime 0.13.7

1.2792.1 Available under license:
MIT License

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2793 babel-plugin-transform-es2015-shorthand-properties 6.24.1

1.2793.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2794 asn1 0.2.4
1.2794.1 Available under license :
Copyright (c) 2011 Mark Cavage, All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2795 inherits 2.0.4 2.0.4
1.2795.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2796 trim-right 1.0.1 1.0.1

1.2796.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2797 twill-common 0.6.0-incubating
1.2797.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1137746870_1614121577.24/0/twill-common-0-6-0-incubating-sources-2-jar/org/apache/twill/filesystem/LocationFactories.java
* /opt/cola/permits/1137746870_1614121577.24/0/twill-common-0-6-0-incubating-sources-2-jar/org/apache/twill/common/Threads.java
* /opt/cola/permits/1137746870_1614121577.24/0/twill-common-0-6-0-incubating-sources-2-jar/org/apache/twill/filesystem/ForwardingLocationFactory.java
* /opt/cola/permits/1137746870_1614121577.24/0/twill-common-0-6-0-incubating-sources-2-jar/org/apache/twill/common/LocalLocation.java
* /opt/cola/permits/1137746870_1614121577.24/0/twill-common-0-6-0-incubating-sources-2-jar/org/apache/twill/common/Cancellable.java
* /opt/cola/permits/1137746870_1614121577.24/0/twill-common-0-6-0-incubating-sources-2-jar/org/apache/twill/filesystem/LocalLocationFactory.java
* /opt/cola/permits/1137746870_1614121577.24/0/twill-common-0-6-0-incubating-sources-2-jar/org/apache/twill/filesystem/Location.java
* /opt/cola/permits/1137746870_1614121577.24/0/twill-common-0-6-0-incubating-sources-2-jar/org/apache/twill/filesystem/LocationFactory.java

1.2798 xmlbuilder 11.0.1 11.0.1

1.2798.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013 Ozgur Ozcitak

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2799 jsreport-debug 2.1.3
1.2799.1 Available under license :
The MIT License (MIT)

Copyright (c) 2016 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2800 intl-format-cache 2.1.0 2.1.0
1.2800.1 Available under license:

Copyright 2014 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2801 engine.io 3.4.2

1.2801.1 Available under license:

(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE
FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2802 async 2.6.0 2.6.0
1.2802.1 Available under license :
Copyright (c) 2010-2017 Caolan McMahon

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2803 got 2.4.0 2.4.0
1.2803.1 Available under license :
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
1.2804 @babel/plugin-syntax-logical-assignment-operators 7.10.4
1.2804.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2805 my-scope-package-b 0.0.0
1.2805.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.
Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The JsonCpp library's source code, including accompanying documentation, tests and demonstration applications, are licensed under the following conditions...

The author (Baptiste Lepilleur) explicitly disclaims copyright in all jurisdictions which recognize such a disclaimer. In such jurisdictions, this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of 2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur, and is released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this software may choose to accept it either as 1) Public Domain, 2) under the conditions of the MIT License (see below), or 3) under the terms of dual Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is described in clear, concise terms at:

http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

========================================================================
Copyright (c) 2007-2010 Baptiste Lepilleur
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT license is compatible with both the GPL and commercial software, affording one all of the rights of Public Domain with the minor nuisance of being required to keep the above copyright notice and license text in the source code. Note also that by accepting the Public Domain "license" you can re-license your copy using whatever license you like.

All files in this directory are provided by the following license if not stated otherwise in the individual file:

This file is provided under a dual BSD/GPLv2 license. When using or redistributing this file, you may do so under either license.

GPL LICENSE SUMMARY

Copyright (c) 2005-2012 Intel Corporation. All rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St - Fifth Floor, Boston, MA 02110-1301 USA.
The full GNU General Public License is included in this distribution
in the file called LICENSE.GPL.

Contact Information:

BSD LICENSE

Copyright (c) 2005-2012 Intel Corporation. All rights reserved.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the
distribution.
* Neither the name of Intel Corporation nor the names of its
  contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

====================
This license applies to all parts of V8 that are not externally
maintained libraries. The externally maintained libraries used by V8
are:

- PCRE test suite, located in
test/mjsunit/third_party/regexp-pcre/regexp-pcre.js. This is based on the
test suite from PCRE-7.3, which is copyrighted by the University
of Cambridge and Google, Inc. The copyright notice and license
are embedded in regexp-pcre.js.
- Layout tests, located in test/mjsunit/third_party/object-keys. These are based on layout tests from webkit.org which are copyrighted by Apple Computer, Inc. and released under a 3-clause BSD license.

- Strongtalk assembler, the basis of the files assembler-arm-inl.h, assembler-arm.cc, assembler-arm.h, assembler-ia32-inl.h, assembler-ia32.cc, assembler-ia32.h, assembler-x64-inl.h, assembler-x64.cc, assembler-x64.h, assembler-mips-inl.h, assembler-mips.cc, assembler-mips.h, assembler.cc and assembler.h. This code is copyrighted by Sun Microsystems Inc. and released under a 3-clause BSD license.

- Valgrind client API header, located at src/third_party/valgrind/valgrind.h This is released under the BSD license.

- The Wasm C/C++ API headers, located at third_party/wasm-api/wasm.{h,hh} This is released under the Apache license. The API's upstream prototype implementation also formed the basis of V8's implementation in src/wasm/c-api.cc.

These libraries have their own licenses; we recommend you read them, as their terms may differ from the terms below.

Further license information can be found in LICENSE files located in sub-directories.

Copyright 2014, the V8 project authors. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1994-2006 Sun Microsystems Inc.
All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Sun Microsystems or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

CC0 1.0 Universal

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any
purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;

ii. moral rights retained by the original author(s) and/or performer(s);

iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;

iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;

v. rights protecting the extraction, dissemination, use and reuse of data in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time
extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

   a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.

   b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not discoverable, all to the greatest extent permissible under applicable law.

   c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work or any use thereof, including without limitation any person's Copyright and Related Rights in the Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents, permissions or other rights required for any use of the Work.
d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and has no duty or obligation with respect to this CC0 or use of the Work.

For more information, please see <http://creativecommons.org/publicdomain/zero/1.0/>

Tests included here are based on build output generated by the six-speed benchmark suite.

Copyright (c) 2015 Kevin Decker

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

----------------------------------------------------------------

Notice that the following BSD-style license applies to this one file (valgrind.h) only. The rest of Valgrind is licensed under the terms of the GNU General Public License, version 2, unless otherwise indicated. See the COPYING file in the source distribution for details.

----------------------------------------------------------------

This file is part of Valgrind, a dynamic binary instrumentation framework.

Copyright (C) 2000-2010 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

----------------------------------------------------------------

Notice that the above BSD-style license applies to this one file (valgrind.h) only. The entire rest of Valgrind is licensed under the terms of the GNU General Public License, version 2. See the COPYING file in the source distribution for details.

----------------------------------------------------------------

// Copyright 2016 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PCRE LICENCE
-------------

PCRE is a library of functions to support regular expressions whose syntax
and semantics are as close as possible to those of the Perl 5 language.

Release 7 of PCRE is distributed under the terms of the "BSD" licence, as
specified below. The documentation for PCRE, supplied in the "doc"
directory, is distributed under the same terms as the software itself.

The basic library functions are written in C and are freestanding. Also
included in the distribution is a set of C++ wrapper functions.

THE BASIC LIBRARY FUNCTIONS
-------------------------------

Written by:    Philip Hazel
Email local part: ph10
Email domain:    cam.ac.uk

University of Cambridge Computing Service,

Copyright (c) 1997-2007 University of Cambridge
All rights reserved.

THE C++ WRAPPER FUNCTIONS
---------------------------

Contributed by:  Google Inc.

Copyright (c) 2007, Google Inc.
All rights reserved.
THE "BSD" LICENCE
----------------

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

End
Copyright (c) 2006 Apple Computer, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder(s) nor the names of any
contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2010 Jonathan Hartley
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the name of the copyright holders, nor those of its contributors
  may be used to endorse or promote products derived from this software without
  specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A. HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting
Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands
as a successor of a language called ABC. Guido remains Python's
principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

<table>
<thead>
<tr>
<th>Release</th>
<th>Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td></td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.2 and above</td>
<td>2.1.1</td>
<td>2001-now</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.
Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

**B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON**

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.


3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and
Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0
-------------------------------------------

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant
permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1
---------------------------------------

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT
5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

---------------------------------------------------------------------

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,
The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Copyright (C) 1993-2004 by Sun Microsystems, Inc. All rights reserved.

Developed at SunSoft, a Sun Microsystems, Inc. business.
Permission to use, copy, modify, and distribute this
software is freely granted, provided that this notice
is preserved.

----------------------------------------------------------

Notice that the following BSD-style license applies to this one
file (valgrind.h) only. The rest of Valgrind is licensed under the
terms of the GNU General Public License, version 2, unless
otherwise indicated. See the COPYING file in the source
distribution for details.

----------------------------------------------------------

This file is part of Valgrind, a dynamic binary instrumentation
framework.

Copyright (C) 2000-2007 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must
   not claim that you wrote the original software. If you use this
   software in a product, an acknowledgment in the product
   documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must
   not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior written
   permission.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

W3C 3-clause BSD License

http://www.w3.org/Consortium/Legal/2008/03-bsd-license.html

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of works must retain the original copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the original copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the W3C nor the names of its contributors may be used to endorse or promote products derived from this work without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2006-2011, the V8 project authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2806 cliui 5.0.0

1.2806.1 Available under license :

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2807 bouncy-castle 1.60

1.2807.1 Available under license :

1I)+()V<clinit><init>[Copyright (c) 2000-2017 The Legion of the Bouncy Castle Inc.(http://www.bouncycastle.org) DEALINGS IN THE SOFTWARE.XINCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR|LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2809 node-preload 0.2.1

1.2809.1 Available under license:
MIT License

Copyright (c) 2019 CFWare, LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2810 shebang-regex 3.0.0

1.2810.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2811 protostuff-json 1.7.1

1.2811.1 Available under license:
No license file was found, but licenses were detected in source scan.

//Licensed under the Apache License, Version 2.0 (the "License”);
//you may not use this file except in compliance with the License.
//You may obtain a copy of the License at
//http://www.apache.org/licenses/LICENSE-2.0
//distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/JsonOutput.java
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/JsonInput.java
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/JsonXIOUtil.java
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/SmileIOUtil.java
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/JsonXOutput.java
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/JsonIOUtil.java
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/JsonXOutput.java
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/JsonIOUtil.java
* /opt/cola/permits/1202446833_1631568401.51/0/protostuff-json-1-7-1-sources-jar/io/protostuff/JsonInputException.java
1.2813 jackson-annotations 2.10.5

1.2813.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the
Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2814 aop-alliance 1.0
1.2814.1 Available under license :
Public Domain

1.2815 argparse 1.0.10 1.0.10
1.2815.1 Available under license :
(The MIT License)

Copyright (C) 2012 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2816 hbase-procedure 2.4.2
1.2816.1 Available under license :

Apache HBase - Procedure
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

Open Source Used In DNAC Platform Antelope 11042
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2817 config 1.0.0

1.2817.1 Available under license:

The MIT License (MIT)
Copyright 2015 Hassan Khan <contact@hassankhan.me>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the Software), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2818 mime 1.3.4
1.2818.1 Available under license:
Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2819 readdirp 3.5.0
1.2819.1 Available under license:
MIT License

Copyright (c) 2012-2019 Thorsten Lorenz, Paul Miller (https://paulmillr.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2820 netty-codec-socks 4.1.42.Final

1.2820.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871333_1595214884.69/0/netty-codec-socks-4-1-42-final-sources-1-
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ http://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 */
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
*   http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1072871333_1595214884.69/0/netty-codec-socks-4-1-42-final-sources-1.jar/io/netty/handler/codec/socks/SocksMessageType.java
* /opt/ws_local/PERMITS_SQL/1072871333_1595214884.69/0/netty-codec-socks-4-1-42-final-sources-1.jar/io/netty/handler/codec/socks/SocksRequestType.java
* /opt/ws_local/PERMITS_SQL/1072871333_1595214884.69/0/netty-codec-socks-4-1-42-final-sources-1.jar/io/netty/handler/codec/socks/SocksAddressType.java
* /opt/ws_local/PERMITS_SQL/1072871333_1595214884.69/0/netty-codec-socks-4-1-42-final-sources-1.jar/io/netty/handler/codec/socks/SocksResponseType.java
* /opt/ws_local/PERMITS_SQL/1072871333_1595214884.69/0/netty-codec-socks-4-1-42-final-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

1.2821 urijs 1.19.6 1.19.6

1.2821.1 Available under license:

The MIT License (MIT)

Copyright (c) 2011 Rodney Rehm

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2822 electron-to-chromium 1.3.55 1.3.55

1.2822.1 Available under license:
Copyright 2018 Kilian Valkhof

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHERWISE, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2823 htrace-core4 4.1.0-incubating

1.2823.1 Available under license:
htrace-core4
Copyright 2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Apache Avro Mapred API
Copyright 2009-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

(an example is provided in the Appendix below).
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same “printed page” as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.2825 kerb-admin 1.0.0
1.2825.1 Available under license:

Kerby-kerb Admin
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2826 wrap-ansi 7.0.0 7.0.0
1.2826.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2827 locate-path 3.0.0
1.2827.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2828 @exodus/schemasafe 1.0.0-rc.3 1.0.0-rc.3
1.2828.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Mathias Buus
Copyright (c) 2020 Exodus Movement

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2829 media-typer 0.3.0 0.3.0

1.2829.1 Available under license:
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
1.2830 batik-codec 1.13

1.2830.1 Available under license:

No license file was found, but licenses were detected in source scan.

# -----------------------------------------------------------------------------
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#   http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# This file registers PNG and JPEG codecs.
#
# $Id: org.apache.batik.ext.awt.image.spi.RegistryEntry 1790950 2017-04-11 10:54:26Z ssteiner $
# -----------------------------------------------------------------------------

org.apache.batik.ext.awt.image.codec.png.PNGRegistryEntry

org.apache.batik.ext.awt.image.codec.imageio.ImageIOJPEGRegistryEntry
org.apache.batik.ext.awt.image.codec.imageio.ImageIOPNGRegistryEntry
org.apache.batik.ext.awt.image.codec.imageio.ImageIOTIFFRegistryEntry

Found in path(s):
* /opt/cola/permits/1146878584_1616835244.27/0/batik-codec-1-13-sources-1-jar/META-INF/services/org.apache.batik.ext.awt.image.spi.RegistryEntry

No license file was found, but licenses were detected in source scan.

# -----------------------------------------------------------------------------
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# This file registers PNG and JPEG writer classes.
#
# $Id: org.apache.batik.ext.awt.image.spi.ImageWriter 1733416 2016-03-03 07:07:13Z gadams $
# -------------------------

# NOTE: the "codec" package is deprecated, there entries are kept here for compatibility with older JVM versions
# (uses "sun.image", which is only supported in Sun Java implementations and was retired in JDK 7)
# org.apache.batik.ext.awt.image.codec.png.PNGImageWriter

org.apache.batik.ext.awt.image.codec.imageio.ImageIOPNGImageWriter
org.apache.batik.ext.awt.image.codec.imageio.ImageIOTIFFImageWriter
org.apache.batik.ext.awt.image.codec.imageio.ImageIOJPEGImageWriter

Found in path(s):

No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/
Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):

1.2831 nocache 1.0.1
1.2831.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2016 Evan Hahn, Adam Baldwin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2832 compressible 2.0.14 2.0.14

1.2832.1 Available under license:

(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Jeremiah Senkpiel <fishrock123@rocketmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
1.2833 babel-helper-bindify-decorators 6.24.1

1.2833.1 Available under license:

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

MIT License

Copyright (c) 2014-2017 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2834 http-cache-semantics 4.1.0
1.2834.1 Available under license :

Copyright 2016-2018 Kornel Lesiski

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2835 request-promise-core 1.1.4
1.2835.1 Available under license :
ISC License

Copyright (c) 2020, Nicolai Kamenzky and contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2836 json-glib 1.1.2-0ubuntu1
1.2836.1 Available under license :
GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.
Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library’s complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

      (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.
When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.
You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>AppendixA.License: JSON-GLib Reference Manual</title>
<meta name="generator" content="DocBook XSL Stylesheets V1.78.1">
<link rel="home" href="index.html" title="JSON-GLib Reference Manual">
<link rel="up" href="index.html" title="JSON-GLib Reference Manual">
<link rel="prev" href="annotation-glossary.html" title="Annotation Glossary">
<meta name="generator" content="GTK-Doc V1.24 (XML mode)">
<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body bgcolor="white" text="black" link="#0000FF" alink="#840084" vlink="#0000FF">
<table class="navigation" id="top" width="100%" summary="Navigation header" cellpadding="2" cellspacing="5">
<td width="100%" align="left" class="shortcuts"></td>
<td><a accesskey="h" href="index.html"><img src="home.png" width="16" height="16" border="0" alt="Home"></a></td>
<td><img src="up-insensitive.png" width="16" height="16" border="0"></td>
<td><a accesskey="p" href="annotation-glossary.html"><img src="left.png" width="16" height="16" border="0" alt="Prev"></a></td>
<td><img src="right-insensitive.png" width="16" height="16" border="0"></td>
</tr></table>
<div class="appendix">
<div class="titlepage">
<div><div><h1 class="title">
AppendixA.License</h1></div></div>
<p>This library is free software; you can redistribute it and/or modify it under the terms of the <em class="citetitle">GNU Library General Public License</em> as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
</p>
<p>This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of
1.2837 require-main-filename 2.0.0 2.0.0

1.2837.1 Available under license:

Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.2838 cliui 4.1.0

1.2838.1 Available under license:

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2839 resolve-from 2.0.0 2.0.0

1.2839.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.2840 shelljs 0.8.4

1.2840.1 Available under license:

Copyright (c) 2012, Artur Adib <arturadib@gmail.com>
All rights reserved.

You may use this project under the terms of the New BSD license as follows:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Artur Adib nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ARTUR ADIB BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2841 big.js 5.2.2

1.2841.1 Available under license:

https://github.com/iriscouch/bigdecimal.js

BigDecimal for Javascript is licensed under the Apache License, version 2.0:

    Apache License
    Version 2.0, January 2004
    http://www.apache.org/licenses/

    TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

    1. Definitions.

    "License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
The MIT Licence (Expat).

Copyright (c) 2018 Michael Mclaughlin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 Daniel Trebbien and other contributors
Portions Copyright (c) 2003 STZ-IDA and PTV AG, Karlsruhe, Germany
Portions Copyright (c) 1995-2001 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.
ICU4J license - ICU4J 1.3.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2001 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

1.2842 serve-static 1.14.1

1.2842.1 Available under license:
(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2843 exit-hook 1.1.1

1.2843.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindresorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2844 preserve 0.2.0

1.2844.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2845 zookeeper 3.5.7

1.2845.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form,
including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark,
and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute
must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices
that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text
file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the
Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices
normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the
License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an
addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license
terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works
as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions
stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for
inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any
additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of
any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product
names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and
reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work
(and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR
PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and
assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or
otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing,
shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or
consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of
such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may
choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2846 worker-loader 1.1.1

1.2846.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2847 netty-tcnative 2.0.1.Final

1.2847.1 Available under license:

No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Bundle-Description: A Mavenized fork of Tomcat Native which incorporates various patches. This artifact is dynamically linked to OpenSSL and Apache APR.
Bundle-License: http://www.apache.org/licenses/LICENSE-2.0
Bundle-SymbolicName: io.netty.tcnative
Archiver-Version: Plexus Archiver
Built-By: norman
Bnd-LastModified: 1492498771597
Bundle-ManifestVersion: 2
Bundle-DocURL: http://netty.io/
Bundle-Vendor: The Netty Project
Tool: Bnd-2.1.0.20130426-122213
Export-Package: io.netty.internal.tcnative;version="2.0.1.Final"
Bundle-Name: Netty/TomcatNative [OpenSSL - Dynamic]
Bundle-Version: 2.0.1.Final
Bundle-NativeCode: META-INF/native/libnetty-tcnative.jnilib;osname=macos;osname=macosx;;processor=x86_64
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_121

Found in path(s):
*/
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources-jar/META-INF/MANIFEST.MF
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources.jar/io/netty/internal/tcnative/Buffer.java
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources.jar/io/netty/internal/tcnative/SSL.java
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources.jar/io/netty/internal/tcnative/SSLContext.java
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources.jar/io/netty/internal/tcnative/Library.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources.jar/io/netty/internal/tcnative/SessionTicketKey.java
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources.jar/io/netty/internal/tcnative/Library.java
/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources-
jar/io/netty/internal/tcnative/CertificateVerifier.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources-
jar/io/netty/internal/tcnative/NativeStaticallyReferencedJniMethods.java
* /opt/cola/permits/1202446267_1631568188.82/0/netty-tcnative-2-0-1-final-sources-
jar/io/netty/internal/tcnative/SniHostNameMatcher.java
1.2848 is-obj 2.0.0

1.2848.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2849 curvesapi 1.06

1.2850 picomatch 2.2.2

1.2850.1 Available under license:

The MIT License (MIT)

Copyright (c) 2017-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2851 jersey-media-multipart 2.25.1
1.2851.1 Available under license:
CDDL+GPL License
http://glassfish.java.net/public/CDDL+GPL_1_1.html

1.2852 regjsparser 0.1.5 0.1.5
1.2852.1 Available under license:
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2853 unicode-match-property-value-ecmascript 1.2.0
1.2853.1 Available under license:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2854 zlib 1.2.8

1.2854.1 Available under license :

/* zlib.h -- interface of the 'zlib' general purpose compression library
   version 1.2.11, January 15th, 2017

   Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

   This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

   Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

   1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
   2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
   3. This notice may not be removed or altered from any source distribution.

   Jean-loup Gailly        Mark Adler
   jloup@gzip.org          madler@alumni.caltech.edu

   */

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2855 ajv 6.12.6

1.2855.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2856 groovy-nio 3.0.8

1.2856.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements. See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership. The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an
 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 * KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1218316510_1634924928.45/0/groovy-nio-3-0-8-sources-jar/org/apache/groovy/nio/runtime/WritablePath.java
* /opt/cola/permits/1218316510_1634924928.45/0/groovy-nio-3-0-8-sources-jar/org/apache/groovy/nio/extensions/NioExtensions.java
* /opt/cola/permits/1218316510_1634924928.45/0/groovy-nio-3-0-8-sources-jar/org/codehaus/groovy/runtime/NioGroovyMethods.java
* /opt/cola/permits/1218316510_1634924928.45/0/groovy-nio-3-0-8-sources-jar/org/codehaus/groovy/runtime/WritablePath.java

1.2857 isexe 2.0.0

1.2857.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.2858 chokidar 3.3.1
1.2858.1 Available under license:
The MIT License (MIT)

Copyright (c) 2012-2019 Paul Miller (https://paulmillr.com), Elan Shanker

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the Software), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2859 archiver 3.0.0
1.2859.1 Available under license:
Copyright (c) 2012-2014 Chris Talkington, contributors.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2860 color-convert 0.5.3

1.2860.1 Available under license:

Copyright (c) 2011 Heather Arthur <fayearthur@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2861 swagger-converter 1.5.1 1.5.1

1.2861.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Apigee Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2862 vm2 3.8.3
1.2862.1 Available under license :
MIT License

Copyright (c) 2014-2019 Patrik Simek and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2863 google-http-client-xml 1.20.0
1.2863.1 Available under license :
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to
You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and
assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
#ifndef BOOST_INTRUSIVE_DETAIL_MPL_HPP
#define BOOST_INTRUSIVE_DETAIL_MPL_HPP

#ifndef BOOST_CONFIG_HPP
#define BOOST_CONFIG_HPP
#endif

#if defined(BOOST_HAS_PRAGMA_ONCE)
# pragma once
#endif

#include <boost/intrusive/detail/config_begin.hpp>
#include <cstddef>
namespace boost {
namespace intrusive {
namespace detail {

template <typename T, typename U>
struct is_same

namespace boost {
namespace intrusive {
namespace detail {

// (C) Copyright Ion Gaztanaga 2006-2014
// (C) Copyright Microsoft Corporation 2014
// Distributed under the Boost Software License, Version 1.0.
// (See accompanying file LICENSE_1_0.txt or copy at
//    http://www.boost.org/LICENSE_1_0.txt)

#include <boost/intrusive/detail/config_begin.hpp>
#include <cstddef>
namespace boost {
namespace intrusive {
namespace detail {

template <typename T, typename U>
struct is_same
{  
    static const bool value = false;
};

template<typename T>
struct is_same<T, T>
{
    static const bool value = true;
};

template<typename T>
struct add_const
{  
    typedef const T type;
};

template<typename T>
struct remove_const
{  
    typedef T type;
};

template<typename T>
struct remove_const<const T>
{  
    typedef T type;
};

template<typename T>
struct remove_cv
{  
    typedef T type;
};

template<typename T>
struct remove_cv<const T>
{  
    typedef T type;
};

template<typename T>
struct remove_cv<const volatile T>
{  
    typedef T type;
};

template<typename T>
struct remove_cv<volatile T>
{  
    typedef T type;
};

template<class T>
struct remove_reference
{  
    typedef T type;
};

template<class T>
struct remove_reference<T&>
{  
    typedef T type;
};
template<class T>
struct remove_pointer
{
    typedef T type;
};

template<class T>
struct remove_pointer<T*>
{
    typedef T type;
};

template<class T>
struct add_pointer
{
    typedef T *type;
};

typedef char one;
struct two {one _[2];};

template<bool C_>
struct bool_
{
    static const bool value = C_;
};

template<class Integer, Integer Value>
struct integer
{
    static const Integer value = Value;
};

typedef bool_<true>        true_;
typedef bool_<false>       false_;

typedef true_  true_type;
typedef false_ false_type;

typedef char yes_type;
struct no_type
{
    char padding[8];
};

template <bool B, class T = void>
struct enable_if_c {
    typedef T type;
};

template <class T>
struct enable_if_c<false, T> {};

template <class Cond, class T = void>
struct enable_if : public enable_if_c<Cond::value, T> {};

template<class F, class Param>
struct apply {
    typedef typename F::template apply<Param>::type type;
};

#if defined(_MSC_VER) && (_MSC_VER >= 1400)

template <class T, class U>
struct is_convertible {
    static const bool value = __is_convertible_to(T, U);
};
#else

template <class T, class U>
class is_convertible {
    typedef char true_t;
    class false_t { char dummy[2]; };
    //use any_conversion as first parameter since in MSVC
    //overaligned types can't go through ellipsis
    static false_t dispatch(...);
    static true_t  dispatch(U);
    static typename remove_reference<T>::type &trigger();
    public:
    static const bool value = sizeof(dispatch(trigger())) == sizeof(true_t);
};
#endif

template<bool C, typename T1, typename T2>
struct if_c
typedef T1 type;
};

template<
    typename T1
 , typename T2
>
struct if_c<false,T1,T2>
{
    typedef T2 type;
};

template<
    typename C
 , typename T1
, typename T2
>
struct if_
{
    typedef typename if_c<0 != C::value, T1, T2>::type type;
};

template<
    bool C
 , typename F1
, typename F2
>
struct eval_if_c
 : if_c<C,F1,F2>::type
{
};

template<
    typename C
 , typename T1
, typename T2
>
struct eval_if
 : if_<C,T1,T2>::type
{
};

// identity is an extension: it is not part of the standard.
template <class T>
struct identity
{
    typedef T type;
};
template<class T, bool Add>
struct add_const_if_c
{
    typedef typename if_c
        < Add
        , typename add_const<T>::type
        , T
        >::type type;
};

//boost::alignment_of yields to 10K lines of preprocessed code, so we
//need an alternative

template <typename T> struct alignment_of;


template <typename T>
struct alignment_of_hack
{
    char c;
    T t;
    alignment_of_hack();
};

template <unsigned A, unsigned S>
struct alignment_logic
{
    static const std::size_t value = A < S ? A : S;
};

template< typename T >
struct alignment_of
{
    static const std::size_t value = alignment_logic
        < sizeof(alignment_of_hack<T>) - sizeof(T)
        , sizeof(T)
        >::value;
};

template<class Class>
class is_empty_class
{
    template <typename T>
    struct empty_helper_t1 : public T
    {
        empty_helper_t1();
        int i[256];
    };
}
struct empty_helper_t2
{ int i[256]; }; public:
static const bool value = sizeof(empty_helper_t1<Class>) == sizeof(empty_helper_t2);
};

template<std::size_t S>
struct ls_zeros
{
    static const std::size_t value = (S & std::size_t(1)) ? 0 : (1 + ls_zeros<(S>>1u)::value);
};

template<>
struct ls_zeros<0>
{
    static const std::size_t value = 0;
};

template<>
struct ls_zeros<1>
{
    static const std::size_t value = 0;
};

template <typename T> struct unvoid_ref { typedef T &type; }

// Infrastructure for providing a default type for T::TNAME if absent.
#define BOOST_INTRUSIVE_INSTANTIATE_DEFAULT_TYPE_TMPLT(TNAME)     
    template <typename T, typename DefaultType>                    
    struct boost_intrusive_default_type_ ## TNAME                  
    {
        template <typename X>                                       
        static char test(int, typename X::TNAME*);                  
        template <typename X>                                       
        static int test(...);                                       
        struct DefaultWrap { typedef DefaultType TNAME; };          
        static const bool value = (1 == sizeof(test<T>(0, 0)));     
        typedef typename                                            
            ::boost::intrusive::detail::if_c                         
            <value, T, DefaultWrap>::TNAME type;            
    }
```c
//
#define BOOST_INTRUSIVE_OBTAIN_TYPE_WITH_DEFAULT(INSTANTIATION_NS_PREFIX, T, TNAME, TIMPL) \
    typename INSTANTIATION_NS_PREFIX \
    boost_intrusive_default_type_## TNAME< T, TIMPL >::type \
//
#define BOOST_INTRUSIVE_INSTANTIATE_EVAL_DEFAULT_TYPE_TMPLT(TNAME)\
    template <typename T, typename DefaultType>\
    struct boost_intrusive_eval_default_type_## TNAME \
    {\
        template <typename X>\
        static char test(int, typename X::TNAME*); \
        \
        template <typename X>\
        static int test(...); \
        \
        struct DefaultWrap \
        {\
            typedef typename DefaultType::type TNAME; \
        }; \
        \
        static const bool value = (1 == sizeof(test<T>(0, 0))); \
        \
        typedef typename \
        ::boost::intrusive::detail::eval_if_c \
        < value \
        , ::boost::intrusive::detail::identity<T> \
        , ::boost::intrusive::detail::identity<DefaultWrap> \
        >::type::TNAME type; \
    }; \
//
#define BOOST_INTRUSIVE_OBTAIN_TYPE_WITH_EVAL_DEFAULT(INSTANTIATION_NS_PREFIX, T, TNAME, TIMPL)\
    typename INSTANTIATION_NS_PREFIX \
    boost_intrusive_eval_default_type_## TNAME< T, TIMPL >::type \
//
#define BOOST_INTRUSIVE_INTERNAL_STATIC_BOOL_IS_TRUE(TRAITS_PREFIX, TYPEDEF_TO_FIND)\
    template <class T>\
    struct TRAITS_PREFIX##_bool \
    {\
        template<bool Add>\
        struct two_or_three {one _[2 + Add];}; \
        template <class U> static one test(...); \
        template <class U> static two_or_three<U::TYPEDEF_TO_FIND> test (int); \
        static const std::size_t value = sizeof(test<T>(0)); \
    };
```

---

Open Source Used In DNAC Platform Antelope 11126
template <class T>
struct TRAITS_PREFIX##_bool_is_true
{
    static const bool value = TRAITS_PREFIX##_bool<T>::value > sizeof(one)*2;
};

//
#define BOOST_INTRUSIVE_HAS_STATIC_MEMBER_FUNC_SIGNATURE(TRAITS_NAME, FUNC_NAME) \
    template <typename U, typename Signature> \
    class TRAITS_NAME \
    { \
    private: \
    template<Signature> struct helper; \
    template<typename T> \
    static ::boost::intrusive::detail::yes_type check(helper<&T::FUNC_NAME>*); \
    template<typename T> static ::boost::intrusive::detail::no_type check(...); \
    public: \
    static const bool value = sizeof(check<U>(0)) == sizeof(::boost::intrusive::detail::yes_type); \
    }; \

//
#define BOOST_INTRUSIVE_HAS_MEMBER_FUNC_CALLED(TRAITS_NAME, FUNC_NAME) \
    template <typename Type> \
    struct TRAITS_NAME \
    { \
    struct BaseMixin \
    { \
        void FUNC_NAME(); \
    }; \
    struct Base : public Type, public BaseMixin { Base(); }; \
    template <typename T, T t> class Helper{}; \
    template <typename U> \
    static ::boost::intrusive::detail::no_type  check(U*, Helper<void (BaseMixin::*)(), &U::FUNC_NAME>* = 0); \
    static ::boost::intrusive::detail::yes_type check(...); \
    static const bool value = sizeof(::boost::intrusive::detail::yes_type) == sizeof(check((Base*)(0))); \
    }; \

//
#define BOOST_INTRUSIVE_HAS_MEMBER_FUNC_CALLED_IGNORE_SIGNATURE(TRAITS_NAME, FUNC_NAME) \
BOOST_INTRUSIVE_HAS_MEMBER_FUNC_CALLED(TRAITS_NAME##_ignore_signature, FUNC_NAME) \

//
template <typename T>
inline T* addressof(T& obj)
{
    return static_cast<T*>(static_cast<void*>(
        reinterpret_cast<const char&>(obj)));
}

} //namespace detail
} //namespace intrusive
} //namespace boost

#include <boost/intrusive/detail/config_end.hpp>

<html>
All the value based traits in this library conform to MPL's requirements for
an Integral Constant type: that includes a number of rather intrusive workarounds
for broken compilers.

Purely as an implementation detail, this means that <code class="computeroutput">true_type</code>
inherits from <code class="computeroutput">boost::mpl::true_</code>,
<code class="computeroutput">false_type</code>
inherits from <code class="computeroutput">boost::mpl::false_</code>,
and <code class="computeroutput">integral_constant</code> (provided <code class="computeroutput">T</code> is not <code class="computeroutput">bool</code>)
inherits from <code class="computeroutput">boost::mpl::integral_c</code>.

Copyright © 2000, 2011 Adobe Systems Inc, David Abrahams,
Frederic Bron, Steve Cleary, Beman Dawes, Aleksey Gurtovoy, Howard Hinnant,
Jesse Jones, Mat Marcus, Itay Maman, John Maddock, Alexander Nasonov, Thorsten
Ottosen, Roman Perepelitsa, Robert Ramey, Jeremy Siek, Robert Stewart and Steven
Watanabe.

Distributed under the Boost Software License, Version 1.0. (See accompanying
file LICENSE_1_0.txt or copy at http://www.boost.org/LICENSE_1_0.txt)
#!/usr/bin/env python
from subprocess import check_output as run
from datetime import datetime
from itertools import groupby
from operator import itemgetter
import re
import magic
def authors(filename):
    log = run(['git', 'log', '--follow',
               '--date=short', '--format=%aN%x09%ad', filename],
               universal_newlines=True)
    for line in log.splitlines():
        author, date = line.split('	')
        if author != 'fix-copyright.py':
            yield author, datetime.strptime(date, '%Y-%m-%d')
def new_copyright(filename, previous):
    def f():
        au = list(authors(filename))
        alldates = map(itemgetter(1), au)
        aup = sorted(au + map(lambda a: (a, None), previous), key=itemgetter(0))
        for author, records in groupby(aup, itemgetter(0)):
            dates = filter(None, map(itemgetter(1), records))
            if not dates: dates = alldates
            start = min(dates)
            end = max(dates)
            fmt = '{0}' if start.year == end.year else '{0}-{1}'
            line = 'Copyright ' + fmt.format(start.year, end.year) + ' ' + author
            key = (start, author)
            yield key, line
        return map(itemgetter(1), sorted(f()))
def fix_copyright(filename):
    # Find copyright block in original file
    prefix = set()
names = []
lines = []
with open(filename, 'r') as f:
    content = list(f)
for i, line in enumerate(content[:15]):
    m = re.match(r'^\W*(\(<P?><prefix>\W*)(?c)\)?\s*?copyright\s*\((?c)\)?\s+s+\d\[4\]\((?c)\)?\s+s+\((?P<name>.+?)\)\s*\(('?P<year>\d+)?\)' line, re.IGNORECASE)
    if m:
        d = m.groupdict()
        prefix.add(d['prefix'])
        lines.append(i)
        names.append(d['name'].strip())
if len(prefix) != 1:
    print 'Not found:', filename
    return
prefix = list(prefix)[0]

print filename
new = iter(new_copyright(filename, names))
with open(filename, 'w') as f:
    for i, line in enumerate(content):
        if i in lines:
            for repl in new:
                print >>f, prefix + repl
        else:
            print >>f, line,
pass

def all_files():
    ls = run(['git', 'ls-files'], universal_newlines=True)
    for filename in ls.splitlines():
        if magic.from_file(filename, mime=True).split('/')[0] == 'text':
            yield filename
for f in all_files():
    fix_copyright(f)
[article Invalid copyright
[quickbook 1.5]
[copyright No year]
]

Maybe this should pass?
<?xml version="1.0" encoding="utf-8"?>

<!-- Copyright (c) 2001-2006 CrystalClear Software, Inc.
   Subject to the Boost Software License, Version 1.0.
   (See accompanying file LICENSE_1_0.txt or http://www.boost.org/LICENSE_1_0.txt)
-->

Open Source Used In DNAC Platform Antelope 11131
The following is the overall license for the boost date_time library. This notice is found in all source files related to the library.

Copyright © 2002 CrystalClear Software, Inc.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CrystalClear Software makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

© 2000-2002

Advertisements

#ifndef BOOST_CONTAINER_CONTAINER_DETAIL_MPL_HPP
#define BOOST_CONTAINER_CONTAINER_DETAIL_MPL_HPP
#ifndef BOOST_CONFIG_HPP
#include <boost/config.hpp>
#endif
#if defined(BOOST_HAS_PRAGMA_ONCE)
#pragma once
#endif
#include <boost/container/detail/config_begin.hpp>
#include <boost/container/detail/workaround.hpp>
#include <cstdlib>
namespace boost {
    namespace container {
        namespace container_detail {
            template <class T, T val>
            struct integral_constant
            {
                static const T value = val;
                typedef integral_constant<T,val> type;
            };

            template< bool C_ >
            struct bool_: integral_constant<bool, C_>
            {
                static const bool value = C_;    
                operator bool() const { return bool_::value; }    
            };

            template< unsigned V_ >
            struct unsigned_: integral_constant<unsigned, V_>
            {
                static const unsigned value = V_;   
                operator unsigned() const { return unsigned_::value; }   
            };

            typedef bool_<true> true_;
            typedef bool_<false> false_;

            typedef true_ true_type;
            typedef false_ false_type;

            typedef char yes_type;
            struct no_type
            {
            
        }
char padding[8];
;

template <bool B, class T = void>
struct enable_if_c {
    typedef T type;
};

template <class T>
struct enable_if_c<false, T> {};  

template <class Cond, class T = void>
struct enable_if : public enable_if_c<Cond::value, T> {};  

template <class Cond, class T = void>
struct disable_if : public enable_if_c<!Cond::value, T> {}; 

template <bool B, class T = void>
struct disable_if_c : public enable_if_c<!B, T> {};  

#if defined(_MSC_VER) && (_MSC_VER >= 1400)

template <class T, class U>
struct is_convertible
{
    static const bool value = __is_convertible_to(T, U);
};

#else

template <class T, class U>
class is_convertible
{
    typedef char true_t;
    class false_t { char dummy[2]; };  
    //use any_conversion as first parameter since in MSVC
    //overaligned types can't go through ellipsis
    static false_t dispatch(...);
    static true_t  dispatch(U);
    static T &trigger();
    public:
    static const bool value = sizeof(dispatch(trigger())) == sizeof(true_t);
};

#endif

template<
    bool C
}
struct if_<
    typename T1
    , typename T2
    >
struct if_<false,T1,T2>
{
    typedef T2 type;
};

template<
    typename T1
    , typename T2
    , typename T3
    >
struct if_<
{
    typedef typename if_<0 != T1::value, T2, T3>::type type;
};

template <class Pair>
struct select1st
{
    typedef Pair argument_type;
    typedef typename Pair::first_type result_type;

template<class OtherPair>
const typename Pair::first_type& operator()(const OtherPair& x) const
{  return x.first;  }

const typename Pair::first_type& operator()(const typename Pair::first_type& x) const
{  return x;  }
};

// identity is an extension: it is not part of the standard.
template <class T>
struct identity
{
    typedef T argument_type;
    typedef T result_type;

typedef T type;
const T& operator()(const T& x) const
{
    return x;
};

template<std::size_t S>
struct ls_zeros
{
    static const std::size_t value = (S & std::size_t(1)) ? 0 : (1u + ls_zeros<(S >> 1u)>::value);
};

template<>
struct ls_zeros<0>
{
    static const std::size_t value = 0;
};

template<>
struct ls_zeros<1>
{
    static const std::size_t value = 0;
};

template <std::size_t OrigSize, std::size_t RoundTo>
struct ct_rounded_size
{
    static const std::size_t value = ((OrigSize-1)/RoundTo+1)*RoundTo;
};

template <typename T> struct unvoid { typedef T type; };
template <> struct unvoid<void> { struct type { }; };
template <> struct unvoid<const void> { struct type { }; };

} //namespace container_detail {
} //namespace container {
} //namespace boost {

#include <boost/container/detail/config_end.hpp>
#endif   //#ifndef BOOST_CONTAINER_CONTAINER_DETAIL_MPL_HPP

This is an attempt to acknowledge early contributions to the garbage collector. Later contributions should instead be mentioned in README.changes.

HISTORY -

Early versions of this collector were developed as a part of research
The garbage collector originated as part of the run-time system for the Russell programming language implementation. The first version of the garbage collector was written primarily by Al Demers. It was then refined and mostly rewritten, primarily by Hans-J. Boehm, at Cornell U., the University of Washington, Rice University (where it was first used for C and assembly code), Xerox PARC, SGI, and HP Labs. However, significant contributions have also been made by many others.

Some other contributors:

More recent contributors are mentioned in the modification history in README.changes. My apologies for any omissions.

The SPARC specific code was originally contributed by Mark Weiser. The Encore Multimax modifications were supplied by Kevin Kenny (kenny@m.cs.uiuc.edu). The adaptation to the IBM PC/RT is largely due to Vernon Lee, on machines made available to Rice by IBM. Much of the HP specific code and a number of good suggestions for improving the generic code are due to Walter Underwood. Robert Bazzie (brazie@diamond.bbn.com) originally supplied the ULTRIX code. Al Dosser (dosser@src.dec.com) and Regis Cridlig (Regis.Cridlig@cl.cam.ac.uk) subsequently provided updates and information on variation between ULTRIX systems. Parag Patel (parag@netcom.com) supplied the A/UX code. Jesper Peterson(jep@mtiame.mtia.oz.au), Michel Schinz, and Martin Tauchmann (martintauchmann@bigfoot.com) supplied the Amiga port. Thomas Funke (thf@zelator.in-berlin.de(?)) and Brian D.Carlstrom (bdc@clark.lcs.mit.edu) supplied the NeXT ports. Douglas Steel (doug@wg.icl.co.uk) provided ICL DRS6000 code. Bill Janssen (janssen@parc.xerox.com) supplied the SunOS dynamic loader specific code. Manuel Serrano (serrano@cornas.inria.fr) supplied linux and Sony News specific code. Al Dosser provided Alpha/OSF/1 code. He and Dave Detlefs(detlefs@src.dec.com) also provided several generic bug fixes. Alistair G. Crooks(ace@uts.amdahl.com) supplied the NetBSD and 386BSD ports. Jeffrey Hsu (hsu@soda.berkeley.edu) provided the FreeBSD port. Brent Benson (brent@jade.ssd.csd.harris.com) ported the collector to a Motorola 88K processor running CX/UX (Harris NightHawk). Ari Huttunen (Ari.Huttunen@hut.fi) generalized the OS/2 port to nonIBM development environments (a nontrivial task). Patrick Beard (beard@cs.ucdavis.edu) provided the initial MacOS port. David Chase, then at Olivetti Research, suggested several improvements. Scott Schwartz (schwartz@groucho.cse.psu.edu) supplied some of the code to save and print call stacks for leak detection on a SPARC. Jesse Hull and John Ellis supplied the C++ interface code. Zhong Shao performed much of the experimentation that led to the current typed allocation facility. (His dynamic type inference code hasn't
made it into the released version of the collector, yet.)

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2002-2003, Trustees of Indiana University.
Copyright 2000-2001, University of Notre Dame.
All rights reserved.

Indiana University has the exclusive rights to license this product under the following license.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* All redistributions of source code must retain the above copyright notice, the list of authors in the original source code, this list of conditions and the disclaimer listed in this license;

* All redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer listed in this license in the documentation and/or other materials provided with the distribution;

* Any documentation included with all redistributions must include the following acknowledgement:

"This product includes software developed at the University of Notre Dame
and the Pervasive Technology Labs at Indiana University. For technical information contact Andrew Lumsdaine at the Pervasive Technology Labs at Indiana University. For administrative and license questions contact the Advanced Research and Technology Institute at 351 West 10th Street, Indianapolis, Indiana 46202, phone 317-278-4100, fax 317-274-5902."

Alternatively, this acknowledgement may appear in the software itself, and wherever such third-party acknowledgments normally appear.

* The name Indiana University, the University of Notre Dame or "Caramel" shall not be used to endorse or promote products derived from this software without prior written permission from Indiana University. For written permission, please contact Indiana University Advanced Research & Technology Institute.

* Products derived from this software may not be called "Caramel", nor may Indiana University, the University of Notre Dame or "Caramel" appear in their name, without prior written permission of Indiana University Advanced Research & Technology Institute.

Indiana University provides no reassurances that the source code provided does not infringe the patent or any other intellectual property rights of any other entity. Indiana University disclaims any liability to any recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise.

LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR OTHER PROPRIETARY RIGHTS. INDIANA UNIVERSITY MAKES NO WARRANTIES THAT SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP DOORS", "WORMS", OR OTHER HARMFUL CODE. LICENSEE ASSUMES THE ENTIRE RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS, AND TO THE PERFORMANCE AND VALIDITY OF INFORMATION GENERATED USING SOFTWARE. The following people hereby grant permission to replace all existing licenses on their contributions to Boost with the Boost Software License, Version 1.0. (boostinspect:nolicense boostinspect:nocopyright)

Aleksey Gurtovoy (agurtovoy@meta-comm.com)
Andrei Alexandrescu (andrewalex - at - hotmail.com) (See Boost list message of August 12, 2004 11:06:58 AM EST)
Andrew Lumsdaine ()
Anthony Williams (anthony -at- justsofaresolutions.co.uk)
Beman Dawes (bdawes@acm.org)
Brad King (brad.king -at- kitware.com) (See Boost list message of Wed, 21 Jul 2004 11:15:46 -0400)
Brian Osman (osman -at- vvisions.com) (See CVS log)
Bruce Barr (schmoost -at- yahoo.com) (See Boost list of Mon, 16 Aug 2004 15:06:43 -0500)
Bruno da Silva de Oliveira (bruno - at - esss.com.br)

Open Source Used In DNAC Platform Antelope  11139
Christain Engstrom (christian.engstrom -at- glindra.org) (See Boost list message of Mon, 30 Aug 2004 14:31:49 +0200)
Cromwell D Enage (sponage -at- yahoo.com) (See Boost list message of August 12, 2004 11:49:13 AM EST)
Dan Gohman (dig -at- cray.com) (See Boost list message of Sat, 21 Aug 2004 10:54:59 +0100)
Dan Nuffer (dan -at- nuffer.name)
Daniel Frey (d.frey -at- gmx.de, daniel.frey -at- aixigo.de)
Daniel Nuffer (dan -at- nuffer.name)
Darin Adler (darin -at- bentspoon.com) (Email to Andreas Huber, see change log)
Daryle Walker (darylew - at - hotmail.com)
Dave Abrahams (dave@boost-consulting.com)
Dave Moore (dmoore -at- viefinancial.com) (See Boost list message of 18 Dec 2003 15:35:50 -0500)
David Abrahams (dave@boost-consulting.com)
Dietmar Kuehl (dietmar_kuehl -at- yahoo.com) (Email to Andreas Huber, see change log)
Douglas Gregor (gregod -at- cs.rpi.edu, dgregor -at- cs.indiana.edu, doug.gregor -at- gmail.com)
Dr John Maddock (john - at - johnmaddock.co.uk)
Edward D. Brey (bre -at- ductape.net) (Email to Andreas Huber, see change log)
Eric Ford (un506n902 -at- sneakemail.com) (See Boost list message of Sun, 15 Aug 2004 10:29:13 +0100)
Eric Friedman (efb@users.sourceforge.net)
Eric Niebler (eric@boost-consulting.com)
Fernando Cacciola (fernando_cacciola@ciudad.com.ar)
Fernando Luis Cacciola Carballal (fernando_cacciola@ciudad.com.ar)
Francois Faure (Francois.Faure -at- imag.fr) (See CVS log)
Gary Powell (powellg - at - amazon.com) (See Boost list message of 10 Feb 2004 14:22:46 -0800)
Genndady Rozental (rogeeff -at- mail.com) (Email to Andreas Huber, see change log)
Gottfried Ganssauge (Gottfried.Ganssauge -at- HAUF.E.DE) (See Boost list message of Mon, 16 Aug 2004 10:09:19 +0200)
Gottfried Ganaue (Gottfried.Ganssauge -at- HAUF.E.DE) (Alternative spelling of Gottfried Ganssauge)
Greg Colvin (gregory.colvin -at- oracle.com) (See Boost list message of Sat, 14 Aug 2004 10:57:00 +0100)
Gregory Colvin (gregory.colvin -at- oracle.com) (See Boost list message of Sat, 14 Aug 2004 10:57:00 +0100)
Gunter Winkler (gunter.winkler -at- unibw-muenchen.de) (See Boost list message of Mon, 16 Aug 2004 10:24:17 +0200)
Hartmut Kaiser (hartmut.kaiser -at- gmail.com)
Herve Bronnimann (hbr -at- poly.edu)
Herv Brnnimann (hbr -at- poly.edu)
Housemarque Oy (Ilari Kuittinen ilari.kuittinen -at- housemarque.fi)
Howard Hinnant (hinnant -at- twcny.rr.com) (See Boost list message of July 25, 2004 3:44:49 PM EST)
Hubert Holin (hubert_holin -at- users.sourceforge.net)
Indiana University ()
Itay Maman (imaman -at- users.sourceforge.net)
Jaakko Jrv (jaarvi -at- osl.iu.edu)
Jaap Suter (j.suter -at- student.utwente.nl) (See Boost list message of Thu, 16 Sep 2004 09:32:43 -0700)
Jeff Garland (jeff - at - crystalcalesharesoftware.com) (see Boost list post of July 25, 2004 19:31:09 -0700)
Jens Maurer (Jens.Maurer@gmx.net)
Jeremy G Siek (jsiek@osl.iu.edu)
Jeremy Siek (jsiek@osl.iu.edu)
Joel de Guzman (joel -at- boost-consulting.com) (See Boost list message of July 25, 2004 8:32:00 PM EST)
John Bandela (jbandela-at-ufl.edu)
John Maddock (john - at - johnmaddock.co.uk)
Open Source Used In DNAC Platform Antelope 11141

John R Bandela (jbandela-at-ufl.edu)
Jonathan Turkanis (turkanis -at- coderage dot com)
Juergen Hunold (hunold -at- ive.uni-hannover.de) (See Boost List Message of Fri, 13 Aug 2004 19:39:55 +0200)
Kevin Henney (kevin -at- curbralan.com) (See Boost list message of Wed, 15 Sep 2004 18:15:17 +0200)
Kresimir Fresl (fresl -at- master.grad.hr) (See Boost List message of August 16, 2004 8:23:35 AM EST)
Lars Guillik Bjannes (larsbj -at- lyx.org) (See Boost list message of Tue, 17 Aug 2004 15:49:02 +0100)
Lie-Quan Lee (liequan - at - slac.stanford.edu, llee - at - cs.indiana.edu)
Maarten Keijzer (mkeijzer -at- cs.vu.nl) (See Boost list message of Wed, 18 Aug 2004 21:43:18 +0100)
Mac Murrett (mmurrett -at- mac.com)
Marc Wintermantel (wintermantel -at- imes.mavt.ethz.ch, wintermantel -at- even-ag.ch) (See CVS log)
Michael Glassford (glassfordm - at - hotmail.com)
Michael Stevens (Michael.Stevens - at - epost.de)
Multi Media Ltd. (pdimov@mmltd.net)
Nicolai M Josuttis (solutions -at- josuttis.com) (See Boost list message of Mon, 30 Aug 2004 10:52:00 +0100)
Nikolay Mladenov (nickm -at- sitius.com) (See Boost list message of Tue, 17 Aug 2004 15:45:33 +0100)
Paul Mensonides (pmenso57 -at- comcast.net) (See Boost list message of July 21, 2004 1:12:21 AM EST)
Pavol Droba (droba -at- topmail.sk)
Peter Dimov (pdimov@mmltd.net)
R W Grosse-Kunstleve (RWGrosse-Kunstleve@lbl.gov)
Ralf W. Grosse-Kunstleve (RWGrosse-Kunstleve@lbl.gov)
Rational Discovery LLC (Greg Landrum Landrum -at- RationalDiscovery.com) (See Boost list post of Tue, 17 Aug 2004 10:35:36 +0100)
Rene Rivera (grafik/redshift-software.com, rrvera@acm.org)
Robert Ramey (ramey@www.rrsd.com)
Roland Richter (roland -at- fllj.kju.at) (See Boost list post of Mon, 16 Aug 2004 22:16:55 +0200)
Roland Schwarz (roland.schwarz -at- chello.at)
Ronald Garcia (garcia -at- cs.indiana.edu) (Email to Andreas Huber, see change log)
Samuel Krempp (kremp -at- trans.ens-cachan.fr) (See Boost list message of Mon, 27 Sep 2004 13:18:36 +0200)
Stefan Seefeld (seefeld -at- sympatico.ca)
Stephen Cleary (scleary -at- jerviswebb.com) (See Boost list message of Tue, 28 Sep 2004 13:11:46 +0100)
Steve Cleary (Variant of Stephen Cleary)
Sylvain Pion (Sylvain.Pion - at - sophia.inria.fr)
The Trustees of Indiana University ()
Thomas Witt (witt - at - ive.uni-hannover.de, witt - at - acm.org, witt - at - styleadvisor.com)
Thorsten Jrgen Ottosen (nesotto - at - cs.auc.dk)
Thorsten Ottosen (nesotto - at - cs.auc.dk)
Toon Knapen (toon dot knapen - at - fft.be)
The Trustees of Indiana University ()
University of Notre Dame ()
Vladimir Prus (ghost@cs.msu.su)
William E. Kempf () (email to Beman Dawes, 9/14/2006 4:18 PM)
Joerg Walter (jhr.walter - at - t-online.de : email to ublas mailing list Mon, 17 Sep 2007 10:17:08 +0200)
Mathias Koch (mkoch - at - idesis.de 7 : email to boost-owner@lists.boost.org Sep 2007 13:20:09 +0200)

--- end ---

Use of this software is granted under one of the following two licenses,
to be chosen freely by the user.

Copyright (c) 2006, 2007 Marcin Kalicinski

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

2. The MIT License

Copyright (c) 2006, 2007 Marcin Kalicinski

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
===========================================================================
Copyright (c) 2007-2011 Barend Gehrels, Amsterdam, the Netherlands.
Copyright (c) 2008-2011 Bruno Lalande, Paris, France.
Copyright (c) 2009-2011 Mateusz Loskot, London, UK.

Use, modification and distribution is subject to the Boost Software License,
Version 1.0. (See accompanying file LICENSE_1_0.txt or copy at
http://www.boost.org/LICENSE_1_0.txt)
===========================================================================

The default copyright note for C++ source files reads:

// Boost.Geometry (aka GGL, Generic Geometry Library)

// Copyright (c) 2007-2011 Barend Gehrels, Amsterdam, the Netherlands.
// Copyright (c) 2008-2011 Bruno Lalande, Paris, France.
// Copyright (c) 2009-2011 Mateusz Loskot, London, UK.

// Parts of Boost.Geometry are redesigned from Geodan's Geographic Library
// (geolib/GGL), copyright (c) 1995-2010 Geodan, Amsterdam, the Netherlands.

// Use, modification and distribution is subject to the Boost Software License,
// Version 1.0. (See accompanying file LICENSE_1_0.txt or copy at
// http://www.boost.org/LICENSE_1_0.txt)

Exceptions:
1) Major work of any author -> might change order, change date
2) Exclusive work of one author including design -> might take sole copyright
Examples:
- adapted geometry types e.g. array, Boost.Polygon
- implemented file-format e.g. WKB
- implemented specific strategy or algorithm e.g. intersections
3) Examples -> might take sole copyright
4) Tests -> might take sole copyright
5) Docs -> might take sole copyright
6) Utilities e.g. converters -> might take sole copyright

The copyright note for other (source) files as .py, Jamfiles, etc is similar
and comments are changed accordingly.
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE article PUBLIC "-/Boost//DTD BoostBook XML V1.0//EN"
"http://www.boost.org/tools/boostbook/dtd/boostbook.dtd">
<article id="copyright_test" last-revision="DEBUG MODE Date: 2000/12/20 12:00:00 $"
xmlns:xi="http://www.w3.org/2001/XInclude">
<title>Copyright Test</title>

Open Source Used In DNAC Platform Antelope 11143
This software is a collection of libraries from the Boost.org site.
Most of the libraries use the Boost Software License 1.0, which reads as follows.

Boost Software License - Version 1.0
------------------------------------

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

There are a few files under a license different from the Boost Software License; however, all licenses do follow these guidelines:

License requirements

- Must be simple to read and understand.
- Must grant permission to copy, use and modify the software for any use (commercial and non-commercial) for no fee.
- Must require that the license appear on all copies of the software source code.
- Must not require that the license appear with executables or other binary uses of the library.
- Must not require that the source code be available for execution or other binary uses of the library.
- May restrict the use of the name and description of the library to the standard version found on the Boost web site.

Files: tools/jam
License:
/*
 */+
/* +   Copyright 1993-2002 Christopher Seiwald and Perforce Software, Inc.
 */+/
/*
/* This file is part of jam.
/*
/* License is hereby granted to use this software and distribute it freely, as long as this copyright notice is retained and modifications are clearly marked.
/*
/* ALL WARRANTIES ARE HEREBY DISCLAIMED.
*/
/* This file is ALSO:
* Copyright 2001-2004 David Abrahams.
* Distributed under the Boost Software License, Version 1.0.
* (See accompanying file LICENSE_1_0.txt or http://www.boost.org/LICENSE_1_0.txt)
*/

* Copyright 1988, 1989 Hans-J. Boehm, Alan J. Demers
* Copyright (c) 1991-1994 by Xerox Corporation. All rights reserved.
* Copyright (c) 1998-1999 by Silicon Graphics. All rights reserved.
* Copyright (c) 1999 by Hewlett-Packard Company. All rights reserved.
*
* THIS MATERIAL IS PROVIDED AS IS, WITH ABSOLUTELY NO WARRANTY EXPRESSED
* OR IMPLIED. ANY USE IS AT YOUR OWN RISK.
*
* Permission is hereby granted to use or copy this program
* for any purpose, provided the above notices are retained on all copies.
* Permission to modify the code and to distribute modified code is granted,
* provided the above notices are retained, and a notice that the code was
* modified is included with the above copyright notice.
*/

Files: boostbook/xsl/caramel

License:
Software License, Version 1.0

Copyright 2002-2003, Trustees of Indiana University.
Copyright 2000-2001, University of Notre Dame.
All rights reserved.

Indiana University has the exclusive rights to license this product under the following license.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* All redistributions of source code must retain the above copyright notice, the list of authors in the original source code, this list of conditions and the disclaimer listed in this license;

* All redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer listed in this license in the documentation and/or other materials provided with the distribution;

* Any documentation included with all redistributions must include the following acknowledgement:

"This product includes software developed at the University of Notre Dame and the Pervasive Technology Labs at Indiana University. For technical information contact Andrew Lumsdaine at the Pervasive Technology Labs at
Indiana University. For administrative and license questions contact the Advanced Research and Technology Institute at 351 West 10th Street, Indianapolis, Indiana 46202, phone 317-278-4100, fax 317-274-5902.”

Alternatively, this acknowledgement may appear in the software itself, and wherever such third-party acknowledgments normally appear.

* The name Indiana University, the University of Notre Dame or "Caramel" shall not be used to endorse or promote products derived from this software without prior written permission from Indiana University. For written permission, please contact Indiana University Advanced Research & Technology Institute.

* Products derived from this software may not be called "Caramel", nor may Indiana University, the University of Notre Dame or "Caramel" appear in their name, without prior written permission of Indiana University Advanced Research & Technology Institute.

Indiana University provides no reassurances that the source code provided does not infringe the patent or any other intellectual property rights of any other entity. Indiana University disclaims any liability to any recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise.

LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR OTHER PROPRIETARY RIGHTS. INDIANA UNIVERSITY MAKES NO WARRANTIES THAT SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP DOORS", "WORMS", OR OTHER HARMFUL CODE. LICENSEE ASSUMES THE ENTIRE RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS, AND TO THE PERFORMANCE AND VALIDITY OF INFORMATION GENERATED USING SOFTWARE.

This package was debianized by Vladimir Prus <ghost@cs.msu.su> on Wed, 17 July 2002, 19:27:00 +0400.

Copyright:

/\  
+\Copyright 1993-2002 Christopher Seiwald and Perforce Software, Inc.  
+\/

This is Release 2.4 of Jam/MR, a make-like program.

License is hereby granted to use this software and distribute it freely, as long as this copyright notice is retained and modifications are clearly marked.

ALL WARRANTIES ARE HEREBY DISCLAIMED.
Some portions are also:

Copyright 2001-2006 David Abrahams.
Copyright 2002-2006 Rene Rivera.
Copyright 2003-2006 Vladimir Prus.

Distributed under the Boost Software License, Version 1.0.
(See accompanying file LICENSE_1_0.txt or http://www.boost.org/LICENSE_1_0.txt)
[article Copyright Test
[quickbook 1.5]
[copyright 1977,1985 Someone else]
]

/////////////////////////////////////////////////////////////////////////
// (C) Copyright Ion Gaztanaga 2005-2012.
//
// Distributed under the Boost Software License, Version 1.0.
// (See accompanying file LICENSE_1_0.txt or copy at
// http://www.boost.org/LICENSE_1_0.txt)
//
//
/////////////////////////////////////////////////////////////////////////

#ifndef BOOST_INTERPROCESS_DETAIL_MPL_HPP
#define BOOST_INTERPROCESS_DETAIL_MPL_HPP

#ifndef BOOST_CONFIG_HPP
#  include <boost/config.hpp>
#endif

#if defined(BOOST_HAS_PRAGMA_ONCE)
#  pragma once
#endif

#include <cstdlib>

namespace boost {
namespace interprocess {
namespace ipcdetail {

template <class T, T val>
struct integral_constant {
  static const T value = val;
  typedef integral_constant<T,val> type;

}}
}
}}

#include <cstdlib>

namespace boost {
namespace interprocess {
namespace ipcdetail {

template <class T, T val>
struct integral_constant {
  static const T value = val;
  typedef integral_constant<T,val> type;

}}
}
}}

open source used in DNAC Platform Antelope 11148
template< bool C_ >
struct bool_ : integral_constant<bool, C_>
{
    static const bool value = C_;
};

typedef bool_<true> true_;
typedef bool_<false> false_;
typedef true_ true_type;
typedef false_ false_type;
typedef char yes_type;
struct no_type
{
    char padding[8];
};
template <bool B, class T = void>
struct enable_if_c {
    typedef T type;
};
template <class T>
struct enable_if_c<false, T> {};
template <class Cond, class T = void>
struct enable_if : public enable_if_c<Cond::value, T> {};
template <class Cond, class T = void>
struct disable_if : public enable_if_c<!Cond::value, T> {};
template <class T, class U>
class is_convertible
{
    typedef char true_t;
    class false_t { char dummy[2]; };  
    static true_t dispatch(U);
    static false_t dispatch(...);
    static T trigger();  
    public:
    static const bool value = sizeof(dispatch(trigger())) == sizeof(true_t);
};
template<
    bool C
template<
    typename T1
    , typename T2
>
struct if_c
{
    typedef T1 type;
};

template<
    typename T1
    , typename T2
>
struct if_c<false,T1,T2>
{
    typedef T2 type;
};

template<
    typename T1
    , typename T2
    , typename T3
>
struct if_
{
    typedef typename if_c<0 != T1::value, T2, T3>::type type;
};

template <class Pair>
struct select1st
//   : public std::unary_function<Pair, typename Pair::first_type>
{
    template<class OtherPair>
    const typename Pair::first_type& operator()(const OtherPair& x) const
    {  return x.first;   }

    const typename Pair::first_type& operator()(const typename Pair::first_type& x) const
    {  return x;   }
};

// identity is an extension: it is not part of the standard.
template <class T>
struct identity
//   : public std::unary_function<T,T>
{
    typedef T type;
    const T& operator()(const T& x) const
    { return x;  }
template<std::size_t S>
struct ls_zeros
{
    static const std::size_t value = (S & std::size_t(1)) ? 0 : (1u + ls_zeros<(S >> 1u)>::value);
};

template<>
struct ls_zeros<0>
{
    static const std::size_t value = 0;
};

template<>
struct ls_zeros<1>
{
    static const std::size_t value = 0;
};

} //namespace ipcdetail {
} //namespace interprocess {
} //namespace boost {

#endif   //#ifndef BOOST_INTERPROCESS_DETAIL_MPL_HPP

[section:mpl MPL Interoperability]

All the value based traits in this library conform to MPL's requirements
for an [../../../../libs/mpl/doc/refmanual/integral-constant.html Integral Constant type]: that includes a number of
rather intrusive
workarounds for broken compilers.

Purely as an implementation detail, this
means that `__true_type` inherits from [../../../../libs/mpl/doc/refmanual/bool.html boost::mpl::true_],
`__false_type` inherits
from [../../../../libs/mpl/doc/refmanual/bool.html boost::mpl::false_], and `__integral_constant<T, v>` inherits from
[../../../../libs/mpl/doc/refmanual/integral-c.html boost::mpl::integral_c<T,v>`) (provided `T` is not `bool`)

[endsect]
Open Source Used In DNAC Platform Antelope 11152
split_path& operator=(const split_path&);
};

std::ostream& operator << (std::ostream& os, const split_path& p)
{
    os << "<a href="" << (p.root / p.file).string() << ">
" << p.file.string() << "</a>";
    return os;
}

std::string make_link_target(const std::string& s)
{
    // convert an arbitrary string into something suitable
    // for an <a> name:
    std::string result;
    for(unsigned i = 0; i < s.size(); ++i)
    {
        result.append(1, static_cast<std::string::value_type>(std::isalnum(s[i]) ? s[i] : '_'));
    }
    return result;
}

void bcp_implementation::output_license_info()
{
    std::pair<const license_info*, int> licenses = get_licenses();

    std::map<int, license_data>::const_iterator i, j;
    i = m_license_data.begin();
    j = m_license_data.end();

    std::ofstream os(m_dest_path.string().c_str());
    if(!os)
    {
        std::string msg("Error opening ");
        msg += m_dest_path.string();
        msg += " for output.");
        std::runtime_error e(msg);
        boost::throw_exception(e);
    }
    os << "<!DOCTYPE HTML PUBLIC ">
" << W3C/ DTD HTML 4.0 Transitional/ EN" >> "\n" << "<html>
" << "<head>
" << "<title>Boost Licence Dependency Information"; 
    if(m_module_list.size() == 1)
    {
        os << " for " << *(m_module_list.begin());
    }
os <<
"</title>\n"
"</head>\n"
"<body>\n"
"<H1>Boost Licence Dependency Information</H1>
if(m_module_list.size() == 1)
{
    os << " for " << *(m_module_list.begin());
}

os <<
"</H1>\n"
"<H2>Contents</h2>
"<pre><a href="#input">Input Information</a>
if(!m_bsl_summary_mode)
    os << " <a href="#summary">Licence Summary</a> <a href="#details">Licence Details</a> <a href="#files">Files with no recognised license</a> <a href="#authors">Files with no recognised copyright holder</a> <a href="#depend">File Dependency Information</a>
if(!m_bsl_summary_mode)
{
    os <<
"Moving to the Boost Software License...\n"
    " <a href="#bsl-converted">Files that can be automatically converted to the Boost Software License</a> <a href="#to-bsl">Files that can be manually converted to the Boost Software License</a> <a href="#not-to-bsl">Files that can <b>NOT</b> be moved to the Boost Software License</a> <a href="#need-bsl-authors">Authors we need to move to the Boost Software License</a> <a href="#copyright">Copyright Holder Information</a>
}

os <<
"Files with no recognised license</a> <a href="#authors">Files with no recognised copyright holder</a>;
if(!m_bsl_summary_mode)
{
    os <<
"Moving to the Boost Software License...\n"
    " <a href="#bsl-converted">Files that can be automatically converted to the Boost Software License</a> <a href="#to-bsl">Files that can be manually converted to the Boost Software License</a> <a href="#not-to-bsl">Files that can <b>NOT</b> be moved to the Boost Software License</a> <a href="#need-bsl-authors">Authors we need to move to the Boost Software License</a> <a href="#copyright">Copyright Holder Information</a>
}

os <<
"Files with no recognised license</a> <a href="#authors">Files with no recognised copyright holder</a>;
if(!m_bsl_summary_mode)
{
    os <<
"Moving to the Boost Software License...\n"
    " <a href="#bsl-converted">Files that can be automatically converted to the Boost Software License</a> <a href="#to-bsl">Files that can be manually converted to the Boost Software License</a> <a href="#not-to-bsl">Files that can <b>NOT</b> be moved to the Boost Software License</a> <a href="#need-bsl-authors">Authors we need to move to the Boost Software License</a> <a href="#copyright">Copyright Holder Information</a>
}

os <<
"Files with no recognised license</a> <a href="#authors">Files with no recognised copyright holder</a>;
if(!m_bsl_summary_mode)
{
    os <<
"Moving to the Boost Software License...\n"
    " <a href="#bsl-converted">Files that can be automatically converted to the Boost Software License</a> <a href="#to-bsl">Files that can be manually converted to the Boost Software License</a> <a href="#not-to-bsl">Files that can <b>NOT</b> be moved to the Boost Software License</a> <a href="#need-bsl-authors">Authors we need to move to the Boost Software License</a> <a href="#copyright">Copyright Holder Information</a>
}

os <<
"Files with no recognised license</a> <a href="#authors">Files with no recognised copyright holder</a>;
if(!m_bsl_summary_mode)
{
    os <<
"Moving to the Boost Software License...\n"
    " <a href="#bsl-converted">Files that can be automatically converted to the Boost Software License</a> <a href="#to-bsl">Files that can be manually converted to the Boost Software License</a> <a href="#not-to-bsl">Files that can <b>NOT</b> be moved to the Boost Software License</a> <a href="#need-bsl-authors">Authors we need to move to the Boost Software License</a> <a href="#copyright">Copyright Holder Information</a>

The following Boost modules were checked:

```cpp
std::list<std::string>::const_iterator si = m_module_list.begin();
std::list<std::string>::const_iterator sj = m_module_list.end();
while(si != sj)
{
    os << *si << "<BR>";
    ++si;
}
```

The Boost path was:

```cpp
os << ":<BR>";
```

// extract the boost version number from the boost directory tree,
// not from this app (which may have been built from a previous
// version):

```cpp
fileview version_file(m_boost_path / "boost/version.hpp");
static const boost::regex version_regex("^[[:blank:]]*#^[[:blank:]]*define[[:blank:]]*BOOST_VERSION[[:blank:]]*+(\d+)+);
boost::cmatch what;
if(boost::regex_search(version_file.begin(), version_file.end(), what, version_regex))
{
    int version = boost::lexical_cast<int>(what.str(1));
    os << "The Boost version is: " << version / 100000 << "." << version / 100 % 100 << "." << version % 100 << "\n";
}
```

// output each license:

```cpp
i = m_license_data.begin();
j = m_license_data.end();
if(!m_bsl_summary_mode)
{
    // start with the summary:

    os << "<a name="summary">\n";
    while(i != j)
    {
        // title:
        os << "<H3>" << licenses.first[i->first].license_name << "</H3>\n";
        // license text:
        os << "<BLOCKQUOTE>" << licenses.first[i->first].license_text << "</BLOCKQUOTE>";
        // Copyright holders:
        os << "This license is used by " << i->second.authors.size() << " authors and " << i->second.files.size() << " files <a href="\">(see details)</a>";
```
// and now the details:
//
i = m_license_data.begin();
j = m_license_data.end();
int license_index = 0;
os << "<a name="details"></a><h2>Licence Details</h2>
while(i != j)
{
    // title:
os << "<H3><A name="" << make_link_target(licenses.first[i->first].license_name) ""></a>" << licenses.first[i->first].license_name << "</H3>";
    // license text:
os << "<BLOCKQUOTE>" << licenses.first[i->first].license_text << "</BLOCKQUOTE>";
    if(!m_bsl_summary_mode || (license_index >= 3))
    {
        // Copyright holders:
os << "<P>This license is used by the following " << i->second.authors.size() << " copyright holders:";
std::set<std::string>::const_iterator x, y;
x = i->second.authors.begin();
y = i->second.authors.end();
while(x != y)
{
os << *x << "<BR>";
++x;
}
os << "</P>";
    }
    // Files using this license:
os << "<P>This license applies to the following " << i->second.files.size() << " files:";
std::set<fs::path, path_less>::const_iterator m, n;
m = i->second.files.begin();
n = i->second.files.end();
while(m != n)
{
os << split_path(m_boost_path, *m) << "<br>";
++m;
}
os << "</P>";
}
else
{
This license is used by " << i->second.authors.size() << " authors (list omitted for brevity).\n";
os << "This license applies to " << i->second.files.size() << " files (list omitted for brevity).\n";
++license_index;
++i;

// Output list of files not found to be under license control:
//
os << "Files With No Recognisable Licence\n";
"The following " << m_unknown_licenses.size() << " files had no recognisable license
information:\n";
std::set<fs::path, path_less>::const_iterator i2, j2;
i2 = m_unknown_licenses.begin();
j2 = m_unknown_licenses.end();
while(i2 != j2)
{
    os << split_path(m_boost_path, *i2) << "\n";
    ++i2;
}

os << "Files With No Recognisable Copyright Holder\n";
"The following " << m_unknown_authors.size() << " files had no recognisable copyright
holder:\n";
i2 = m_unknown_authors.begin();
j2 = m_unknown_authors.end();
while(i2 != j2)
{
    os << split_path(m_boost_path, *i2) << "\n";
    ++i2;
}

if(!m_bsl_summary_mode)
{
    // Output list of files that have been moved over to the Boost
    // Software License, along with enough information for human
    // verification.
    //
    os << "Files that can be automatically converted to the Boost Software
    License\n";
    << "The following " << m_converted_to_bsl.size() << " files can be automatically converted to the Boost
    Software License, but require manual verification before they can be committed to CVS:\n";
    if (!m_converted_to_bsl.empty())
    {
typedef std::map<fs::path, std::pair<std::string, std::string>, path_less>::const_iterator conv_iterator;

conv_iterator i = m_converted_to_bsl.begin(), ie = m_converted_to_bsl.end();

int file_num = 1;
while (i != ie)
{
    os << "<P>[" << file_num << "] File: <tt>" << split_path(m_boost_path, i->first) << "\n<br>" << "<tt><tr><td>\n \n <pre>" << i->second.first << "</pre></td>\n <td><pre>" << i->second.second << "</pre></td>\n</table>\n";
    ++i;
    ++file_num;
}

// Output list of files that could be moved over to the Boost Software License

//

os << "<h2><a name="to-bsl"></a>Files that could be converted to the Boost Software License</h2>

<P>The following " << m_can_migrate_to_bsl.size() << " files could be manually converted to the Boost Software License, but have not yet been:</p>
<BLOCKQUOTE><p>

i2 = m_can_migrate_to_bsl.begin();
j2 = m_can_migrate_to_bsl.end();
while(i2 != j2)
{
    os << split_path(m_boost_path, *i2) << "\n";
    ++i2;
}

os << "</p></BLOCKQUOTE>";

// Output list of files that can not be moved over to the Boost Software License

//

os << "<h2><a name="not-to-bsl"></a>Files that can NOT be converted to the Boost Software License</h2>

<P>The following " << m_cannot_migrate_to_bsl.size() << " files cannot be converted to the Boost Software License because we need the permission of more authors:</p>
<BLOCKQUOTE><p>

i2 = m_cannot_migrate_to_bsl.begin();
j2 = m_cannot_migrate_to_bsl.end();
while(i2 != j2)
{
    os << split_path(m_boost_path, *i2) << "\n";
    ++i2;
}

os << "</p></BLOCKQUOTE>";

// Output list of authors that we need permission for to move to the BSL

//

os << "<h2><a name="need-bsl-authors"></a>Authors we need for the BSL</h2>

";
Permission of the following authors is needed before we can convert to the Boost Software License. The list of authors that have given their permission is contained in `<code>more/blanket-permission.txt</code>`.

```cpp
std::copy(m_authors_for_bsl_migration.begin(), m_authors_for_bsl_migration.end(),
          std::ostream_iterator<std::string>(os, "<br>
"));
```

// output a table of copyright information:

```cpp
os << "<H2><a name="copyright"></a>Copyright Holder Information</H2><table border="1">
```

```cpp
std::map<std::string, std::set<fs::path, path_less> >::const_iterator ad, ead;
ad = m_author_data.begin();
ead = m_author_data.end();
while(ad != ead)
{
    os << "<tr><td>" << ad->first << "<td>";
    std::set<fs::path, path_less>::const_iterator fi, efi;
    fi = ad->second.begin();
efi = ad->second.end();
    while(fi != efi)
    {
        os << split_path(m_boost_path, *fi) << " ";
        ++fi;
    }
    os << "</td></tr><br>
";
    ++ad;
}
```

// output file dependency information:

```cpp
os << "<H2><a name="depend"></a>File Dependency Information</H2><BLOCKQUOTE><pre>
```
```
conv_iterator i = m_converted_to_bsl.begin(),
    ie = m_converted_to_bsl.end();
while(i != ie)
{
    bad_paths.insert(i->first);
    ++i;
} 
fi = bad_paths.begin();
efi = bad_paths.end();
os << "<P>For brevity, only files not under the BSL are shown</P>
";
while(fi != efi)
{
    os << split_path(m_boost_path, *fi);
    dep = m_dependencies.find(*fi);
    last_dep = m_dependencies.end();
    std::set<fs::path, path_less> seen_deps;
    if (dep != last_dep)
        while(true)
        {
            os << " -> ";
            if(fs::exists(m_boost_path / dep->second))
                os << split_path(m_boost_path, dep->second);
            else if(fs::exists(dep->second))
                os << split_path(fs::path(), dep->second);
            else
                os << dep->second.string();
            if(seen_deps.find(dep->second) != seen_deps.end())
            {
                os << " <I>(Circular dependency!)</I>";
                break; // circular dependency!!!
            }
            seen_deps.insert(dep->second);
            last_dep = dep;
            dep = m_dependencies.find(dep->second);
            if((dep == m_dependencies.end()) || (0 == compare_paths(dep->second, last_dep->second)))
                break;
        }
    os << "\n";
    ++fi;
}
os << "</pre></BLOCKQUOTE>\n";
}
os << "</body></html>";
if(!os)
{
    std::string msg("Error writing to ");
msg += m_dest_path.string();
msg += ".";
std::runtime_error e(msg);
boost::throw_exception(e);
}
}

1.2865 readable-stream 2.2.7

1.2865.1 Available under license:

Node.js is licensed for use as follows:

```
==
Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
==
```

This license applies to parts of Node.js originating from the
https://github.com/joyent/node repository:

```
==
Copyright Joyent, Inc. and other Node contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
```
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2866 hk2-api 2.2.0-b21

1.2866.1 Available under license:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or
C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor
either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell,
offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions
thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or
portions of such combination).

c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or
otherwise makes the Modifications available to a third party.

d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has
deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor
Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the
Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of
Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made
available in Source Code form and that Source Code form must be distributed only under the terms of this License.
You must include a copy of this License with every copy of the Source Code form of the Covered Software You
distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable
form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a
medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You
represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to
grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the
Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered
Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial
Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the
applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee
for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However,
you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must
make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone,
and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the
Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.
3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.


You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A
PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. section 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

cat hk2-api-2.2.0-b21-sources/META-INF/inhabitants/default
#
# DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
#
# Copyright (c) 2010-2011 Oracle and/or its affiliates. All rights reserved.
#
# The contents of this file are subject to the terms of either the GNU
# General Public License Version 2 only ("GPL") or the Common Development
# and Distribution License("CDDL") (collectively, the "License"). You
# may not use this file except in compliance with the License. You can
#
# obtain a copy of the License at
# https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
# or package/legal/LICENSE.txt. See the License for the specific
# language governing permissions and limitations under the License.
#
# When distributing the software, include this License Header Notice in each
# file and include the License file at package/legal/LICENSE.txt.
#
# GPL Classpath Exception:
# Oracle designates this particular file as subject to the "Classpath"
# exception as provided by Oracle in the GPL Version 2 section of the License
# file that accompanied this code.
#
# Modifications:
# If applicable, add the following below the License Header, with the fields
# enclosed by brackets {} replaced by your own identifying information:
# "Portions Copyright [year] [name of copyright owner]"
#
# Contributor(s):
# If you wish your version of this file to be governed by only the CDDL or
# only the GPL Version 2, indicate your decision by adding "[Contributor]
# elects to include this software in this distribution under the [CDDL or GPL
# Version 2] license." If you don't indicate a single choice of license, a
# recipient has the option to distribute your version of this file under
# either the CDDL, the GPL Version 2 or to extend the choice of license to
# its licensees as provided above. However, if you add GPL Version 2 code
# and therefore, elected the GPL Version 2 license, then the option applies
# only if the new code is made subject to such option by the copyright
# holder.
#
# handcoded
As a special exception, the copyright holders of this library give you
permission to link this library with independent modules to produce an
executable, regardless of the license terms of these independent
modules, and to copy and distribute the resulting executable under
terms of your choice, provided that you also meet, for each linked
independent module, the terms and conditions of the license of that
module. An independent module is a module which is not derived from
or based on this library. If you modify this library, you may extend
this exception to your version of the library, but you are not
obligated to do so. If you do not wish to do so, delete this
exception statement from your version.

GNU GENERAL PUBLIC LICENSE
   Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate
parts of the General Public License.  Of course, the commands you use may
be called something other than 'show w' and 'show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary.  Here is a sample; alter the names:

    Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

    <signature of Ty Coon>, 1 April 1989
    Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs.  If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
1.2867 jsprim 1.4.1

1.2867.1 Available under license:

Copyright (c) 2012, Joyent, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2868 node-ensure 0.0.0 0.0.0

1.2868.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2869 aop-alliance 1.3.1.7-5.ph2
1.2869.1 Available under license:
Public Domain

1.2870 msexcel-builder-extended 0.0.8
1.2870.1 Available under license:
The MIT License (MIT)

Copyright (c) 2013 Charlie Zheng

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2871 babel-plugin-syntax-dynamic-import
6.18.0
1.2871.1 Available under license:

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

MIT License

Copyright (c) 2014-2016 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

BSD License

For "regenerator" software

Copyright (c) 2014, Facebook, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2015 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2872 audience-annotations 0.5.0
1.2872.1 Available under license:

Apache Yetus - Audience Annotations
Copyright 2015-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2873 safe-json-stringify 1.2.0

1.2873.1 Available under license:
No license file was found, but licenses were detected in source scan.

The MIT License (MIT)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1110637689_1607050467.34/0/safe-json-stringify-1-2-0-3-tgz/package/README.md

1.2874 ajv-keywords 3.5.2

1.2874.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2875 @babel/helper-explode-assignable-expression 7.13.0
1.2875.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2876 mongodb 2.2.36 2.2.36

1.2876.1 Available under license:

MongoDB-core uses third-party libraries or other resources that may be distributed under licenses different than the MongoDB-core software.

In the event that we accidentally failed to list a required notice, please bring it to our attention through any of the ways detailed here:

mongodb-dev@googlegroups.com

The attached notices are provided for information only.

For any licenses that require disclosure of source, sources are available at https://github.com/mongodb/node-mongodb-native.

1) License Notice for nan.h

---------------------------
The MIT License (MIT)
---------------------------

Copyright (c) 2016 NAN contributors

NAN contributors listed at <https://github.com/nodejs/nan#contributors>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{ }" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2877 guava 11.0.2
1.2877.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

vendor: https://code.google.com/p/guava-libraries/
license: http://www.apache.org/licenses/LICENSE-2.0
source: https://code.google.com/p/guava-libraries/source/browse/?name=v11.0.2

1.2878 babel-plugin-transform-es2015-literals

6.22.0 6.22.0

1.2878.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2879 mongodb-core 2.1.20

1.2879.1 Available under license:

MongoDB-core uses third-party libraries or other resources that may be distributed under licenses different than the MongoDB-core software.

In the event that we accidentally failed to list a required notice, please bring it to our attention through any of the ways detailed here:

mongodb-dev@googlegroups.com

The attached notices are provided for information only.

For any licenses that require disclosure of source, sources are available at https://github.com/mongodb/node-mongodb-native.

1) License Notice for nan.h

-----------------------------------

The MIT License (MIT)

Copyright (c) 2016 NAN contributors

NAN contributors listed at <https://github.com/nodejs/nan#contributors>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
1.2880 superagent 2.3.0

1.2880.1 Available under license:
(The MIT License)

Copyright (c) 2014-2016 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2881 java-sizeof 0.0.5

1.2881.1 Available under license:
No license file was found, but licenses were detected in source scan.

/**
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
*/
*    http://www.apache.org/licenses/LICENSE-2.0
*    
*    Unless required by applicable law or agreed to in writing, software
*    distributed under the License is distributed on an "AS IS" BASIS,
*    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
*    See the License for the specific language governing permissions and
*    limitations under the License.
*/

Found in path(s):
*    /opt/cola/permits/1135892501_1613626827.26/0/java-sizeof-0-0-5-sources-1-
jar/com/carrotsearch/sizeof/Constants.java
*    /opt/cola/permits/1135892501_1613626827.26/0/java-sizeof-0-0-5-sources-1-
jar/com/carrotsearch/sizeof/RamUsageEstimator.java
No license file was found, but licenses were detected in source scan.

/**
*    This class contains black magic stuff based mostly on proprietary
*    APIs and hacks. Use at your own risk.
*/

Found in path(s):
*    /opt/cola/permits/1135892501_1613626827.26/0/java-sizeof-0-0-5-sources-1-
jar/com/carrotsearch/sizeof/BlackMagic.java

1.2882 okio 2.4.3
1.2882.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are
under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2883 batik-parser 1.13

1.2883.1 Available under license:

Apache Batik

Copyright 1999-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for Standardization for the definition of character entities used in the software's documentation.

This product includes images from the Tango Desktop Project (http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme (http://www.jesusda.com/projects/pasodoble).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2884 call-bind 1.0.2
1.2884.1 Available under license:

MIT License

Copyright (c) 2020 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2885 nanomatch 1.2.13

1.2885.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2886 istanbul-lib-source-maps 4.0.0
1.2886.1 Available under license:

Copyright 2015 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2887 babel-template 6.26.0
1.2887.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being
made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International.  SEE THE
ECMA
CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-
international.org/memento/codeofconduct.htm FOR
INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO
IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2888 socket.io-client 1.7.4 1.7.4
1.2888.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014 Guillermo Rauch

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2889 libflac 1.3.1

1.2889.1 Available under license :

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of "copyleft", which means that derivative
works of the document must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document
straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the
copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.
4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all
the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retile any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.
The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.
8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version
number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and
license notices just after the title page:

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.
Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

      (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under
3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- Copyright (c) 2004-2009 Josh Coalson -->
<!-- Copyright (c) 2011-2016 Xiph.Org Foundation -->
<!-- Permission is granted to copy, distribute and/or modify this document -->
<!-- under the terms of the GNU Free Documentation License, Version 1.1 -->
<!-- or any later version published by the Free Software Foundation; -->
<!-- with no invariant sections. -->
<!-- A copy of the license can be found at http://www.gnu.org/copyleft/fdl.html -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta name="author" content="Josh Coalson" />
<meta name="description" content="A free, open source codec for lossless audio compression and decompression"/>
<meta name="keywords" content="free,lossless,audio,codec,encoder,decoder,compression,compressor,archival,archive,archiving,backup,music"/>
<link rel="shortcut icon" type="image/x-icon" href="favicon.ico"/>
<link rel="stylesheet" type="text/css" href="flac.css"/>
<title>FLAC - license</title>
</head>

<body>

<div class="logo">
<a href="index.html"><img src="images/logo130.gif" alt="FLAC Logo" align="middle" border="0" hspace="0" /></a>
</div>

<div class="above_nav"></div>
<div class="navbar">
&nbsp;<a href="index.html">home</a> | 
&nbsp;<a href="faq.html">faq</a> | 
&nbsp;<a href="documentation.html">documentation</a> | 
&nbsp;<a href="developers.html">developers</a> | 
&nbsp;<a href="changelog.html">changelog</a> | 
&nbsp;<a href="http://xiph.org/flac">more</a>
</div>

<div class="below_nav"></div>
FLAC is a free codec in the fullest sense. This page explicitly states all that you may do with the format and software.

The FLAC and Ogg FLAC formats themselves, and their specifications, are fully open to the public to be used for any purpose (the FLAC project reserves the right to set the FLAC specification and certify compliance). They are free for commercial or noncommercial use. That means that commercial developers may independently write FLAC or Ogg FLAC software which is compatible with the specifications for no charge and without restrictions of any kind. There are no licensing fees or royalties of any kind for use of the formats or their specifications, or for distributing, selling, or streaming media in the FLAC or Ogg FLAC formats.

The FLAC project also makes available software that implements the formats, which is distributed according to Open Source licenses as follows:

The reference implementation libraries are licensed under the New BSD License. In simple terms, these libraries may be used by any application, Open or proprietary, linked or incorporated in whole, so long as acknowledgement is made to Xiph.org Foundation when using the source code in whole or in derived works. The Xiph License is free enough that the libraries have been used in commercial products to implement FLAC, including in the firmware of hardware devices where other Open Source licenses can be problematic. In the source code these libraries are called `libFLAC` and `libFLAC++`. The rest of the software that the FLAC project provides is licensed under the GNU General Public License (GPL). This software includes various utilities for converting files to and from FLAC format, plugins for audio players, et cetera. In general, the GPL allows redistribution as long as derived works are also made available in source code form according to compatible terms.

Neither the FLAC nor Ogg FLAC formats nor any of the implemented encoding/decoding methods are covered by any known patent.

FLAC is one of a family of codecs of the Xiph.org Foundation, all created according to the same free ideals. For some other codecs' descriptions of the Xiph License see the Speex and Vorbis license pages.

If you would like to redistribute parts or all of FLAC under different terms, contact the FLAC-dev mailinglist.

Open Source Used In DNAC Platform Antelope 11239
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.
To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices
      stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in
      whole or in part contains or is derived from the Program or any
      part thereof, to be licensed as a whole at no charge to all third
      parties under the terms of this License.

   c) If the modified program normally reads commands interactively
      when run, you must cause it, when started running for such
      interactive use in the most ordinary way, to print or display an
      announcement including an appropriate copyright notice and a
      notice that there is no warranty (or else, saying that you provide
      a warranty) and that users may redistribute the program under
      these conditions, and telling the user how to view a copy of this
      License. (Exception: if the Program itself is interactive but
      does not normally print such an announcement, your work based
      on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.
4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
Copyright (C) 2000-2009 Josh Coalson
Copyright (C) 2011-2016 Xiph.Org Foundation

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

- Neither the name of the Xiph.org Foundation nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2890 find-up 4.1.0 4.1.0

1.2890.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2891 encodeurl 1.0.2

1.2891.1 Available under license:
(The MIT License)

Copyright (c) 2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
1.2892 regexpu-core 2.0.0 2.0.0

1.2892.1 Available under license:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2893 is-plain-object 2.0.4

1.2893.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2894 babel-plugin-syntax-exponentiation-operator 6.13.0 6.13.0
1.2894.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2895 avatica-core 1.17.0
1.2895.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to...
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and where ever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Apache Calcite -- Avatica
Copyright 2012-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.2896 supports-color 2.0.0 2.0.0
1.2896.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2897 babel-plugin-transform-class-properties 6.24.1

1.2897.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2898 summernote 0.8.11 0.8.11

1.2898.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015~ Summernote Team (https://github.com/orgs/summernote/people)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2899 filename-regex 2.0.1
1.2899.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015, 2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2900 engine.io 1.8.5 1.8.5

1.2900.1 Available under license:
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2901 @babel/runtime 7.13.10
1.2901.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2902 jline 3.12.1
1.2902.1 Available under license :

Copyright (c) 2002-2018, the original author or authors. All rights reserved.

https://opensource.org/licenses/BSD-3-Clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution.

Neither the name of JLine nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2903 activation-api 1.1

1.2903.1 Available under license :

# Notices for Eclipse Project for JAF

This content is produced and maintained by the Eclipse Project for JAF project.

* Project home: https://projects.eclipse.org/projects/ee4j.jaf

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Distribution License v. 1.0, which is available at http://www.eclipse.org/org/documents/edl-v10.php.

SPDX-License-Identifier: BSD-3-Clause

## Source Code

The project maintains the following source code repositories:
# Third-party Content

This project leverages the following third party content.

JUnit (4.12)

* License: Eclipse Public License

Copyright (c) 2018 Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2904 blob 0.0.4

1.2904.1 Available under license :

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2905 send 0.13.1
1.2905.1 Available under license :
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2906 timed-out 2.0.0 2.0.0
1.2906.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2907 jetty-servlet 9.4.41.v20210516
1.2907.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0, or the Apache Software License
2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation
distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.

"Recipient” means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient
copyright rights in its Contribution, if any, to grant the copyright
license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under
its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties
      and conditions, express and implied, including warranties or
      conditions of title and non-infringement, and implied warranties or
      conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for
      damages, including direct, indirect, special, incidental and
      consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are
      offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such
      Contributor, and informs licensees how to obtain it in a reasonable
      manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained
within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore,
if a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and
other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such Commercial
Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the
remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS.

W I T H O U T W A R R A N T I E S OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

=================================================================================================

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
=================================================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----

Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----

Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util
org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.2908 slash 3.0.0 3.0.0

1.2908.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
1.2909 bouncy-castle 1.64

1.2909.1 Available under license:

Copyright (c) 2000-2019 The Legion of the Bouncy Castle Inc. (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following organisations and people have contributed to the Bouncy Castle Cryptography Package.

Thanks, may your castles never deflate!

Donors

The following people donated financially to help with the release of 1.63:

Joshua Hight

The following people donated financially to help with the release of 1.61:
<br />
Bihari Babu

The following people donated financially to help with the release of 1.60:

Jens Neuhalfen and perillamint.

The following people donated financially to help with the release of 1.59:

Brian Reid.

The following people and organisations donated financially to help with the release of 1.57:

Joop Kaashoek and Pexus LLC.

We also wish to acknowledge financial and collaborative support from <a href="https://www.cisco.com/">CISCO</a> and additional financial support from <a href="https://www.primekey.com/">PrimeKey</a> towards developing the EST API for RFC 7030 support.

The following people and organisations donated financially to help with the release of 1.56:

DidiSoft, Cotiviti, Atanas Krachev, Encryptomatic LLC, LogicalAnswersIncSupporter

The following people and organisations donated financially to help with the release of 1.55:

Digistamp, RAM NAG

The following people and organisations donated financially to help with the release of 1.54:

Lobster GmbH

The following people and organisations donated financially to help with the release of 1.53:

Sheba, Ishmal Bartley, and Li-Chang Johnny Lo

The following people and organisations donated financially to help with the release of 1.52:

lecker&#064;buetterchen.de, Charles Proxy, Gunny Mills, Morgan Le Douget, Ben Whitaker, and Emilio Navarrete Lineros.
The following people and organisations donated financially to help with the release of 1.5.1:

Gup & Boz @ Alki Seattle, Bytemine Gmbh, Ted Pennings, Atanas Krachev, PrimeKey Solutions AB, Martin Paljak, CorseraFri19980116, CPU Terminator, Lindsay Bradford, kares, Philius, and Aaron Anderson.

Organisations

Holders of <a href="https://www.cryptoworkshop.com">Crypto Workshop Support Contracts</a>. Without the consulting time left over from support contracts being contributed back to working on the Bouncy Castle APIs, progress would be impossible. You know who you are! <a href="http://www.atlassian.com">Atlassian Software Systems</a> donation of Confluence and JIRA licences.

<a href="http://www.grierforensics.com">Grier Forensics</a>, for collaborating in the development of the S/MIME Toolkit and DANE SMIMEA functionality.

TU-Darmstadt, Computer Science Department, RBG, for the initial lightweight client side TLS implementation, which is based on MicroTLS and for help with qTESLA implementation. MicroTLS was developed by Erik Tews under the supervision of Dipl.-Ing. Henning Baer and Prof. Max Muehlhaeuser. qTESLA assistance was provided by Nina Bindel and Yinhua Xu.

TU-Darmstadt, Computer Science Department, RBG, for the initial Post Quantum provider, which was based on the FlexiProvider. The FlexiProvider was developed by the Theoretical Computer Science Research Group at TU-Darmstadt, Computer Science Department, RBG under the supervision of Prof. Dr. Johannes Buchmann. More information on the history of FlexiProvider can be found at: <a href="http://www.flexiprovider.de">http://www.flexiprovider.de</a>

Voxeo Labs - sponsorship of the initial development of APIs for DTLS 1.0 (RFC 4347), DTLS-SRTP key negotiation (RFC 5764), and server side TLS 1.1 (RFC 4346) and tested WebRTC compatibility. More information on Voxeo Labs can be found at <a href="http://voxeolabs.com">http://voxeolabs.com</a>

Core Infrastructure Initiative - financial support towards developing the TLS API and JSSE provider that appeared in 1.5.6.

Additional CertPath testing and validation data from the CertPath testing tool developed by <a href="https://www.cryptosource.de">cryptosource GmbH</a> and <a href="https://www.mtg.de">media Transfer AG</a> both located in Darmstadt, Germany.

Micro Focus - additional support towards further developing the TLS/DTLS API and the BCJSSE provider.

People

Tito Pena &lt;Fortunato.Pena@AdNovum.CH&gt; - initial RC5 implementation

Michael Lee &lt;yfli@#064;mira.net&gt; - initial RC6 implementation, MD2 implementation

Nuno Santos &lt;nsantos@student.dei.uc.pt&gt; - finding toString bug in certificate object.
Brett Sealey &lt;bretts&#064;mortbay.com&gt; - fixing the missing return problem in JDKKeyFactory (affected SSLeay private keys).

Victor A. Salaman &lt;salaman&#064;teknos.com&gt; - fixing the bug in Cipher.java which caused it to ignore specified providers, fixing the bug in RSAKeyGenerator which caused keys to be occasionally produced 1 bit too small.

Eran Librach &lt;eran&#064;valicert.com&gt; - spotting and fixing the classLoader bug that occurs if javax.crypto and the provider aren't sharing the same classpath (occurs in JDK 1.3 and greater).

Jonathan Knudsen &lt;jonathan&#064;LearningPatterns.com&gt; - porting information and restrictions when using the lightweight library with the MIDP environment.

Markus Niedermann &lt;markus.niedermann&#064;softwired-inc.com&gt; - porting information and restrictions when using the lightweight library with the MIDP environment.

Mike Benham &lt;moxie&#064;thoughtcrime.org&gt; - detection and fixing of an incorrect weak key in the DES key generation support classes. Suggestions for simplifying DESedeParameter objects. Optimisations for the Blowfish engine and BufferedBlockCipher class.

Soren Hilmer &lt;hilmer&#064;mail.tele.dk&gt; - initial implementation of netscape certificate request classes.

Uwe Guenther &lt;uwe&#064;cscc.de&gt; - detection and fixing of 3 incorrect semi-weak keys in the DES key generation support classes.

Markus Bradtke &lt;mab&#064;informatik.uni-kiel.de&gt; - fixing of a logic error in the JDKKeyStore class.

Waclaw SIEREK &lt;waclaw.sieriek&#064;tpg.pl&gt; - fix to setOddParity in the DESParameter class. Assistance with adding ordering to X509 names for certificate generation, proper processing of byte strings in the ASN.1 package, further simplifications and additional classes to improve pkcs7 support, bug fixes in CertPath API.

Ly-Na Phu &lt;lyna.phu&#064;init-consulting.de&gt; - assistance in the addition of ISO 9796-1 padding.

Stefan Köpsell &lt;sk13&#064;mail.inf.tu-dresden.de&gt; - making the jdk 1.1 version of the collections API available. For further details see http://sourceforge.net/projects/jcf/

Carmen Bastiaans &lt;cbastiaa&#064;microbits.com.au&gt; - fixing the improper null pointer problem in the setting of certificates in the PKCS12 key store.

Tomas Gustavsson &lt;tomasmg&#064;primekey.se&gt; - initial implementation of the AuthorityInformationAccess, SubjectKeyIdentifier, AuthorityKeyIdentifier, CRLNumber, CRLReason, CertificatePolicies, V2TBSCertListGenerator, and X509V2CRLGenerator classes in the ASN.1 library. Additions to GeneralName class, other bug fixes in the X.509 package. Initial implementation of the CertificationRequest classes. getRevocationReason() patch for OCSP. Patch to SemanticsInformation to prevent ClassCastException.

Eugen Kuleshov &lt;euxx&#064;hotmail.com&gt; - optimisations for Blowfish, assistance with PKCS12/keytool interoperability.

Megan Woods &lt;meganwoods&#064;sekurafile.com&gt; - initial implementation of ECIES.

Christian Geuer-Pollmann &lt;geuerp&#064;apache.org&gt; - adding IVs to the AESWrap implementations. Initial implementation of DESedeWrap.

Michael M&#252;hle &lt;michael&#064;mouling.de&gt; - contributing the initial CertPath implementation and...
compatibility classes, fixing provider bug in JDK 1.1 java.security.cert.CertificateFactory compatibility class. </li>
<li>Michael Mansell &lt;me#064;michaelmansell.com&gt; - fixing the parsing of the empty DER set in the ASN.1 library. </li>
<li>Eike Recker &lt;eike.recker#064;gmx.de&gt; - fixing misspelling of provider reference for RSA/1 and RSA/2. </li>
<li>Chris Southern &lt;CSouthern#064;baltimore.com&gt; - fixing misuse of specified provider in the PKCS10 certification request class. </li>
<li>Sidney Markowitz &lt;sidney#064;sidney.com&gt; - fixing null pointed exception on unknown OID in X509Name class, initial implementation of the three AES engines. </li>
<li>Chris Kerr &lt;ckerr#064;filonet.ca&gt; - initial implementation of the cms, asn1.cms, and the mail/smime packages, assistance in simplifying the ASN.1 package, miscellaneous other optimisations, NIST CertPath certification test, PKIXPolicyNode class, CertPath subtree validation and policy tree construction. </li>We also wish to acknowledge the generosity of Filonet Corporation for allowing Chris to make the initial cms and mail/smime packages available to us. </li>
<li>Mike Bean &lt;mbean#064;lucentradius.com&gt; - fixing the fall through bug in the IV algorithm parameters class. </li>
<li>Martin Petraschek &lt;e9526225#064;student.tuwien.ac.at&gt; - fixing ASN1 tagging so tag values up to 30 are now supported. </li>
<li>Jess Garms &lt;jgarms#064;yahoo.com&gt; - fixing 112/168 key size bug for DESede key generation. </li>
<li>Mike Bremford &lt;mike#064;big.faceless.org&gt; - contributing the initial PKCS7 implementation. </li>
<li>Shankar Srinivasan &lt;sr002#064;yahoo.com&gt; - S/Mime interoperability testing and debugging. </li>
<li>Stef Hoeben &lt;ilsestef#064;skynet.be&gt; - adding Montgomery multiplication to the BigInteger class. </li>
<li>Klaudiusz Ciosk &lt;kciosk#064;max.com.pl&gt; - improving the compatibility of the SMIME package with the Sun JCE. </li>
<li>Thomas Houtekier &lt;Thomas.Houtekier#064;tectrade.net&gt; - S/Mime testing and debugging. Interoperability with Biztalk. </li>
<li>Don Hillsberry &lt;hillsber#064;dialcorp.com&gt; - S/Mime testing and debugging. </li>
<li>Kazuo Furuya &lt;kfuruya#064;infoteria.co.jp&gt; - fixing root certificate chaining bug in PKCS12 key store. </li>
<li>Jason Novotny &lt;jd&064;blal.gov&gt; - initial work on the openSSL PEM processing. </li>
<li>Joel Hockey &lt;joel.hockey#064;qsipayments.com&gt; - initial work on the openSSL PEM processing. </li>
<li>John Steenbruggen &lt;JohnS#064;geotrust.com&gt; - fixing CertificationRequestInfo to handle cert request info objects without attribute blocks. </li>
<li>Justin Chapweske &lt;justin#064;chapweske.com&gt; - ordering patch for Tiger message digest. </li>
<li>John Serox &lt;jserox#064;hotmail.com&gt; - fixing null pointer exception in constructor for ExtendedKeyUsage. Fixing of base OID bug in KeyPurposeId. Compliance of KeyUsage extension return value with security API. </li>
<li>Sascha Weinreuter &lt;Sascha.Weinreuter#064;cit.de&gt; - fixed SMIME saveChanges() bug. </li>
<li>Andre Wehnert &lt;av5#064;mail.inf.tu-dresden.de&gt; - fixing key schedule problem in RC5-64, fixing buffer cleaning issue in buffered block cipher. </li>
<li>Luigi Lo Iacono &lt;lo_iacono#064;etu-inf.uni-siegen.de&gt; - adding SIC mode to the blockciphers in the provider. </li>
<li>Tim Sakach &lt;tsakach#064;certivo.net&gt; - SMIME v2 compatibility patches. </li>
<li>Marcus Povey &lt;mpovey#064;brookes.ac.uk&gt; - adding the PGP mode to the lightweight API and the
Open Source Used In DNAC Platform Antelope

Sebastian Clauß &lt;sc2@inf.tu-dresden.de&gt; - adding randomness setting to the certificate and CRL generators.

Nicolas Bielza &lt;nicolas.bielza@alligacom.com&gt; - isolating the tagging bug in the ASN.1 library that was misrepresenting some ASN.1 constructed data types. Contributions to the streaming S/MIME classes.

Casey Marshall &lt;rsdio@metastatic.org&gt; - fixing the clone problem with Macs in the clean room JCE.

Rick Zeldes &lt;rick.zeldes@eds.com&gt; - initial code for CMS/SMIME CompressedData.

Jarek Gawor &lt;gawor@mcs.anl.gov&gt; - fixing ASN.1 sequence unpacking in BasicConstraints constructor.

Brett Neumeier &lt;random@rnd.cx&gt; - patch to OriginatorIdentifierOrKey object, improvements to encoders package, introduction of UrlBase64.

Graham Coles &lt;graham.coles@retail-logic.com&gt; - patch to isParityAdjusted in DESKeySpec.

Jouni von Kattche&lt;j.kattchee@seeburger.de&gt; - patch to SMIMEGenerator for preventing class cast exceptions with BodyParts containing Multipart objects.

Matteo Artuso &lt;matartuso@64;libero.it&gt; - picking up the possible overread in ASN1InputStream.

Julian Morrison &lt;julian@64;extropy.demon.co.uk&gt; - spotting the slow down in Diffie-Hellman key generation.

Elmar Sonnenschein &lt;es0@64;esosmail.de&gt; - fix to long conversion in clean room SecureRandom.

Jouni von Kattchee &lt;J.Kattchee@seeburger.de&gt; - patch to SMIMEGenerator for preventing class cast exceptions with BodyParts containing Multipart objects.

Bryan Lovquist &lt;bkl@cps.com.au&gt; - Other provider compatibility fixes for CMS signing.

Artem Portnoy &lt;Artem_Portnoy@ibi.com&gt; - generalisations for CMSProcessableBodyPart in S/MIME. Header fix for mime messages.

Johannes Nicolai &lt;johannes.nicolai@64;novosec.com&gt; - further enhancements to OCSP response generation, fix to CertificateID issuer.

Marc Doberva &lt;marc.doberva@64;lex-si.com&gt; - help in isolating the JSSE/BC RSA key issue.

Jan Dvorak &lt;jan.dvorak@64;mathan.cz&gt; - initial implementation of the light weight Null block cipher.

Joe Cohen &lt;jcohen@forumsys.com&gt; - converting the ArrayOutOfBoundsException in DERInputStream into what it should have been.

Chris Long&lt;aclong@ece.cmu.edu&gt; - adding public key decoding to PEMReader.

Hes Siemelink&lt;hes@64;izemail.com&gt; - finding BigInteger loop problem in prime generation.

Jiri Urbanec &lt;jiri.urbanec@64;logicacmg.com&gt; - patch to fix defect in DERBMPString.equals().

Justin Kolb &lt;jkolb@64;pristx.com&gt; - patch to DSA signature generation in OpenPGP. Fix for the unexpected "Unexpected end of ZLIB input stream" exception.

Ralf Hauser &lt;ralfhauser@64;gmx.ch&gt; - patch to exception handling in PublicKeyRing, PEMReader, 1.4 build script, X509 Certificate Factory, CertPathValidatorUtilities, fromAddress null check in SignedMailValidator, ReadOnceInputStream testing utility in MIME tests.

Michal Dvorak &lt;M_Dvorak@kb.cz&gt; - getNextUpdate patch for OCSP SingleResp.

Klaus Greve Fiorentini &lt;Klaus@novosec.com&gt; - array fix in PGP PublicKeyEncSessionPacket.
Olivier Refalo &lt;Olivier_Refalo@#064fpl.com&gt; - null pointer exception fix for JDK 1.3 CMSSignedData objects.

Mariusz Bandola &lt;mariusz.bandola@#064cryptotech.com.pl&gt; - patch to DERGeneralizedTime. Compliance patch for OCSP TBSRequest class. Patch to X509Name for delaing with general objects in sequences.

Brien Oberstein &lt;brien.oberstein@#064transacttools.net&gt; - patch to S2K algorithm in OpenPGP, initial PGP version 3 secret key support, initial PGP version 3 signature generation, RIPEMD160 addition to PGPUtility.

Ian Haywood &lt;iian@#064haywood.bpa.nu&gt; - addition of getSignatureType to PGPSignature.

Jonathan Edwards &lt;js4gull@#064mac.com&gt; - initial support for reading multiple files from a PGP key file.

Andrew Thornton &lt;andrew@#064caren.cam.ac.uk&gt; - patch for RSA PUBLIC KEY in PEMReader.

Gregor Leander &lt;gl@#064bos-bremen.de&gt; - initial parsing of multiple sequence entries in an X.500 Name.

Antoon Bosselaers &lt;Antoon.Bosselaers@#064esat.kuleuven.ac.be&gt; - help with RIPEMD320 implementation.

Peter Sylvester &lt;Peter.Sylvester@#064edelweb.fr&gt; - improvements to the ASN.1 BasicConstraints object.

Doug &lt;ummmm@#064myrealbox.com&gt; - addition of isEncryptionKey method to OpenPGP public keys.

Francois Staes &lt;fstaes@#064netconsult.be&gt; - improvements to DEBitString, DERGeneralizedTime and initial implementation of DERGeneralString, addition of settable signed object info to CMSSignedDataGenerator, patch to DH key agreement.

W.R. Dittmer &lt;wdittmer@#064es.vu.nl&gt; - patch to decoding of SignatureCreationTime in BCPG. Patch to PGPKeyPair to fix nullpointer exception.

Perez Paz Luis Alberto &lt;laperez@#064banxico.org.mx&gt; - patch to use of BitString in X.500 name.

James Wright &lt;James_Wright@#064harte-hanks.com&gt; - patches for dealing with "odd" ArmoredInputStreams.

Jim Ford &lt;jim@#064muirford.com&gt; - patch to PGPSecretKey to avoid null pointer exception on encoding secret keys, comments on KeyExpirationTime, getBitStrength for ElGamal keys. Signature creation time patch for newly created v4 signatures.

Michael Hausler &lt;haeusler@#064ponton-consulting.de&gt; - extra aliases for provider.

Sai Pullabhotla &lt;psai@#064linoma.com&gt; - fix to PGP compressed data generator to improve compression levels. Performance improvements for KeyBasedLargeFileProcessor.

Joseph Miller &lt;joseph@#064digieweb.net.nz&gt; - addition of ZeroBytePadding.

Lars &lt;xyz@#064sagemdenmark.dk&gt; - patch to explicit padded mode for CBC block cipher MAC.

Jeroen van Vianen &lt;jeroen@#064vianvianen.nl&gt; - the Signed and Encrypted mail example.

Jun Sun &lt;JSun@#064diversinet.com&gt; - patch to SecureRandom to work around problem in wtk 1.0.4 and wtk 2.1.

Petr Dukem &lt;pdukem@#064email.cz&gt; - patch to CMSSignedDataGenerator to allow it to work with PKCS11 providers.

Filipe Silva &lt;filipe.silva@#064wedoconsulting.com&gt; - patch to fix overread issue in BCPGInputStream.

Alpesh Parmar &lt;alps@#064linuxmail.org&gt; - patch for class cast problem in PGPPublicKey.getSignatures().

Jay Gengelbach &lt;jgengelbach@#064webmethods.com&gt; - patch to fix isSigningKey in PGPSecretKey class, patch to hashedPackets in PGP signatureGenerator, initial cut for indefinite length output.

Doug &lt;doug@#064tigerprivacy.com&gt; - public key ring patches for ElGamal Signatures, problem key ring data.
Matthew Mundy &lt;mmundy1@umbc.edu&gt; - infinite loop prevention patch to PKCS5S2ParametersGenerator.

Tom Cargill &lt;cargill@064profcon.com&gt; - spelling patch in provider.

Breitenstrom Christian &lt;C.Breitenstrom@064-systems.com&gt; - compatibility patch to SignaturePacket, DetachedSignatureProcessor.

Zanotti Mirko &lt;zanotti@064cad.it&gt; - patch to ordered equality test for X509Name.

Nicola Scendoni &lt;nscendoni@064babelps.it&gt; - patch to add sorting to CertPath validation.

Ville Skyttä &lt;ville.skytta@064iki.fi&gt; - patch to CRLDistPoint for cRLIssuer field. KeyStore compliance on add patches. DiffieHellman patch for provider compliance. Support for PEM object "TRUSTED CERTIFICATE". Exception handling patch in PEMReader. JavaDoc clean up.

Bruce Gordon &lt;bruce.gordon@064savvis.net&gt; - patch to secret key creation encoding NullPointerException in OpenPGP, speed up for BCPGInputStream.

Miles Whiteley &lt;Miles.Whiteley@064savvis.net&gt; - "223" fix for BCPGInputStream new packets.

Albert Moliner &lt;amoliner@064evintia.com&gt; - initial TSP implementation.

Carlos Lozano &lt;carlos@064evintia.com&gt; - initial TSP implementation, patch to SignerInformation for supporting repeated signers, initial updates for supporting repeated attributes in CMS.

Javier Delgadillo &lt;javi@064javi.codewarp.org&gt; - initial Mozilla PublicKeyAndChallenge classes.

Joni Hakala &lt;joni.hakala@064cern.ch&gt; - initial implementations of VOMS Attribute Certificate Validation, IetfAttrSyntax, and ObjectDigestInfo. We also wish to thank the &lt;a href="http://www.eu-egee.org">EGEE project</a> for making the work available.

Rolf Schillinger &lt;rolf@064sr-wum.de&gt; - initial implementation of Attribute Certificate generation.

Sergey Bahtin &lt;Sergey_Bahtin@064yahoo.com&gt; - fix for recovering certificate aliases in BKS and UBER key stores. Initial implementations of GOST-28147, GOST-3410, EC GOST-3410, GOST OFB mode (GOFB) and GOST-3411.

Franck Leroy &lt;Franck.Leroy@064keynectis.com&gt; - ANS.1 set sorting. Contributions to TSP implementation. Test vectors for Bleichenbacher's forgery attack.

Atsuhioko Yamakata &lt;ymnk@064jcraft.com&gt; - patch for improving use of Montgomery numbers in BigInteger library. Patch to use size of private exponent in DH parameters.

Nickolay Bolshakov &lt;tyrex@064rekssoft.ru&gt; - patch for class cast exception in AuthorityInformationAccess class.

Soren Hilmer &lt;soren.hilmer@064tietoenator.com&gt; - patches for CertID with issuerSerial set in TSP implementation, additional compliance testing.

Steve Mitchell &lt;mitchell@064intertrust.com&gt; - patch for stateful path validator fix. Patch to allow BigInteger class to create negative numbers from byte arrays. Additions to allow different providers to be used for asymmetric/symmetric encryption in OpenPGP.

Optimisation to avoid redundant verification in path validator. Suggestion to use PKIXParameters.getSigProvider() correctly.

Dirk Eisner &lt;D.Eisner@seeburger.de&gt; - initial implementation of ISO 7816-4 padding.

Julien Pasquier &lt;julienpasquier@064free.fr&gt; - initial implementation of attribute classes from RFC 3126. Fix to KEKIdentifier, OtherKeyAttribute parsing. Initial ContentHints class.

Matteo &lt;matartuso@064libero.it&gt; - sequence patch to ASN1Dump.

Andrew Paterson &lt;andrew.paterson@064burnsec.com&gt; - patches to PGP tools, isRevoked method on PGPPublicKey.

Vladimir Molotkov &lt;vladimir.n.molotkov@064intel.com&gt; - extensive provider exception handling compliance testing.

Florin Kollan &lt;adlocflo@064web.de&gt; - fix to ElGamalKeyParameters equality testing.

Pavel Vassiliev &lt;paulvas@064gmail.com&gt; - Initial GOST28147Mac implementation.

Tom Pesman &lt;tom@064tnux.net&gt; - addition of DES-EDE encryption for RSAPrivate keys to...
PEMWriter.</li>
<li>Lukasz Kowalczyk &lt;lukasz.b.kowalczyk@#064gmail.com&gt; - patch to fix parsing issue with OpenSSL PEM based certificate requests.</li>
<li>Arndt Hasch &lt;Arndt.Hasch@#064maxence.de&gt; - additional fix for partial reading with new style PGP packets.</li>
<li>Fix Bernd (KCDP 11) &lt;bernd.fix@#064credit-suisse.com&gt; - fix for 31 byte issue and exception throwing by Whirlpool.</li>
<li>David M. Lee &lt;dmlee@#064crossroads.com&gt; - code for add and remove secret key in the PGPSecretKeyRing class. Additions to S/MIME and CMS unit tests.</li>
<li>Mike Dillon &lt;md5@#064embody.org&gt; - additional checks for PGP secret and public key construction, patches to copyWithNewPassword.</li>
<li>tu-vi cung &lt;t2cung@#064hotmail.com&gt; - patch for out of bounds problem in getDecoderStream method.</li>
<li>Chris Schultz &lt;cschultz@#064gmail.com&gt; - fix for InputStream constructor for X509V2AttributeCertificate.</li>
<li>David M. Lee &lt;dmlee@#064crossroads.com&gt; - implementation assistance with streaming CMS classes.</li>
<li>Joel Rees &lt;rees@#064ddcom.co.jp&gt; - fix to correct getOID methods from returning same set on X.509 attribute certificates.</li>
<li>Francesc Sau &lt;francesc.sau@#064partners.netfocus.es&gt; - micro fix for tsp Accuracy class.</li>
<li>Larry Bugbee &lt;lbugbee@#064mac.com&gt; - initial ECNR implementation.</li>
<li>Remi Blancher &lt;Remi.Blancher@#064keynectis.com&gt; - Contributions to TSP implementation. Initial implementation of RFC 3739 and ICAO ASN.1 classes.</li>
<li>Brian O'Rourke &lt;bianorourke@#064gmail.com&gt; - patch for signature creation time override in OpenPGP.</li>
<li>Andreas Schwier &lt;andreas.schwier@#064cardcontact.de&gt; - initial implementation of ISO9797 MAC Algorithm 3, addition of DES-EDE 64 MAC to the provider, fix to EC point encoding, addition of EC and RSA-PSS OIDs to CMS, addition of AES-CMAC and DESede-CMAC to JCE provider.</li>
<li>David Josse &lt;david.josse@#064transacttools.net&gt; - Patch for trailer function in version 2 signature packets.</li>
<li>Kishimoto Kazuhiko &lt;kazu-k@#064hi-ho.ne.jp&gt; - RFC 3280 updates to policy processing in the CertPath validator. Additional test data not covered by NIST.</li>
<li>Lawrence Tan &lt;lwrnctan@#064gmail.com&gt; - Large field OID sample test data. Missing key types in JDKKeyFactory.</li>
<li>Carlos Valiente &lt;superdupont@#064gmail.com&gt; - Addition of CRL writing to the PEMWriter class.</li>
<li>Keyon AG, Martin Christinat, &lt;a href="http://www.keyon.ch">http://www.keyon.ch</a&gt; - fixing incorrect ASN.1 encoding of field elements in X9FieldElement class.</li>
<li>Olaf Keller, &lt;olaf.keller.bc@#064bluewin.ch&gt; - initial implementation of the elliptic curves over binary fields F2m. Additional tests and modifications to elliptic curve support for both F2m and Fp. Performance improvements to F2m multiplication. Initial implementation of WNAF/WTNAF point multiplication. Improvement to k value generation in ECDSA.</li>
<li>J&ouml;rg Eichhorn &lt;eichhorn@#064ponton-consulting.de&gt; - patch to fix EOF read on SharedFileInputStream, support for F2m compression.</li>
<li>Karsten Ohme &lt;widerstand@#064t-online.de&gt; - initial check against for out of range data on non byte aligned RSA keys. Addition of equals/hashCode on ECCurve.Fp. Additional curve type support for Fp, contributions to F2m compression. F2m decoding for ECPointUtil. Infinity fix and prime192v2 fix for Fp. Extra validation for RSA key creation. Fix to name typos for some OpenSSL key generators. RFC-1779 table, improved RFC 2253 compliance for X509Name. Additional constructor validation for X.509/ESS ASN.1 classes. Validation
for Printable, IA5, and Numeric Strings. Fix for RFC 5280 NameConstraint checking for RDNs.<li>
Support for surrogate pairs in DERUTF8String, DER UTF8 test. Additional X.509 name attributes for ISIS-MTT, RFC 3039, addition of indirect CRL support, initial X509 LDAP CertStore implementation, CertificatePair class, and X509/CertificatePair class. Contributions to X509/Store/Parser infrastructure and design. CertPath support for implicit DSA parameters and a range of NameConstraints. Addition of support for V1 attribute certificates and attribute certificate path validation. Initial classes for ASN.1 ISIS-MTT support. Enhancements for improving compliance with the NIST CertPath tests.<li>
Carlos Lozano Ruiz &lt;carlos@tradise.com&gt; - patch for &amp;lt;ctrl&amp;gt;&amp;lt;m&amp;gt; only handling in CRLOutputStream.<li>
John Alfred Prufrock &lt;j.a.prufrock&amp;#064gmail.com&gt; - mods to GOST-3411 and MD2 to support ExtendedDigest.<li>
Stefan Neusatz Guilhen &lt;sneusatz&amp;#064gmail.com&gt; - initial version of RoleSyntax, improvements to AttributeCertificateHolder and AttributeCertificateIssuer.<li>
Marzio Lo Giudice &lt;marzio.logiudice&amp;#064gmail.com&gt; - fix to endianess in KDF2BytesGenerator, additional KDF2 tests.<li>
Georg Lippold &lt;georg.lippold&amp;#064gmx.de&gt; - initial implementation of NaccacheStern cipher.<li>
Chris Viles &lt;chris_viles&amp;#064yahoo.com&gt; - fix to SignatureSubpacket critical bit setting.<li>
Pasi Eronen &lt;Pasi.Eronen&amp;#064nokia.com&gt; - extra toString() support for ASN.1 library. Initial patch for large OID components.<li>
Maria Ivanova &lt;maria.ivanova&amp;#064gmail.com&gt; - support for tags &gt; 30 in ASN.1 parsing.<li>
Armin H&auml;ruml;ferling &lt;arminha&amp;#064student.ethz.ch&gt; - first cut of internationalisation, initial PKIX validation classes.<li>
Marius Schilder &lt;mschilder&amp;#064google.com&gt; - main set of test vectors for Bleichenbacher's forgery attack.<li>
Xavier Le Vouch &lt;xavier&amp;#064brittanysoftware.com&gt; - general code clean ups.<li>
Erik Tews &lt;e_tews&amp;#064de.informatik.tu-darmstadt.de&gt; - initial threaded random seed generator, constant-time PKCS#1.5 decoding.<li>
Thomas Dixon &lt;treykomusha&amp;#064gmail.com&gt; - initial implementations of TEA/XTEA, Salsa20, ISAAC, and Noekeon. XTEA enhancements.<li>
Frank Cornelis &lt;info&amp;#064frankcornelis.be&gt; - addition of crlAccessMethod in X509ObjectIdentifiers.<li>
Rui Joaquim &lt;rjoaquim&amp;#064cc.isel.ipl.pt&gt; - initial implementation of RSA blinding for signatures.<li>
David Stacey &lt;DStacey&amp;#064allantgroup.com&gt; - addition of trust packet checking on revocation signatures in PGP/SecretKeyRing.<li>
Martijn Brinkers &lt;list&amp;#064mitm.nl&gt; - better exception handling in CMS enveloping, "just in time" modifications for CRL and Sequence evaluation.<li>
Julius Davies &lt;juilisdavies&amp;#064gmail.com&gt; - additional modes and algorithm support in PEMReader.<li>
Matthias &lt;g&amp;#064rttnner.de&gt; - GnuPG compatibility changes for PBEMFileProcessor.<li>
Olga K&amp;amp;uuml;thler &lt;olga.kaethler&amp;#0644hjp-consulting.com&gt; - initial implementation of TeleTrusT EC curves, additional ISO 9797 MACs, contributions to EAC OIDs, addition of EAC algorithms to CMS Signing.<li>
Germano Rizzo &lt;germano.rizzo&amp;#064gmail.com&gt; - initial implementation of CMac, EAX, HC-128, and HC-256, optimisations for Salsa20.
<li>Núria Marí &lt;numaa@hotmail.com&gt; - patch for alternate data type recognition in CMSSignedDataParser.</li>
<li>Janis Schuller &lt;js@#064tzi.de&gt; - addition of NotationData packets for OpenPGP.</li>
<li>Michael Samblanet &lt;mike@#064samblanet.com&gt; - patches towards improved Sun/default provider support in CMS.</li>
<li>Ramon Keller &lt;ramon.keller@#064gmx.ch&gt; - patch to deal with null revocations return from other CRL in X509V2CRLGenerator.</li>
<li>Mark Nelson &lt;mark@#064nbr.com&gt; - correction to excluded DN in name constraints processing for PKIX processing.</li>
<li>Eugene Golushkov &lt;eugene_gff@#064ukr.net&gt; - mask fix to single byte read in TlsInputStream.</li>
<li>Julien Pasquier &lt;julienpasquier@#064free.fr&gt; - additional classes for supporting signature policy and signer certificates in the ASN.1 esf and ess libraries.</li>
<li>Patrick Knopp &lt;tpknopp@#064mtg.de&gt; - fix for named curve recognition in ECGOST key generation.</li>
<li>Jakub Gwozdz &lt;gwozdziu@#064rpg.pl&gt; - addition of getTsa() to TimeStampTokenInfo.</li>
<li>Bartosz Malkowski &lt;bmalkow@#064tigase.org&gt; - initial implementation of VMPC cipher, VMPCRandomGenerator, VMPCMac.</li>
<li>Tal Yacobi &lt;tal.yacobi@octavian-tech.com&gt; - fix for issue in OpenPGP examples [#BJA-55].</li>
<li>Massimiliano Ziccardi &lt;massimiliano.ziccardi@#064gmail.com&gt; - support for counter signature reading in CMS API, update for multiple counter signature attributes, JCA compliance patch for PEM parsing in CertificateFactory.</li>
<li>Andrey Pavlenko &lt;andrey.a.pavlenko@#064gmail.com&gt; - security manager patch for PKCS1Encoding property check.</li>
<li>1 Ross Nicoll &lt;jrn@#064jrn.me.uk&gt; - improved exception handling for getInstance() in ASN.1 library.</li>
<li>Matthew Stevenson &lt;mavricknz@#064yahoo.com&gt; - patch to constructor for CRMF CertSequence.</li>
<li>Gabriele Contini &lt;gcontini@#064hotpop.com&gt; - identified a bug in ASN.1 library with handling of unterminated NDEF's.</li>
<li>Roelof Naude &lt;roelof.naude@#064epiuse.com&gt; - patch for TLS client to send empty client certs in response to HP_CERTIFICATE_REQUEST.</li>
<li>Patrick Peck &lt;peck@#064signature.at&gt; - identified problem with DERApplicationSpecific and high tag numbers in ASN.1 library.</li>
<li>Michael LeMay &lt;lemaymd@#064lemaymd.com&gt; - identified problem with EAX [#BJA-93].</li>
<li>Alex Dupre &lt;ale@#064FreeBSD.org&gt; - fix to use of Signature rather than SignatureSpi in provider [#BJA-90]. Addition of null provider use to SignedPublicKeyAndChallenge and PKCS10CertificationRequest [#BJA-102].</li>
<li>Michael Schoene &lt;michael@#064sigrid-und-michael.de&gt; - fix of improper handling of null in ExtendedPKIXParameters.setTrustedACIssuers(), check for V2 signingCertificate attribute in TimeStampResponse.validate().</li>
<li>Ion Larrañaga &lt;ilarra@#064s21sec.com&gt; - fix to default partial packet generation in BCPGOutputStream.</li>
<li>Bob Kerns &lt;bob.kerns@positscience.com&gt; - fix to hashCode for X509CertificateObject.</li>
<li>Stefan Meyer &lt;stefan.meyer@#064ewe.de&gt; - backport for PKIXCertPathValidator and SMIMESignedMailReviewer.</li>
<li>Robert J. Moore &lt;Robert.J.Moore@#064allanbank.com&gt; - speedups for OpenPGPCFB mode, clean room JCE patches.</li>
<li>Rui Hodai &lt;rui@#064po.ntts.co.jp&gt; - speed ups for Camellia implementation, CamelliaLightEngine.
Emir Bucalovic &lt;emir.bucalovic@#064mail.com&gt; initial implementation of Grain-v1 and Grain-128.

Torbjorn Svensson &lt;tobbe79@#064gmail.com&gt; initial implementation of Grain-v1 and Grain-128.

Paul FitzPatrick &lt;bouncycastle_pfitz@#064fitzpatrick.cc&gt; error message fix to X509LDAPCertStoreSpi, comparison fix to BCStrictStyle.

Henrik Andersson &lt;k.henrik.andersson@#064gmail.com&gt; addition of UniqueIssuerID to certificate generation.

Cagdas Cirit &lt;cagdascirit@#064gmail.com&gt; subjectAlternativeName fix for x509CertStoreSelector.

Harakiri &lt;harakiri_23@#064yahoo.com&gt; datahandler patch for attached parts in SMIME signatures.

Pedro Henrique &lt;pmhenriques@#064gmail.com&gt; explicit bounds checking for DESMIME signatures, code simplification for OAPIDerecoding.

Lothar Kimmelinger &lt;job@#064kimmeringer.de&gt; verbose mode for ASN1Dump, support for DERExternal, DNS performance fix for SMIME API.

Richard Farr &lt;r.farr.se@#064gmail.com&gt; initial SRP-6a implementation.

Thomas Castiglione &lt;castiglione@#064au.ibm.com&gt; patch to encoding for CRMF OptionalValidity.

Elisabetta Romani &lt;eromani@#064sogei.it&gt; patch for recognising multiple counter signatures.

Robin Lundgren &lt;737lundgren@#064gmail.com&gt; CMPCertificate constructor from X509CertificateStructure fix.

Petr Kadlec &lt;mormegil@#064centrum.cz&gt; fix to sign extension key and IV problem in HC-128, HC-256.

Andreas Antener &lt;antener_a@#064gmx.ch&gt; fix to buffer reset in AsymmetricBufferedBlockCipher.

Harendra Rawat &lt;hsrawat@#064yahoo.com&gt; fix for BERConstructedOctetString.

Rolf Lindemann &lt;lindemann@#064trustcenter.de&gt; patch for PKCS12 key store to support more flexible attribute specifications [BMA-42].

Alex Artamonov &lt;alexart.home@#064gmail.com&gt; name look up patch for GOST-2001 parameters.

Mike Lyons &lt;mlyons@#064layer7tech.com&gt; work arounds for EC JDK bug 6738532 and JSSE EC naming conventions.

Chris Cole &lt;chris_h_cole@#064yahoo.com&gt; identified a problem handling null passwords when loading a BKS keystore.

Tomas Krivanek &lt;tom@#064atack.cz&gt; added checking of Sender header to SignedMailValidator.

Michael &lt;emfau@#064t-online.de&gt; correction of field error in getResponse method in CertRepMessage.

Trevor Perrin &lt;trevor@#064cryptography.com&gt; addition of constant time equals to avoid possible timing attacks.

Markus Kiläring &lt;markus@#064primekey.se&gt; several enhancements to TimeStampResponseGenerator.

Dario Novakovic &lt;darionis@#064yahoo.com&gt; fix for NPE when checking revocation reason on CRL without extensions.

Michael Smith &lt;m.smith@#064cnbc.com&gt; bug fixes and enhancements to the CMP and CRMF classes, initial Master List classes.

Andrea Zilio &lt;andrea.zilio@#064gmail.com&gt; fix for PEM password encryption of private keys.

Alex Birkett &lt;alex@#064birkett.co.uk&gt; added support for EC cipher suites in TLS client (RFC 4492) [#BJA-291].

Wayne Grant &lt;wayne@#064gmail.com&gt; additional OIDs for PKCS10 and certificate generation support.

Frank Cornelis &lt;info@#064frankcornelis.be&gt; additional support classes for CAdES, enhancements to OCSP classes.

Jan Ditbberner &lt;jan@#064ditbberner.info&gt; addHeader patch for SMIME generator.

Bob McGowan &lt;boab.mcgoo@#064btinternet.com&gt; patch to support different object and mgf digests in
PSS signing.

- Ivo Matheis &lt;i.matheis&#064seeburger.de&gt;: fix to padding verification in ISO-9796-1.
- Marco Sandrini &lt;messche&#064gmail.com&gt;: patch to add IV to ISO9797Alg3Mac.
- Alf Malf &lt;alfilmalf&#064hotmail.com&gt;: removal of unnecessary limit in CMSContentInfoParser.
- Alfonso Massa &lt;alfonso.massa&#064insiel.it&gt;: contributions to CMS time stamp classes.
- Giacomo Boccardo &lt;gboccardo&#064unimaticaspa.it&gt;: initial work on CMSTimeStampedDataParser.
- Arnis Tartu &lt;arnis&#064ut.ee&gt;: patches for dealing with OIDs with specific key sizes associated in CMS.
- Janusz Sikociski &lt;j.Sikocinski&#064#064gdzie.pl&gt;: addition of Features subpacket support to OpenPGP API.
- Jurij Hudolejiev &lt;jhudolejiev&#064gmail.com&gt;: JavaDoc fix to CMSSignedDataParser.
- Liane Velten &lt;liane.velten&#064#064hpj-consulting.com&gt;: fine tuning of code for DHParameters validation.
- Shawn Willden &lt;swillden&#064google.com&gt;: additional functionality to PGPKeyRing.
- Atanas Krachev &lt;akrachev&#064gmail.com&gt;: added support for revocation signatures in OpenPGP.
- Mickael Laiking &lt;mickael.laiking&#064#064keynectis.com&gt;: initial cut of EAC classes.
- Tim Buktu &lt;tbuktu&#064#064hotmail.com&gt;: Initial implementation of NTRU signing and encryption.
- Bernd &lt;rbernd&#064gmail.com&gt;: Fix for open of PGP literal data stream with UTF-8 naming.
- Steing Inge Morisbak &lt;stein.inge.morisbak&#064#064BEKK.no&gt;: Test code for lower case Hex data in PEM headers.
- Andreas Schmid &lt;andreas.schmid&#064#064ntngtech.com&gt;: Additional expiry time check in PGPPublicKeys.
- Phil Steitz &lt;phil.steitz&#064gmail.com&gt;: Final patch eliminating JCE dependencies in the OpenPGP BC classes.
- Petar Petrov &lt;p.petrov&#064#064bers-soft.com&gt;: Testing and debugging of UTF-8 OpenPGP passwords.
- Daniel Fitzpatrick &lt;daniel.f.nwr&#064#064gmail.com&gt;: Initial implementation of ephemeral key support for IES, initial implementations of RSA-KEM and ECIES-KEM, initial implementation of homogeneous projective coordinates for EC.
- Andy Neilson &lt;Andy.Neilson&#064quest.com&gt;: a further patches to deal with multiple providers and PEMReader.
- Ted Shaw &lt;xiao.xj&#064gmail.com&gt;: patch to MiscPEMGenerator for handling new PKCS10CertificationRequests.
- Eleriseth &lt;Eleriseth&#064WPECGLtYbVi8Rl6Y7Vzl2Lvd2EUVW99v3yNV3IWROG8.fms&gt;: speed up for SIC/CTR mode. Provider compatibility generalisations for EC operations.
- Kenny Root &lt;kenny&#064#064the-b.org&gt;: patch for issuerAltName, subjectAltName support in X509CertificateObject. BaseBlockCipher.getIV() patch for AEAD.
- Maarten Bodewes &lt;maarten.bodewes&#064gmail.com&gt;: initial implementation of HKDF and NIST SP 800-108 MAC based KDF functions.
- Philip Clay &lt;pilf_b&#064#064yahoo.com&gt;: Initial implementation of J-PAKE.
- Brian Carlstrom &lt;bdc&#064#064carlstrom.com&gt;: compliance patches for some JCA/JCE keystore and cipher classes, miscellaneous code quality improvements, intial provider PBKDF2WithHmacSHA1 SecretKeyFactory.
- Samuel Lid&eacute;n Borell &lt;samuel&#064#064primekey.se&gt;: patch to add DSTU-4145 to DefaultSignatureAlgorithmFinder.
- Sergio Demian Lerner &lt;sergiolerner&#064#064certimix.com&gt;: pointing out isInfinity issue in ECDSASigner.
signature verification.</li>
<li>Tim Whittington &lt;Tim.Whittington@orionhealth.com&gt; patch to remove extra init call in CMac, additional of Memoable interface for Digest classes, initial implementation of GMAC, further correctness tests for IV and reset processing in OCB, CCM, and block cipher reset. Initial implementation of Skein, XSalsa20, ChaCha, reduced round Salsa20, Threefish, and the Poly1305 MAC. Documentation updates. Added OCB support to Nokeon and CAST6 in the provider, exception testing for CTS, optimisations for CCM, provider support for AAD cipher methods, safe CipherInput/OutputStream implementations for use with AAD and subsequent bug fixes, cleanup after IDEA patent expiry, work on JCE SipHash support, optimisations for AESFastEngine, further work on EncodableDigest for SHA-2 digests, contributions to BCrypt/OpenBSDBCrypt, PGP API documentation and code quality work.</li>
<li>Marcus Lundblad &lt;marcus.lundblad@primekey.se&gt; patch for working around JDK jarsigner TSP bug, optional setting of IssuerSerial in TimeStampTokenGenerator, additional extensions enhancement for time stamp token generation.</li>
<li>Andrey Zhozhin &lt;zhozhin@xrm.ru&gt; patch for override of TSP SignerInfo attributes.</li>
<li>Sergey Tiunov &lt;st5555d@064gmail.com&gt; initial cut of DVCS classes.</li>
<li>Damian Kolasa &lt;fatfredy@064gmail.com&gt; ASN1Sequence patch for class cast issue in X9Curve.</li>
<li>Ash Hughes &lt;ashley.hughes@064blueyonder.co.uk&gt; patches for supporting PGPSecretKeyRing/PGPSecretKeys encodings with empty private keys, initial code for PGPSignatureSubpacketVector.getEmbeddedSignatures().</li>
<li>Andreas Reiter &lt;andreas.reiter@iaik.tugraz.at&gt; Reported incomplete status of CertificateVerify processing in (D)TLS server, and provided fix.</li>
Kieran Miller <kieran.miller@gmail.com> initial implementation for RFC 5649 key wrap with padding.

Oliver Ehli <ehli@064arago.de> Additional support for BSI plain ECDSA in the provider.

Daniel Heldt <Daniel.Heldt@cryptovision.com> Initial support for encodable state message digests.

Robert Bushman <bouncycastle@064traxel.com> Clean up of DirectKeySignature example.

Maurice Aarts <aarts@064risecure.com> updated to KDF generator to follow NIST SP 800-108.

Franziskus Kiefer <https://github.com/franziskuskiefer> initial implementation of Cramer-Shoup.

KB Sriram <mail_kb@064yahoo.com> testing for odd encodings for PGP User Attribute Subpackets.

Marco Schulze <marco@064nightlabs.de> Reported verification bug in GenericSigner.


dstutz <https://github.com/dstutz> added iteration count setters to PKCS#12 PBE mac/key generator builders.

Tobias Wich <tobias.wich@064ecsec.de> Provided patch for TLS to work around servers sending Supported Elliptic Curves extension unexpectedly.

Hauke Mehrts <hauke@064hauke-m.de> TLS patch to add ECDHE_ECDSA CCM ciphersuites from RFC 7251.

Daniel Zimmerman <dmz@064galois.com> Further key quality improvements to RSAKeyPairGenerator.

Jens Kapitza <j.kapitza@schwarze-allianz.de> Iterable support in OpenPGP API, code cleanup in OpenPGP API.

Johan Eklund <johan@064primekey.se> update to RFC 6960 for OCSPObjectIdentifiers.

nikosn <https://github.com/nikosn> Fix to encoding of EC private keys to ensure encoding matches order length.

Axel von dem Bruch <axel-vdb@064riseup.net> Contributions to BCrypt/OpenBSDBCrypt, original version of Blake2bDigest.

Derek Atkins <derek@064ihtfp.com> Documentation fixes to X9ObjectIdentifiers.

Peter Jr Halicky <peto@064halicky.sk> Correction to notification/error message handling in SignedMailValidator.


Thomas Belot <thomas.belot@BC@064gmail.com> initial CertPathLoopTest for demonstrating stack overflow issue.

Rich DiCroce <https://github.com/rdicroce> Initial implementation of server-side TLS-SRP support. TLS API extension to support non-blocking usage.

Björn Kautler <https://github.com/Vampire> Refinements to cert path validation (authority key addition, certificate order preservation).

Dominik Schümann <https://github.com/dschuermann> method for returning signatures/verifications without user IDs on PGPPublicKey, method for exposing S2K in PGPSecretKey, constants for GNU protection modes in S2K classes, optional version header for armored output.

Michael <https://github.com/MSKnete> initial fix for bitStrength issue for OpenPGP EC keys, correction for generic type on RecipientInformationStore.

Tobias Wagner <tobias.wagner@064n-design.de> Fix SecureRandom handling in BcAsymmetricKeyWrapper [#BJA-536].

Sergio Giro <https://github.com/sgiro> Fixed adding of additional stores from CRL distribution point [#BJA-537]. Fixed missing null check for CRL certificate issuer [#BJA-538], removal of risky zeroisation code in PBE.java, check for salt in PBEKeys that require it.

bschuette <https://github.com/bschuette> Fixed typo in DefaultSignatureAlgorithmIdentifierFinder,
additional methods on CMSSignedDataParser.</li>
<li>Leonard Dallot &lt;https://github.com/dallotTazTag&gt; Fix to S2K usage of none on changing passwords on keys without passwords originally.</li>
<li>Jan Willem Janssen &lt;j.w.janssen+bouncycastle&#064#064lxtreme.nl&gt; Support for DSAParameters in lightweight SubjectPublicKeyInfoFactory, initial object signer verifier for BC lightweight EC.</li>
<li>Sebastian Oerding &lt;sebastian.oerding@robotron.de&gt; Fixes to toString() in x509.CertificatePolicies.</li>
<li>Kai Kramer &lt;kai.kramer&#064#064gmail.com&gt; Code to deal with orphaned chain certificates in the PKCS#12 KeyStore.</li>
<li>Benoit Charles &lt;benoit.charles&#064#064opentrust.com&gt; Fix for IES data length check on decryption.</li>
<li>akwizgran &lt;https://github.com/akwizgran&gt; Fixed clone of key in Blake2bDigest copy constructor, blake2b reset issue for variant keys.</li>
<li>Matthias Edelhoff &lt;Matthias.Edelhoff&#064#064cryptovision.com&gt; BasicConstraintsValidation pathlen fix in PKIX certpath classes.</li>
<li>Lukasz Deputat &lt;lukasz.deputat&#064#064gmail.com&gt; Fixed bugs in TlsUtils read methods [#BJA-592].</li>
<li>Justin Ludwig &lt;https://github.com/justinludwig&gt; Iterator fix for PGObjectFactory to handle stream packets at start of iterated data.</li>
<li>Andr&eacute;eacute; Berenguel &lt;https://github.com/aberenguel&gt; Fix to include ECNamedCurveSpec in EC AlgorithmParameterSpi</li>
<li>Slawomir Jaranowski &lt;https://github.com/slawekjaranowski&gt; Patch to make cipher/hash/signature name methods in PGP internal API public.</li>
<li>Andrey Vasilyev &lt;https://github.com/andrey-vasilyev&gt; Initial implementation of GOST R 34.11-2012.</li>
<li>William Glanton &lt;wglanton77&#064#064gmail.com&gt; Fixed bug in Poly1305 [#BJA-620].</li>
<li>jdvorak001 &lt;https://github.com/jdvorak001&gt; Speed improvements for ASN.1 ObjectIdentifier cache.</li>
<li>Joseph Naegle &lt;jnaegle&#064#064grierforensics.com&gt; Patch for handling multiple certificates in a DANE SMIMEA entry.</li>
<li>Andrew Bonventure &lt;https://github.com/andybons&gt; NullPointer patch for WNafUtil.</li>
<li>The Google Security Team (Project Wycheproof) &lt;https://github.com/google/wycheproof&gt; defect analysis and additional test cases for the provider.</li>
<li>Gorka Irazoqui &lt;girazoki&#064#064wpi.edu&gt; from Intel Security Center of Excellence &lt;https://security-center.intel.com/&gt; detection of the issue with AESFastEngine (CVE-2016-1000339), additional suggestions for improvement to hardening of AESEngine and finding cache sensitivities in EC key generation/signing.</li>
<li>Joerg Senekowitsch &lt;joerg.senekowitsch&#064#064veridos.com&gt; patch to deal with hard coded boolean in EAC ECDSApublicKey.</li>
<li>Na Yu &lt;na.yu&#064#064samsung.com&gt; Constructor patches to CMC PKIData.</li>
<li>Evangelos Karatsisios &lt;ekaratsisios&#064#064mtg.de&gt; Corrected use of explicit tagging in X.509 PolicyConstraints class.</li>
<li>VivleSoren &lt;https://github.com/VivleSoren&gt;; additional constructor for McElieceCCA2PrivateKeyParameters.</li>
<li>mtausig &lt;https://github.com/mtausig&gt;; JavaDoc fix for MCESSerializedDataGenerator.</li>
<li>Sebastian Wolfgang Roland &lt;sebastianwolfgang.roland&#064#064stud.tu-darmstadt.de&gt; Initial XMSS/XMSS-MT implementation.</li>
<li>didisoft &lt;https://github.com/didisoft&gt; test code for PGP signature removal involving user ids.</li>
Mike Safonov &lt;https://github.com/MikeSafonov&gt;: initial implementation of GOST3410-2012 for lightweight provider and JCA, parameters patches for ECGOST keys, initial implementation of GOST3412-2015, addition of fromExtensions() for CRLDistPoint.

Artem Storozhuk &lt;storojs72@gmail.com&gt;: initial implementation of DSTU7564 (digest) and DSTU7624 (cipher) and their associated modes.

Andreas Glaser &lt;andreas.glaser@0x4di.de.com&gt;: patch to recognise ANSSI curves for PKCS#10 requests.

codeborne &lt;https://github.com/chxp&gt;: patch to correct OIDs used in public key digest parameters for ECGOST-2012.

FauxFaux &lt;https://github.com/FauxFaux&gt;: patch for JDK 1.9 update to DRBG.java.

4garbage &lt;https://github.com/4garbage&gt;: patch to allow GOST3410-94 private keys encoded as integers.

ekszz &lt;https://github.com/ekszz&gt;: corrections to SM2 signer to include default identity value.

jminer &lt;https://github.com/jminer&gt;: fix to Blake2b for hashes in range of 2**64-127 to 2**64.

str4d &lt;https://github.com/str4d&gt;: initial implementation of Blake2s.

Scott Woodward &lt;scott@064bit3consulting.com&gt;: performance fixes for CTRSP800DRBG.

David Strawn &lt;https://github.com/isomarcte&gt;: fix for off by one error in SCRYPT bounds checking.

chris mcconn &lt;0xchrismcconn@064gmail.com&gt;: identification of serialisation issue with XMSS/XMSSMT private keys (see also CVE-2018-1000613).

ZZMarquis &lt;https://github.com/ZZMarquis&gt;: offset patches for SM2 encryption and decryption, improvement to Array constant time comparison.

Andreas Kretschmer &lt;https://github.com/Akretsch&gt;: NPE fix for CertTemplate.getVersion().

Armin Lunkeit, Michael Tautenhahn &lt;0xmtl&gt;: identification of M-R test issue on higher certainty values in RSA key pair generation.

Vincent Breitmoser &lt;https://github.com/Valodim&gt;: fix to ignore unnecessary checksum calculator on PGP secret key encryption.

Adam Vartanian &lt;https://github.com/flooey&gt;: use of ShortBuffer exception and buffer size pre-check in Cipher.doFinal().

Bernd &lt;https://github.com/ecki&gt;: Fix to make PGPUtil.pipeFileContents use buffer and not leak file handle.

Shartung &lt;https://github.com/shartung&gt;: Additional EC Key Agreement algorithms in support of German BSI TR-03111.

Paul Schaub &lt;https://github.com/vanitasvitae&gt;: bringing PGPSecretKey.getUserIds() into line with PGPPublicKey.getUserIds(). Exception message fix in BcPublicKeyDataDecryptorFactory. Additional tests on PGP key ring generation.

Nick of Nexxar &lt;https://github.com/nros&gt;: update to OpenPGP package to handle a broader range of EC curves.

catbref &lt;https://github.com/catbref&gt;: sample implementation of RFC 7748/Ed25519 (incorporated work from github users Valdim and str4d as well).

gerlion &lt;https://github.com/gerlion&gt;: detection of concurrency issue with pre-1.60 EC math library.

fgrieu &lt;fgrieu@fgrieu.com&gt;: detection and suggested fixes for possible timing vulnerability in OAEPEncoding and RSACoreEngine.

MTG &lt;https://github.com/mtgag&gt;: patch for decoding issues in PKIPublicationInfo and CertificatedKeyPair.

Andreas Gadermaier &lt;up.gadermaier@064gmail.com&gt;: initial version of Argon2 PBKDF algorithm.

Tony Washer &lt;tony.washer@yahoo.co.uk&gt;: review of qTesla, Java 1.9 module code, additional test code and debugging for GOST, DSTU, and ECNR algorithms. Initial lightweight implementation of the ZUC ciphers and macs.
Vincent Bouckaert &lt;https://github.com/veebee&gt; initial version of RFC 4998 ASN.1 classes.


Gabriel Sroka &lt;https://github.com/gabrielsroka&gt; corrected comments in RSA validation.

sarah-mdv &lt;https://github.com/sarah-mdv&gt; improvements to JceKeyTransRecipientInfoGenerator, tests for JournalingSecureRandom, initial implementation of JournaledAlgorithm.

Vincent Bouckaert &lt;https://github.com/veebee&gt; initial version of RFC 4998 ASN.1 classes.


Gabriel Sroka &lt;https://github.com/gabrielsroka&gt; corrected comments in RSA validation.

sarah-mdv &lt;https://github.com/sarah-mdv&gt; improvements to JceKeyTransRecipientInfoGenerator, tests for JournalingSecureRandom, initial implementation of JournaledAlgorithm.

Jesse Feinman &lt;https://github.com/jessefeinman&gt; performance optimisation in RSAKeyParameters.

Gillis95 &lt;https://github.com/Gillis95&gt; improved JSSE compatibility for setEnabledCipherSuites.

Haemin Yoo &lt;https://github.com/yoohaemin&gt; Javadoc fixes.

Antoine Toulme &lt;https://github.com/atoulme&gt; Initial implementation of EthereumIESEngine.

Golden Loooy &lt;https://github.com/looly&gt; Patch for addition of C1C3C2 mode to SM2Engine.

Moses Palm&ordm;&amp;quot;&lt;https://github.com/trueSec&gt;&amp;quot; TrueSec &lt;Henrik.Palmer&#064truesec.se&gt; Additional improvements to constant time comparisons.

Ren&ordm;&amp;acute; Korthaus &lt;https://github.com/securitykernel&gt; Update to XMSS/XMSS^MT OID values to bring them in line with RFC 8391.

THausherr &lt;https://github.com/THausherr&gt; Addition generic support for CMS/TSP functions.

Gaylor Bosson &lt;https://github.com/Giltoniel&gt; Initial implementation of Blake2xs.

gaellalire &lt;https://github.com/gaellalire&gt; Patch for unprotected PGP private keys as SExpr.

im-scooter &lt;https://github.com/im-scooter&gt; Patch for case-insensitive behaviour of Param-Z.

Nick hitchan &lt;https://github.com/hitchan&gt; Fix for typo in engineInitSign() in EdEc SignatureSpi.

dbusche &lt;https://github.com/dbusche&gt; Argon2 optimisations.

Daniel Heldt &lt;https://github.com/dheldt&gt; Fixing encodings in unicode tests to allow a wider range of Java compilers to work. Tweak to inheritance in JceKeyAgreeRecipient.

Ugochukwu Mmaduekwe &lt;https://github.com/Xor-el&gt; Fix for initially bugged legacy Integers.numberOfLeadingZeros method.

Gsealy &lt;https://github.com/Gsealy&gt; addition of PBKDF2withHmacSM3 to the provider.

aphuang2013 &lt;https://github.com/aphuang2013&gt; update to path validation in EST service for ClearPath EST.

Kevin Herron &lt;https://github.com/kevinherron&gt; Initial ChaCha20Poly1305 prototype.

vkreml &lt;https://github.com/vkreml&gt; GOST compliance change for DefaultCMSSignatureEncryptionAlgorithmFinder.

Tobias Ospelt &lt;tobias&#064pentagrid.ch&gt; Identification of 1.63 regression in ASN.1 parsing.

---

1.2910 babel-template 6.26.0 6.26.0

1.2910.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2911 parse-duration 0.1.2
1.2911.1 Available under license:

The MIT License

Copyright (c) 2013 Jake Rosoman <jkroso@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2912 lodash-assignin 4.2.0

1.2912.1 Available under license :

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally
maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2913 fast-deep-equal 1.1.0

1.2913.1 Available under license:
MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2914 buffer-fill 1.0.0

1.2914.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
1.2915 glew 1.3.1-1ubuntu0.16.04.2

1.2915.1 Available under license:

The OpenGL Extension Wrangler Library
Copyright (C) 2002-2007, Milan Ikits <milan ikits[@]ieee org>
Copyright (C) 2002-2007, Marcelo E. Magallon <mmagallo[@]debian org>
Copyright (C) 2002, Lev Povalahev
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The name of the author may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mesa 3-D graphics library
Version: 7.0
Copyright (C) 1999-2007 Brian Paul  All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2007 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

/*
 ** Copyright (c) 2007 The Khronos Group Inc.
 **
 ** Permission is hereby granted, free of charge, to any person obtaining a
 ** copy of this software and/or associated documentation files (the
 ** "Materials"), to deal in the Materials without restriction, including
 ** without limitation the rights to use, copy, modify, merge, publish,
 ** distribute, sublicense, and/or sell copies of the Materials, and to
** permit persons to whom the Materials are furnished to do so, subject to
** the following conditions:
**
** The above copyright notice and this permission notice shall be included
** in all copies or substantial portions of the Materials.
**
** THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
** EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
** MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
** IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
** CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
** TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
** MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.
*/
/*
* Mesa 3-D graphics library
* Version:  7.0
* 
* Copyright (C) 1999-2007 Brian Paul All Rights Reserved.
* 
* Permission is hereby granted, free of charge, to any person obtaining a
* copy of this software and associated documentation files (the "Software"),
* to deal in the Software without restriction, including without limitation
* the rights to use, copy, modify, merge, publish, distribute, sublicense,
* and/or sell copies of the Software, and to permit persons to whom the
* Software is furnished to do so, subject to the following conditions:
* 
* The above copyright notice and this permission notice shall be included
* in all copies or substantial portions of the Software.
* 
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
* OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
* BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
* AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
* CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

/*
 ** The OpenGL Extension Wrangler Library
 ** Copyright (C) 2008-2017, Nigel Stewart <nigels[]users.sourceforge.net>
 ** Copyright (C) 2002-2008, Milan Ikits <milan[ki]nts.iee.org>
 ** Copyright (C) 2002-2008, Marcelo E. Magallon <mmagallo[de]debian.org>
 ** Copyright (C) 2002, Lev Povalahev
 ** All rights reserved.
 */
** Redistribution and use in source and binary forms, with or without
** modification, are permitted provided that the following conditions are met:
**
** * Redistributions of source code must retain the above copyright notice,
**   this list of conditions and the following disclaimer.
** * Redistributions in binary form must reproduce the above copyright notice,
**   this list of conditions and the following disclaimer in the documentation
**   and/or other materials provided with the distribution.
** * The name of the author may be used to endorse or promote products
**   derived from this software without specific prior written permission.
**
** THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
** AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
** IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
** ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
** LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
** CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
** SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
** INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
** CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
** ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
** THE POSSIBILITY OF SUCH DAMAGE.
*/

1.2916 jnr-netdb 1.1.6

1.2917 zip-stream 2.1.3
1.2917.1 Available under license :

Copyright (c) 2014 Chris Talkington, contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2918 hadoop-yarn-common 3.2.2

1.2918.1 Available under license:

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The binary distribution of this product bundles binaries of org.iq80.leveldb:leveldb-api (https://github.com/dain/leveldb), which has the following notices:

* Copyright 2011 Dain Sundstrom <dain@iq80.com>
* Copyright 2011 FuseSource Corp. http://fusesource.com

The binary distribution of this product bundles binaries of AWS SDK for Java - Bundle 1.11.563 (https://github.com/aws/aws-sdk-java), which has the following notices:

This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The binary distribution of this product bundles binaries of Gson 2.2.4,
which has the following notices:

The Netty Project
=================

Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based
WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging
facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'ArrayDeque', written by Josh
Bloch of Google, Inc:
This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
  * HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified version of Roland Kuhn's ASL2 AbstractNodeQueue, which is based on Dmitriy Vyukov's non-intrusive MPSC queue. It can be obtained at:

* LICENSE:
  * license/LICENSE.abstractnodequeue.txt (Public Domain)
  * HOMEPAGE:
  * https://github.com/akka/akka/blob/wip-2.2.3-for-scala-2.11/akka-actor/src/main/java/akka/dispatch/AbstractNodeQueue.java

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
  * HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
  * HOMEPAGE:
  * https://code.google.com/p/libdivsufsort/

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
  * HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:
This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the
equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/snappy/

This product contains a modified portion of UnsignedBytes LexicographicalComparator from Guava v21 project by Google Inc, which can be obtained at:

* LICENSE:
  * license/COPYING (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/guava

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * http://code.google.com/p/caliper/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which
can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains a modified portion of 'JDOM 1.1', which can be obtained at:

* LICENSE:
  * https://github.com/hunterhacker/jdom/blob/jdom-1.1/core/LICENSE.txt
* HOMEPAGE:
  * http://www.jdom.org/

The binary distribution of this product bundles binaries of Commons Codec 1.4, which has the following notices:

* src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.javacontains test data from http://aspell.net/test/orig/batch0.tab. Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)
* src/test/org/apache/commons/codec/language/bm/package-info.java contains text data from http://stevemorse.org/phoneticinfo.htm with permission from the original authors.

Original source copyright:Copyright (c) 2008 Alexander Beider & Stephen P. Morse.
The binary distribution of this product bundles binaries of Commons Lang 2.6, which has the following notices:
* This product includes software from the Spring Framework, under the Apache License 2.0 (see: StringUtils.containsWhitespace())

The binary distribution of this product bundles binaries of Apache Log4j 1.2.17, which has the following notices:
* ResolverUtil.java
  Copyright 2005-2006 Tim Fennell
Dumbster SMTP test server
  Copyright 2004 Jason Paul Kitchen
TypeUtil.java
  Copyright 2002-2012 Ramnivas Laddad, Juergen Hoeller, Chris Beams

The binary distribution of this product bundles binaries of
"Java Concurrency in Practice" book annotations 1.0, which has the following notices:
* Copyright (c) 2005 Brian Goetz and Tim Peierls Released under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.5)
Official home: http://www.jcip.net Any republication or derived work distributed in source code form must include this copyright and license notice.

The binary distribution of this product bundles binaries of
Jetty :: Http Utility 9.3.19.,
Jetty :: IO Utility 9.3.19.,
Jetty :: Security 9.3.19.,
Jetty :: Server Core 9.3.19.,
Jetty :: Servlet Handling 9.3.19.,
Jetty :: Utilities 9.3.19.,
Jetty :: Utilities :: Ajax,
Jetty :: Webapp Application Support 9.3.19.,
Jetty :: XML utilities 9.3.19.,
which has the following notices:
* =============================================================
  Jetty Web Container
  Copyright 1995-2016 Mort Bay Consulting Pty Ltd.
  ==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

-----
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

-----
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

-----
Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html
OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

The binary distribution of this product bundles binaries of Snappy for Java 1.0.4.1, which has the following notices:
* This product includes software developed by Google
  Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
(Apache 2.0 license)

This library containd statically linked libstdc++. This inclusion is allowed by "GCC RUnTime Library Exception"

== Contributors ==
* Tatu Saloranta
* Providing benchmark suite
* Alec Wysoker
* Performance and memory usage improvement

The binary distribution of this product bundles binaries of
Xerces2 Java Parser 2.9.1, which has the following notices:
* ==============WARNING Zone For License Version 2.0=================================
== NOTICE file corresponding to section 4(d) of the Apache License. ==
== Version 2.0, in this case for the Apache Xerces Java distribution. ==

Apache Xerces Java
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

The binary distribution of this product bundles binaries of
Logback Classic Module 1.1.2,
Logback Core Module 1.1.2,
which has the following notices:
* Logback: the reliable, generic, fast and flexible logging framework.
  Copyright (C) 1999-2012, QOS.ch. All rights reserved.

The binary distribution of this product bundles binaries of
Apache HBase - Annotations 1.2.6,
Apache HBase - Client 1.2.6,
Apache HBase - Common 1.2.6,
Apache HBase - Hadoop Compatibility 1.2.6,
Apache HBase - Hadoop Two Compatibility 1.2.6,
Apache HBase - Prefix Tree 1.2.6,
Apache HBase - Procedure 1.2.6,
Apache HBase - Protocol 1.2.6,
Apache HBase - Server 1.2.6,
which has the following notices:
* Apache HBase
  Copyright 2007-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

--

This product incorporates portions of the 'Hadoop' project

Copyright 2007-2009 The Apache Software Foundation

Licensed under the Apache License v2.0

--

Our Orca logo we got here: http://www.vectorfree.com/jumping-orca
It is licensed Creative Commons Attribution 3.0.
See https://creativecommons.org/licenses/by/3.0/us/
We changed the logo by stripping the colored background, inverting it and then rotating it some.

Later we found that vectorfree.com image is not properly licensed.
The original is owned by vectorportal.com. The original was relicensed so we could use it as Creative Commons Attribution 3.0.
The license is bundled with the download available here:

This product includes portions of the Bootstrap project v3.0.0

Copyright 2013 Twitter, Inc.

Licensed under the Apache License v2.0

This product uses the Glyphicons Halflings icon set.

http://glyphicons.com/

Copyright Jan Kovak

Licensed under the Apache License v2.0 as a part of the Bootstrap project.

--

This product includes portions of the Guava project v14, specifically
'hbase-common/src/main/java/org/apache/hadoop/hbase/io/LimitInputStream.java'

Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0

The binary distribution of this product bundles binaries of
Phoenix Core 4.7.0,
which has the following notices:
Apache Phoenix
Copyright 2013-2016 The Apache Software Foundation

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

This also includes:

The phoenix-spark module has been adapted from the phoenix-spark library
distributed under the terms of the Apache 2 license. Original source copyright:
Copyright 2014 Simply Measured, Inc.
Copyright 2015 Intersect Software Inc.

The file bin/daemon.py is based on the file of the same name in python-daemon 2.0.5
(https://pypi.python.org/pypi/python-daemon/). Original source copyright:
# Copyright 20082015 Ben Finney <ben+python@benfinney.id.au>
# Copyright 20072008 Robert Niederreiter, Jens Klein
# Copyright 20042005 Chad J. Schroeder
# Copyright 2003 Clark Evans
# Copyright 2002 Noah Spurrier
# Copyright 2001 Jrgen Hermann

The binary distribution of this product bundles binaries of
Plexus Cipher: encryption/decryption Component 1.4, which has the following notices:
* The code in this component contains a class - Base64 taken from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/Base64.java which is Apache license: http://www.apache.org/licenses/LICENSE-2.0

The PBE key processing routine PBECipher.createCipher() is adopted from http://juliusdavies.ca/svn/not-yet-commons-ssl/tags/commons-ssl-0.3.10/src/java/org/apache/commons/ssl/OpenSSL.java which is also Apache APL-2.0 license: http://www.apache.org/licenses/LICENSE-2.0

The binary distribution of this product bundles binaries of
software.amazon.ion:ion-java 1.0.1, which has the following notices:
* Amazon Ion Java Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

The binary distribution of this product bundles binaries of
joda-time:joda-time:2.9.9 which has the following notices:
* = NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

This product includes software developed by
Joda.org (http://www.joda.org/).

The binary distribution of this product bundles binaries of
Ehcache 3.3.1, which has the following notices:
* Ehcache V3 Copyright 2014-2016 Terracotta, Inc.

The binary distribution of this product bundles binaries of
snakeyaml (https://bitbucket.org/asomov/snakeyaml), which has the following notices:
* Copyright (c) 2008, http://www.snakeyaml.org

The binary distribution of this product bundles binaries of
swagger-annotations (https://github.com/swagger-api/swagger-core), which has the following notices:
* Copyright 2016 SmartBear Software

The binary distribution of this product bundles binaries of
metrics-core 3.2.4 which has the following notices:
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:
Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE HADOOP SUBCOMPONENTS:

The Apache Hadoop project contains subcomponents with separate copyright
notices and license terms. Your use of the source code for the these
subcomponents is subject to the terms and conditions of the following
licenses.

For the org.apache.hadoop.util.bloom.* classes:

/**
 * Copyright (c) 2005, European Commission project OneLab under contract
 * 034819 (http://www.one-lab.org)
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the following
 * conditions are met:
 * - Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * - Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the distribution.
 * - Neither the name of the University Catholique de Louvain - UCL
 * nor the names of its contributors may be used to endorse or
 * promote products derived from this software without specific prior
 * written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
* ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*/

For portions of the native implementation of slicing-by-8 CRC calculation
in src/main/native/src/org/apache/hadoop/util:

Copyright (c) 2008,2009,2010 Massachusetts Institute of Technology.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
* Neither the name of the Massachusetts Institute of Technology nor
  the names of its contributors may be used to endorse or promote
  products derived from this software without specific prior written
  permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other portions are under the same license from Intel:
http://sourceforge.net/projects/slicing-by-8/

/*++
* Copyright (c) 2004-2006 Intel Corporation - All Rights Reserved
*
* This software program is licensed subject to the BSD License,
* available at http://www.opensource.org/licenses/bsd-license.html

* Abstract: The main routine
* __*/

For src/main/native/src/org/apache/hadoop/io/compress/lz4/[lz4.h,lz4.c,lz4hc.h,lz4hc.c],

/*@ */

LZ4 - Fast LZ compression algorithm
Header File
Copyright (C) 2011-2014, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :
- LZ4 source repository : http://code.google.com/p/lz4/
- LZ4 public forum : https://groups.google.com/forum/#!forum/lz4c
*/

For hadoop-common-project/hadoop-common/src/main/native/gtest

---------------------------------------------------------------------
Copyright 2008, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
re2j 1.1

GO license
This is a work derived from Russ Cox's RE2 in Go, whose license http://golang.org/LICENSE is as follows:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-hdfs-project/hadoop-hdfs-native-client/src/main/native/fuse-dfs/util/tree.h
---------------------------------------------------------------------
Copyright 2002 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldbjni (https://github.com/fusesource/leveldbjni), which is available under the following license:

Copyright (c) 2011 FuseSource Corp. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of FuseSource Corp. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For hadoop-yarn-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-nodemanager/src/main/native/container-executor/impl/compat/{fstatat|openat|unlinkat}.h:

Copyright (c) 2012 The FreeBSD Foundation
All rights reserved.

This software was developed by Pawel Jakub Dawidek under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of leveldb (http://code.google.com/p/leveldb/), which is available under the following license:

Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles binaries of snappy (http://code.google.com/p/snappy/), which is available under the following license:

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.js
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dataTables.bootstrap.css
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery.dataTables.min.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/dt-1.10.7/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-full-2.0.0.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/dust-helpers-1.1.1.min.js

--------------------------------------------------------------------------------

Copyright (c) 2010 Aleksander Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/moment.min.js

--------------------------------------------------------------------------------

Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OPEN SOURCE USED IN DNAC PLATFORM ANTELOPE

OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The binary distribution of this product bundles these dependencies under the following license:

hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/bootstrap-3.3.7
hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/boostrap.min.js
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap.min.css
hadoop-tools/hadoop-sls/src/main/html/css/bootstrap-responsive.min.css
bootstrap v3.3.6
broccoli-asset-rev v2.4.2
broccoli-funnel v1.0.1
datatables v1.10.8
em-helpers v0.5.13
em-table v0.1.6
ember v2.2.0
ember-array-contains-helper v1.0.2
ember-bootstrap v0.5.1
ember-cli v1.13.13
ember-cli-app-version v1.0.0
ember-cli-babel v5.1.6
ember-cli-content-security-policy v0.4.0
ember-cli-dependency-checker v1.2.0
ember-cli-htmlbars v1.0.2
ember-cli-htmlbars-inline-precompile v0.3.1
ember-cli-ic-ajax v0.2.1
ember-cli-inject-live-reload v1.4.0
ember-cli-jquery-ui v0.0.20
ember-cli-qunit v1.2.1
ember-cli-release v0.2.8
ember-cli-shims v0.0.6
ember-cli-sri v1.2.1
ember-cli-test-loader v0.2.1
ember-cli-uglify v1.2.0
ember-d3 v0.1.0
ember-data v2.1.0
ember-disable-proxy-controllers v1.0.1
ember-export-application-global v1.0.5
ember-load-initializers v0.1.7
ember-qunit v0.4.16
ember-qunit-notifications v0.1.0
ember-resolver v2.0.3
ember-spin-spinner v0.2.3
ember-truth-helpers v1.2.0
jquery v2.1.4
jquery-ui v1.11.4
loader.js v3.3.0
momentjs v2.10.6
qunit v1.19.0
select2 v4.0.0
snippet-ss v1.11.0
spin.js v2.3.2
Azure Data Lake Store - Java client SDK 2.0.11
JCodeings 1.0.8
Joni 2.1.2
Mockito 1.8.5
JUL to SLF4J bridge 1.7.25
SLF4J API Module 1.7.25
SLF4J LOG4J-12 Binding 1.7.25

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
./hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/nvd3-1.8.5.* (css and js files)

Copyright (c) 2011-2014 Novus Partners, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the
License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For:

```
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-nvd3-1.0.9.min.js
```

The MIT License (MIT)

Copyright (c) 2014 Konstantin Skipor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:

```
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-1.6.4.min.js
hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/angular-route-1.6.4.min.js
```

The MIT License

Copyright (c) 2010-2017 Google, Inc. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/jquery-3.3.1.min.js
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/jquery.js
- hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jquery

Apache HBase - Server which contains JQuery minified javascript library version 1.8.3
Microsoft JDBC Driver for SQLServer - version 6.2.1.jre7

MIT License

Copyright (c) 2003-2017 Optimatika

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- oj! Algorithms - version 43.0

------------------------------------------------------------------------

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

For:

hadoop-yarn-project/hadoop-yarn/hadoop-yarn-common/src/main/resources/webapps/static/jt/jquery.jstree.js

--------------------------------------------------------------------------------

Copyright (c) 2014 Ivan Bozhanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For:
- hadoop-hdfs-project/hadoop-hdfs/src/main/webapps/static/d3-3.5.17.min.js

D3 is available under a 3-clause BSD license. For details, see:
- hadoop-tools/hadoop-sls/src/main/html/js/thirdparty/d3-LICENSE

The binary distribution of this product bundles these dependencies under the following license:
- HSQLDB Database 2.3.4

(HSQL License)

"COPYRIGHTS AND LICENSES (based on BSD License)

For work developed by the HSQL Development Group:

Copyright (c) 2001-2016, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ""AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL HSQ DEVELOPMENT GROUP, HSQDB.ORG,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For work originally developed by the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ""AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE HYPersonic SQL GROUP,
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the
Hypersonic SQL Group."
The binary distribution of this product bundles these dependencies under the following license:
Java Servlet API 3.1.0
servlet-api 2.5
jsp-api 2.1
jsr311-api 1.1.1
Glassfish Jasper 6.1.14
Servlet Specification 2.5 API 6.1.14

(CDDL 1.0)
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:
A. Any file that results from an addition to,
deletion from or modification of the contents of a file containing Original
Software or previous Modifications;
B. Any new file that contains any part of the Original Software
or previous Modification; or
C. Any new file that is contributed or otherwise made available
under the terms of this License.

1.10. Original Software means the Source Code and Executable form of
computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or
hereafter acquired, including without limitation, method, process, and apparatus
claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which
modifications are made and (b) associated documentation included in or
with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights
under, and complying with all of the terms of, this License. For legal entities,
You includes any entity which controls, is controlled by, or is under common control
with You. For purposes of this definition, control means (a) the power, direct
or indirect, to cause the direction or management of such entity, whether by
contract or otherwise, or (b) ownership of more than fifty percent (50%) of the
outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance
with Section 3.1 below and subject to third party intellectual property claims,
the Initial Developer hereby grants You a world-wide, royalty-free,
non-exclusive license:

(a) under intellectual property rights (other than
patent or trademark) Licensable by Initial Developer, to use, reproduce, modify,
display, perform, sublicense and distribute the Original Software (or portions
thereof), with or without Modifications, and/or as part of a Larger Work;
and

(b) under Patent Claims infringed by the making, using or selling of
Original Software, to make, have made, use, practice, sell, and offer for sale,
and/or otherwise dispose of the Original Software (or portions
thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are
effective on the date Initial Developer first distributes or otherwise makes the
Original Software available to a third party under the terms of this
License;
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent
Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform
recipients of any such Covered Software in Executable form as to how they can
obtain such Covered Software in Source Code form in a reasonable manner on or
through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are
governed by the terms of this License. You represent that You believe Your
Modifications are Your original creation(s) and/or You have sufficient rights to
grant the rights conveyed by this License.

3.3. Required Notices. You must
include a notice in each of Your Modifications that identifies You as the
Contributor of the Modification. You may not remove or alter any copyright,
patent or trademark notices contained within the Covered Software, or any
notices of licensing or any descriptive text giving attribution to any
Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code
form that alters or restricts the applicable version of this License or the
recipients rights hereunder. You may choose to offer, and to charge a fee for,
wrong, support, indemnity or liability obligations to one or more recipients
of Covered Software. However, you may do so only on Your own behalf, and not on
behalf of the Initial Developer or any Contributor. You must make it absolutely
clear that any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial Developer
and every Contributor for any liability incurred by the Initial Developer or
such Contributor as a result of warranty, support, indemnity or liability terms
You offer.

3.5. Distribution of Executable Versions. You may distribute the
Executable form of the Covered Software under the terms of this License or under
the terms of a license of Your choice, which may contain terms different from
this License, provided that You are in compliance with the terms of this License
and that the license for the Executable form does not attempt to limit or alter
the recipients rights in the Source Code form from the rights set forth in this
License. If You distribute the Covered Software in Executable form under a
different license, You must make it absolutely clear that any terms which differ
from this License are offered by You alone, not by the Initial Developer or
Contributor. You hereby agree to indemnify the Initial Developer and every
Contributor for any liability incurred by the Initial Developer or such
Contributor as a result of any such terms You offer.

3.6. Larger Works. You
may create a Larger Work by combining Covered Software with other code not
governed by the terms of this License and distribute the Larger Work as a single
product. In such a case, You must make sure the requirements of this License are
fulfilled for the Covered Software.
4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the
termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision
that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformulated only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
jersey-client 1.19
jersey-core 1.19
jersey-grizzly2 1.19
jersey-grizzly2-servlet 1.19
jersey-json 1.19
jersey-server 1.19
jersey-servlet 1.19
jersey-guice 1.19
Jersey Test Framework - Grizzly 2 Module 1.19
JAXB RI 2.2.3
Java Architecture for XML Binding 2.2.11
grizzly-framework 2.2.21
grizzly-http 2.2.21
grizzly-http-server 2.2.21
grizzly-http-servlet 2.2.21
1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.
1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.
1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.
1.4. Executable means the Covered Software in any form other than Source Code.
1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.
1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.
1.7. License means this document.
1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. Modifications means the Source Code and Executable form of any of the following:
   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;
   B. Any new file that contains any part of the Original Software or previous Modification; or
   C. Any new file that is contributed or otherwise made available under the terms of this License.
1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.
1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.
1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such
2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third
party.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted:
(1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from
this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient’s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.
You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.
When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE
IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT,
SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its
utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

The binary distribution of this product bundles these dependencies under the following license:
Protocol Buffer Java API 2.5.0

----------------------------------------------------------------------
This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicopsInternals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicopsInternals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

For:
XML Commons External Components XML APIs 1.3.04
--------------------------------------------------------------------------------

By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:
- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
- Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

The binary distribution of this product bundles these dependencies under the following license:
JUnit 4.11
Eclipse JDT Core 3.1.1
--------------------------------------------------------------------------------
(EPL v1.0)
Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
   where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone
acting on such Contributor’s behalf. Contributions do not include additions to
the Program which: (i) are separate modules of software distributed in
conjunction with the Program under their own license agreement, and (ii) are not
derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when
combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.

"Recipient” means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free copyright license to
reproduce, prepare derivative works of, publicly display, publicly perform,
distribute and sublicense the Contribution of such Contributor, if any, and such
derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed
Patents to make, use, sell, offer to sell, import and otherwise transfer the
Contribution of such Contributor, if any, in source code and object code form.
This patent license shall apply to the combination of the Contribution and the
Program if, at the time the Contribution is added by the Contributor, such
addition of the Contribution causes such combination to be covered by the
Licensed Patents. The patent license shall not apply to any other combinations
which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to
its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other intellectual
property rights of any other entity. Each Contributor disclaims any liability to
Recipient for claims brought by any other entity based on infringement of
intellectual property rights or otherwise. As a condition to exercising the
rights and licenses granted hereunder, each Recipient hereby assumes sole
responsibility to secure any other intellectual property rights needed, if any.
For example, if a third party patent license is required to allow Recipient to
distribute the Program, it is Recipient’s responsibility to acquire that license
before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright
rights in its Contribution, if any, to grant the copyright license set forth in
this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its
own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and
      conditions, express and implied, including warranties or conditions of title and
      non-infringement, and implied warranties or conditions of merchantability and
      fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for
      damages, including direct, indirect, special, incidental and consequential
      damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by
      that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor,
      and informs licensees how to obtain it in a reasonable manner on or through a
      medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the
Program.

Each Contributor must identify itself as the originator of its Contribution, if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore, if
a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses, damages
and costs (collectively "Losses") arising from claims, lawsuits and other legal
actions brought by a third party against the Indemnified Contributor to the
extent caused by the acts or omissions of such Commercial Contributor in
connection with its distribution of the Program in a commercial product
offering. The obligations in this section do not apply to any claims or Losses
relating to any actual or alleged intellectual property infringement. In order
to qualify, an Indemnified Contributor must: a) promptly notify the Commercial
Contributor in writing of such claim, and b) allow the Commercial Contributor to
control, and cooperate with the Commercial Contributor in, the defense and any
related settlement negotiations. The Indemnified Contributor may participate in
any such claim at its own expense.
For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient’s patent(s), then such Recipient’s rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient’s rights under this Agreement shall terminate if it fails to
comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

The binary distribution of this product bundles these dependencies under the following license:
JSch 0.1.54
ParaNamer Core 2.3
JLine 0.9.94
leveldbjni-all 1.8
Hamcrest Core 1.3
ASM Core 5.0.4
ASM Commons 5.0.2
ASM Tree 5.0.2

(3-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The binary distribution of this product bundles these dependencies under the following license:
FindBugs-jsr305 3.0.0
dnsjava 2.1.7, Copyright (c) 1998-2011, Brian Wellington. All rights reserved.

(2-clause BSD)
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.
The binary distribution of this product bundles these dependencies under the following license:
"Java Concurrency in Practice" book annotations 1.0

(CCAL v2.5)
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the
to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works; to create and reproduce Derivative Works; to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works; to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this license.
License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have
received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The binary distribution of this product bundles these dependencies under the following license:

jamon-runtime 2.4.1

(MPL 2.0)

Mozilla Public License
Version 2.0

1. Definitions

1.1. Contributor

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.
1.2. Contributor Version
means the combination of the Contributions of others (if any) used by a
Contributor and that particular Contributor's Contribution.

1.3. Contribution
means Covered Software of a particular Contributor.

1.4. Covered Software
means Source Code Form to which the initial Contributor has attached the notice
in Exhibit A, the Executable Form of such Source Code Form, and Modifications of
such Source Code Form, in each case including portions thereof.

1.5. Incompatible With Secondary Licenses
means

that the initial Contributor has attached the notice described in Exhibit B to
the Covered Software; or

that the Covered Software was made available under the terms of version 1.1 or
earlier of the License, but not also under the terms of a Secondary License.

1.6. Executable Form
means any form of the work other than Source Code Form.

1.7. Larger Work
means a work that combines Covered Software with other material, in a separate
file or files, that is not Covered Software.

1.8. License
means this document.

1.9. Licensable
means having the right to grant, to the maximum extent possible, whether at the
time of the initial grant or subsequently, any and all of the rights conveyed by
this License.

1.10. Modifications
means any of the following:

any file in Source Code Form that results from an addition to, deletion from, or
modification of the contents of Covered Software; or

any new file in Source Code Form that contains any Covered Software.

1.11. Patent Claims of a Contributor
means any patent claim(s), including without limitation, method, process, and
apparatus claims, in any patent Licensable by such Contributor that would be
infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. Secondary License
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. Source Code Form
means the form of the work preferred for making modifications.

1.14. You (or Your)
means an individual or a legal entity exercising rights under this License. For legal entities, You includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section
2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party
modifications of Covered Software, or (ii) the combination of its Contributions
with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its
Contributions.

This License does not grant any rights in the trademarks, service marks, or
logos of any Contributor (except as may be necessary to comply with the notice
requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute
the Covered Software under a subsequent version of this License (see Section
10.2) or under the terms of a Secondary License (if permitted under the terms of
Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are
its original creation(s) or it has sufficient rights to grant the rights to its
Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable
copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in
Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any
Modifications that You create or to which You contribute, must be under the
terms of this License. You must inform recipients that the Source Code Form of
the Covered Software is governed by the terms of this License, and how they can
obtain a copy of this License. You may not attempt to alter or restrict the
recipients rights in the Source Code Form.
3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.
4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an as is basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this
7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.
10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If you choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - Incompatible With Secondary Licenses Notice

This Source Code Form is Incompatible With Secondary Licenses, as defined by the Mozilla Public License, v. 2.0.

The binary distribution of this product bundles these dependencies under the following license:
JDOM 1.1

/*--
Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

*/--

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:

"This product includes software developed by the JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

The binary distribution of this product bundles these dependencies under the following license:
HBase Server 1.2.4

This project bundles a derivative image for our Orca Logo. This image is available under the Creative Commons By Attribution 3.0 License.
Creative Commons Legal Code

Attribution 3.0 Unported

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be
considered an Adaptation (as defined above) for the purposes of this License.

c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to
broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.
The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution (“Attribution Parties”) in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URL, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by
this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination
a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant
jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at https://creativecommons.org/.


Copyright (C) 2007 The Guava Authors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under
the License.

1.2919 aws4 1.10.1

1.2919.1 Available under license:

Copyright 2013 Michael Hart (michael.hart.au@gmail.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2920 delayed-stream 1.0.0 1.0.0

1.2920.1 Available under license:

Copyright (c) 2011 Debuggable Limited <felix@debuggable.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2921 jsreport-text 2.0.0
1.2921.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license
document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this
License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer
system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
Version.

e) Provide Installation Information, but only if you would otherwise
be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is
necessary to install and execute a modified version of the
Combined Work produced by recombining or relinking the
Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)


You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:
a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.2922 apache-log4j 1.2.17

1.2922.1 Available under license:

Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
Open Source Used In DNAC Platform  Antelope  11389

limitations under the License.
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
* 
* http://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

1.2923 depd 1.1.2

1.2923.1 Available under license :
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2924 jju 1.2.1 1.2.1
1.2924.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2925 bluebird 3.5.1
1.2925.1 Available under license:

The MIT License (MIT)

Copyright (c) 2013-2017 Petka Antonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2926 extend-shallow 2.0.1
1.2926.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2927 commons-http-client 3.1
1.2927.1 Available under license:
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and whereon such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or
consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of
such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may
choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or
rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf
and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend,
and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: HOW TO APPLY THE APACHE LICENSE TO YOUR WORK

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by
brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and
description of purpose be included on the same "printed page" as the copyright notice for easier identification within
third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.2928 safer-buffer 2.1.2 2.1.2

1.2928.1 Available under license:
MIT License

Copyright (c) 2018 Nikita Skovoroda <chalkerx@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.2929 lodash-bind 4.2.1

1.2929.1 Available under license :

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.2930 nimbus-jose-jwt 4.41.3

1.2930.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * nimbus-jose-jwt
 *
 * Copyright 2012-2016, Connect2id Ltd.
 * *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use
 * this file except in compliance with the License. You may obtain a copy of the
 * License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed
 * under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-
jar/com/nimbusds/jwt/JWTClaimsSet.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-
jar/com/nimbusds/jose/crypto/AAD.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-
jar/com/nimbusds/jose/PayloadTransformer.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-
Open Source Used In DNAC Platform Antelope 11399

jar/com/nimbusds/jose/util/IntegerUtils.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jwt/proc/BadJWTException.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/LegacyConcatKDF.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/PasswordBasedEncrypter.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/AuthenticatedCipherText.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/JWSObject.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/MACVerifier.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/JOSEProvider.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/proc/BadJWSException.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/ConcatKDF.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/package-info.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/DeflateHelper.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/util/X509CertUtils.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/util/JSONObjectUtils.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/util/RestrictedResourceRetriever.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/bc/package-info.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/CriticalHeaderParamsAware.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/JWEAlgorithm.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/util/ByteUtils.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/proc/SecurityContext.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/jca/package-info.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2.jar/com/nimbusds/jose/crypto/CompositeKey.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-
jar/com/nimbusds/jose/JWSSigner.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/proc/JOSEProcessor.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jwt/JWT.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/AlgorithmFamily.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/util/DefaultResourceRetriever.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/ECDHDecrypter.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/jwk/KeyOperation.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/jwk/source/ImmutableJWKSet.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/jwk/ECPParameterTable.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/factories/DefaultJWEDecrypterFactory.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/RSACryptoProvider.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/KeySourceException.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/CipherHelper.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/AESCBC.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/RSAEncrypter.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jwt/proc/ClockSkewAware.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/jca/JCASupport.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/jwk/JWKMetadata.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/jwk/RemoteKeySourceException.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/jwk/source/JWKSSource.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/jwk/source/JWKSource.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/Header.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/JOSEException.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/BaseJWSProvider.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/JWEDecrypter.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/JWEDecrypter.java

Open Source Used In DNAC Platform Antelope 11401
jar/com/nimbusds/jose/JWSHeader.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jwt/proc/JWTClaimsVerifier.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jwt/proc/JWTParser.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/JWSProvider.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/crypto/RSA1_5.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/crypto/DirectCryptoProvider.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/util/Base64.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/util/ResourceRetriever.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/proc/DefaultJOSEProcessor.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/jwk/JWK.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/jwk/package-info.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/jca/JCAAware.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/crypto/LegacyAESGCM.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/crypto/ECDSA.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/crypto/factories/package-info.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/KeyLengthException.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/crypto/ECDHEncrypter.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/jwk/AssymetricJWK.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jwt/package-info.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/package-info.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/crypto/CriticalHeaderParamsDeferral.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/jwk/SecretJWK.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/jwk/KeyConverter.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/jwk/KeyUse.java
* /opt/cola/permits/1136447733_1613767076.49/jar/com/nimbusds/jose/jwk/package-info.java
jar/com/nimbusds/jose/Algorithm.java
No license file was found, but licenses were detected in source scan.


Found in path(s):
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/META-INF/maven/com.nimbusds/nimbus-jose-jwt/pom.xml
No license file was found, but licenses were detected in source scan.

/*
 * nimbus-jose-jwt
 *
 * Copyright 2012-2016, Connect2id Ltd and contributors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not use
 * this file except in compliance with the License. You may obtain a copy of the
 * License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed
 * under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the
 * specific language governing permissions and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/utils/ConstantTimeUtils.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/AESGCM.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/AESGCMKW.java
* /opt/cola/permits/1136447733_1613767076.49/0/nimbus-jose-jwt-4-41-3-sources-2-jar/com/nimbusds/jose/crypto/utils/package-info.java
1.2931 hbase-server 2.4.2

1.2931.1 Available under license:

Apache HBase - Server
Copyright 2007-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2932 propagate 2.0.1
1.2932.1 Available under license :
   The MIT License

Copyright (c) 2015-2019 Pedro Teixeira and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses. This copy of Jackson JSON processor `jackson-module-jaxb-annotations` module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
depths to use, copy, modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
c Persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2935 raml-parser 0.8.18 0.8.18
1.2935.1 Available under license :

// Copyright Joyent, Inc. and other Node contributors.
//
// Permission is hereby granted, free of charge, to any person obtaining a
// copy of this software and associated documentation files (the
// "Software"), to deal in the Software without restriction, including
// without limitation the rights to use, copy, modify, merge, publish,
// distribute, sublicense, and/or sell copies of the Software, and to permit
// persons to whom the Software is furnished to do so, subject to the
// following conditions:
//
// The above copyright notice and this permission notice shall be included
// in all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
// OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
// MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
// NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
// DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
// OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
// USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2936 jersey-common 2.25.1
1.2936.1 Available under license :

CDDL+GPL License
http://glassfish.java.net/public/CDDL+GPL_1_1.html
1.2937 nconf 0.10.0

1.2937.1 Available under license:

Copyright (C) 2011 Charlie Robbins and the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2938 avahi 0.6.32~rc+dfsg-1ubuntu2.3

1.2938.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.
This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.
GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices
stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no
charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a
   table of data to be supplied by an application program that uses
   the facility, other than as an argument passed when the facility
   is invoked, then you must make a good faith effort to ensure that,
   in the event an application does not supply such function or
   table, the facility still operates, and performs whatever part of
   its purpose remains meaningful.

   (For example, a function in a library to compute square roots has
   a purpose that is entirely well-defined independent of the
   application. Therefore, Subsection 2d requires that any
   application-supplied function or table used by this function must
   be optional: if the application does not supply it, the square
   root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.
3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.
If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least
three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses...
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms
of the ordinary General Public License).

To apply these terms, attach the following notices to the library.
It is safest to attach them to the start of each source file to most
effectively convey the exclusion of warranty; and each file should
have at least the "copyright" line and a pointer to where the full
notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or
your school, if any, to sign a "copyright disclaimer" for the library,
if necessary. Here is a sample; alter the names:
Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.2939 @babel/plugin-transform-literals

7.12.13

1.2939.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2940 jedis 2.10.2

1.2941 node-releases 1.1.71
1.2941.1 Available under license:

The MIT License

Copyright (c) 2017 Sergey Rubanov (https://github.com/chicoxyzzy)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2942 object.assign 4.1.2

1.2942.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2943 ms 2.0.0
1.2943.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2944 javassist 3.18.2-GA
1.2944.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Javassist, a Java-bytecode translator toolkit.
 * Copyright (C) 2004 Bill Burke. All Rights Reserved.
 *
 * The contents of this file are subject to the Mozilla Public License Version
 * 1.1 (the "License"); you may not use this file except in compliance with
 * the License. Alternatively, the contents of this file may be used under
 * the terms of the GNU Lesser General Public License Version 2.1 or later,
 * or the Apache License Version 2.0.
 *
 * Software distributed under the License is distributed on an "AS IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
 * for the specific language governing rights and limitations under the
 * License.
 */

Found in path(s):
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-
jar/javassist/bytecode/annotation/ArrayMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/DoubleMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/EnumMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/MemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/Annotation.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/CharMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/ShortMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/ClassMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/StringMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/BooleanMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/LongMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/IntegerMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/AnnotationMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/ByteMemberValue.java
* /opt/cola/permits/110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5.jar/javassist/bytecode/annotation/FloatMemberValue.java

No license file was found, but licenses were detected in source scan.

/*
 * Javassist, a Java-bytecode translator toolkit.
 * Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.
 *
 * The contents of this file are subject to the Mozilla Public License Version
 * 1.1 (the "License"); you may not use this file except in compliance with
 * the License. Alternatively, the contents of this file may be used under
 * the terms of the GNU Lesser General Public License Version 2.1 or later,
 * or the Apache License Version 2.0.
 *
 * Software distributed under the License is distributed on an "AS IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
 * for the specific language governing rights and limitations under the
 * License.
 */
Open Source Used In DNAC Platform Antelope 11427

Found in path(s):
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/scopedpool/ScopedClassPoolRepository.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/convert/TransformReadField.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/util/proxy/MethodHandler.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/LocalVariableTypeAttribute.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/analysis/Analyzer.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/Bytecode.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/CodeAttribute.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/ClassMap.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/Runtime/DoClass.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/Expr/Cast.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/CtConstructor.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/CtArray.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/Compiler/ast/FieldDecl.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/CtNewClass.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/StackMapTable.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/ast/Keyword.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/convert/TransformCall.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/EnclosingMethodAttribute.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/stackmap/TypeTag.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/JvstCodeGen.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/ast/Pair.java
* /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/ast/InstanceOfExpr.java
Open Source Used In DNAC Platform Antelope 11433

- jar/javassist/compiler/ast/ASTList.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/LoaderClassPath.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/tools/rmi/RemoteException.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/annotation/AnnotationsWriter.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/tools/Dump.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/runtime/Cflow.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/scopedpool/ScopedClassPoolFactory.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/SymbolTable.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/ast/Stmt.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/analysis/Subroutine.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/convert/TransformNewClass.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/ClassFile.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/util/proxy/ProxyObjectOutputStream.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/ast/MethodDecl.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/LineNumberAttribute.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/tools/reflect/Compiler.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/util/proxy/ProxyFactory.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/bytecode/ByteStream.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/MemberResolver.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/ProceedHandler.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/convert/TransformAfter.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/SyntaxError.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/ast/Member.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/compiler/TokenId.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/tools/reflect/Metaobject.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/expr/FieldAccess.java
- * /opt/cola/permits/1110923499_1606873622.06/0/javassist-3-18-2-ga-sources-5-jar/javassist/tools/reflect/Metaobject.java
1.2945 buffer-crc32 0.2.13

1.2945.1 Available under license:

The MIT License

Copyright (c) 2013 Brian J. Brennan

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the
Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE
FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT
OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

1.2946 has-symbols 1.0.1

1.2946.1 Available under license:

MIT License

Copyright (c) 2016 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2947 fs 0.0.2

1.2947.1 Available under license:

# Released under MIT License

Copyright (c) 2021 Cisco.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2948 de.huxhorn.lilith.data.logging 0.9.44

1.2948.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified
version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:
0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

/*
 * Lilith - a log event viewer.
 * Copyright (C) 2007-2011 Joern Huxhorn
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Lesser General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 *
 */

/*
 * Copyright 2007-2011 Joern Huxhorn
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
1.2949 mongodb-driver-core 3.12.4

1.2949.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2005 Brian Goetz and Tim Peierls
 * Released under the Creative Commons Attribution License
 * (http://creativecommons.org/licenses/by/2.5)
 * Official home: http://www.jcip.net
 *
 * Any republication or derived work distributed in source code form
 * must include this copyright and license notice.
 */

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
  jar/com/mongodb/annotations/NotThreadSafe.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
  jar/com/mongodb/annotations/Immutable.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
  jar/com/mongodb/annotations/ThreadSafe.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 * Original Work: MIT License, Copyright (c) [2015-2018] all contributors
 * https://github.com/marianobarrios/tls-channel
 */

Found in path(s):
/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software

No license file was found, but licenses were detected in source scan.
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/event/ConnectionPoolClosedEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/operation/QueryHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/connection/ConnectionPoolSettings.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/connection/ConnectionPoolSettings.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/connection/BufferProvider.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/client/model/geojson/codecs/MultiLineStringCodec.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/internal/connection/ClusterClockAdvancingSessionContext.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/internal/connection/DeleteProtocol.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/event/ServerMonitorListener.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/client/model/Projections.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/event/ConnectionCheckedInEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/client/model/geojson/CoordinateReferenceSystem.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/bulk/BulkWriteUpsert.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/client/model/UpdateManyModel.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/bulk/MongoInterruptedException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/internal/connection/InternalConnection.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/ServerCursor.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/bulk/WriteConcernError.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/operation/AsyncOperationExecutor.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1.jar/com/mongodb/operation/package-info.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/DefaultServerMonitorFactory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/CreateUserOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/internal/connection/UpdateMessage.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/changestream/OperationType.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/UpdateOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/event/ClusterClosedEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb.Driver-core-3-12-4-sources-1-jar/com/mongodb/internal/operation/IndexHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/BsonField.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/event/package-info.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/internal/connection/CommandMessage.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/event/ConnectionClosedEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/IndexOptionDefaults.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/TaggableReadPreference.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/vault/EncryptOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/diagnostics/logging/Logger.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/ReadPreference.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/management/ConnectionPoolStatistics.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/event/ConnectionAddedEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/event/CommandEventMulticaster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/connection/Server.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/WriteConcernResult.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/geojson/Polygon.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/FindAndDeleteOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/BulkWriteOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
Open Source Used In DNAC Platform Antelope 11445
Open Source Used In DNAC Platform Antelope 11446
jar/com/mongodb/internal/connection/DefaultAuthenticator.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/binding/ClusterAwareReadWriteBinding.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/SniSslHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/RenameCollectionOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/geojson/GeometryCollection.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/MongoSocketClosedException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/authentication/NativeAuthenticationHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/UpdateOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ServerOpeningEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/ChangeEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/async/AsyncAggregateResponseBatchCursor.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/diagnostics/logging/Loggers.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/RequestMessage.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/DBRef.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/DeleteMessage.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ConnectionPoolOpenedEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/CompressedMessage.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/DBRefCodec.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/event/ServerMonitorListenerMulticaster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/Authenticator.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/DeleteOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/WriteProtocol.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/operation/AsyncOperations.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/MessageHeader.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
Open Source Used In DNAC Platform Antelope  11450
Open Source Used In DNAC Platform Antelope 11452

jar/com/mongodb/internal/event/EventListenerHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/CreateIndexesOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/ServerAddress.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/CurrentOpOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/geojson/codecs/MultiPolygonCodec.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/changestream/UpdateDescription.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/ExplainHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/MongoSocketWriteException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/gridfs/model/package-info.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ClusterEventMulticaster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/MongoNodeIsRecoveringException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/DefaultDnsSrvRecordMonitorFactory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/result/DeleteResult.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/selector/package-info.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/GetMoreMessage.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/FindOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/ClusterClock.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/session/SessionContext.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/KillCursorProtocol.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/ProtocolExecutor.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/bulk/IndexRequest.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/WriteConcernException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ServerMonitorEventMulticaster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/async/SingleResultCallback.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
-
jar/com/mongodb/internal/session/ClientSessionContext.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/UpdateOneModel.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/CollationStrength.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/internal/connection/SSHHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/BuildersHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/Indexes.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/geojson/Position.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/UnwindOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/changestream/ChangeStreamDocumentCodec.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/connection/ConnectionDescription.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/ValidationOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/ExplainVerbosity.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/TextSearchOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/UpdateUserOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/ErrorCategory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/MongoGridFSException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/connection/ClusterType.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/MapReduceStatistics.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/ListCollectionsOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/AsyncWritableByteChannel.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/geojson/GeoJsonObjectType.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/AbstractByteBufBsonDocument.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/event/ConnectionMessageReceivedEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/binding/AsyncReadWriteBinding.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/MongoServerException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/connection/UnixSocketChannelStream.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/connection/AsyncCompletionHandler.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/session/package-info.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/client/model/DeleteOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/connection/netty/NettyStreamFactory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/MongoIncompatibleDriverException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/connection/SocketStreamFactory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/internal/connection/MultiServerCluster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/internal/connection/MongoCredentialWithCache.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/event/ClusterOpeningEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/operation/OperationExecutor.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/event/ServerHeartbeatSucceededEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/operation/QueryBatchCursor.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/event/ClusterListener.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/operation/DropCollectionOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/MongoCompressor.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/internal/connection/SaslAuthenticator.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/CursorType.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/client/model/FindOneAndReplaceOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/event/CommandStartedEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1.jar/com/mongodb/AuthenticationMechanism.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-core-3-12-4-sources-1-
jar/com/mongodb/internal/operation/Operations.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/MapReduceToCollectionOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ConnectionPoolListenerAdapter.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/CommandResultDocumentCodec.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/ChangeStreamBatchCursor.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/geojson/codecs/GeometryCollectionCodec.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/ReturnDocument.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/ReadConcern.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/connection/ServerVersion.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/connection/SslSettings.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/CommandEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/OpCode.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/diagnostics/logging/package-info.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/InternalConnectionFactory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/BaseCluster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/DBObjectCodec.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ServerClosedEvent.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/BsonWriterHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/connection/ClusterDescription.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/InsertManyOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/DefaultConnectionPool.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/capi/MongoCryptHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/CommandProtocol.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/EstimatedDocumentCountOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/MongoClientSettings.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/binding/AsyncClusterAwareReadWriteBinding.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/FieldTrackingBsonWriter.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/connection/BulkWriteBatchCombiner.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/BucketAutoOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/CompositeByteBuf.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/ServerMonitorFactory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/event/ConnectionPoolListenerMulticaster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/diagnostics/logging/SLF4JLogger.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/KillCursorsMessage.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/NativeAuthenticator.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/WriteModel.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/connection/StreamFactoryFactory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/BasicDBObject.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/AsynchronousSocketChannelStream.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/ParallelCollectionScanOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/FindOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/DnsMultiServerCluster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/DecimalFormatHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ClusterListenerAdapter.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/MongoQueryException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ConnectionPoolEventMulticaster.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-

jar/com/mongodb/client/gridfs/model/GridFSFile.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/connection/ServerConnectionState.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/event/ServerMonitorListenerAdapter.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/binding/ClusterBinding.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/geojson/codecs/NamedCoordinateReferenceSystemCodec.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/ModelException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/CollationAlternate.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/DropIndexOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/MergeOptions.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/connection/netty/NettyBufferProvider.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/bulk/BulkWriteResult.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/connection/package-info.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/operation/ParallelCollectionScanOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/bulk/UpdateRequest.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/MongoClientException.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/ReplyHeader.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/client/model/geojson/CoordinateReferenceSystemType.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/thread/DaemonThreadFactory.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/ResponseCallback.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
jar/com/mongodb/internal/connection/Compressor.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/client/model/AggregationLevel.java
No license file was found, but licenses were detected in source scan.

/___LICENSE__

* Copyright 2008-present MongoDB, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*    http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/internal/connection/ReadConcernAwareNoOpSessionContext.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/internal/operation/ServerVersionHelper.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/CommitTransactionOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/TransactionOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/lang/Nullable.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/lang/NonNullApi.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/lang/package-info.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/operation/AbortTransactionOperation.java
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright 2012 The Netty Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License);
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
  jar/com/mongodb/connection/netty/ReadTimeoutHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License);
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 * Original Work: MIT License, Copyright (c) [2015-2018] all contributors
 * https://github.com/marianobarrios/tls-channel

Open Source Used In DNAC Platform Antelope 11466
/*
**
* Writes a sequence of bytesProduced to this channel from a subsequence of the
* given buffers.
* 
* <p>
* See { @link GatheringByteChannel#write(ByteBuffer[], int, int) } for more
* details of the meaning of this signature.
* 
* This method behaves slightly different than the interface specification,
* with respect to non-blocking responses, see { @link #write(ByteBuffer) }
* for more details.
* 
* @param srcs   The buffers from which bytesProduced are to be retrieved
* @param offset The offset within the buffer array of the first buffer from
*     which bytesProduced are to be retrieved; must be non-negative and no
*     larger than <tt>srcs.length</tt>
* @param length The maximum number of buffers to be accessed; must be
*     non-negative and no larger than <tt>srcs.length</tt>
* &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n

]
which bytesProduced are to be transferred; must be non-negative and no larger than \(<tt>dsts.length</tt>\)
* @param length The maximum number of buffers to be accessed; must be non-negative and no larger than \(<tt>dsts.length</tt>\)
* &nbsp;-&nbsp;\(<tt>offset</tt>\)
* @return The number of bytesProduced read, or \(<tt>-1</tt>\) if the channel has reached end-of-stream; contrary to the behavior specified in \{ @link ByteChannel\}, this method never returns 0, but throws \{ @link WouldBlockException\}
* @throws IndexOutOfBoundsException If the preconditions on the \(<tt>offset</tt>\) and \(<tt>length</tt>\) parameters do not hold
* @throws WouldBlockException if the channel is in non-blocking mode and the IO operation cannot be completed immediately
* @throws NeedsTaskException if the channel is not configured to run tasks automatically and a task needs to be executed to complete the operation
* @throws SSLEngineException if the \{ @link SSLEngine\} throws a SSLEngineException
* @throws IOException if the underlying channel throws an IOException
*/

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/internal/connection/tlschannel/TlsChannel.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2008-present MongoDB, Inc.
* Copyright 2017 Tom Bentley
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-jar/com/mongodb/internal/authentication/SaslPrep.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2008-present MongoDB, Inc.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Written by Doug Lea with assistance from members of JCP JSR-166
Expert Group and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

Returns an array containing all of the elements in this deque, in
proper sequence (from first to last element).

The returned array will be "safe" in that no references to it are
maintained by this deque. (In other words, this method must allocate
a new array). The caller is thus free to modify the returned array.

This method acts as bridge between array-based and collection-based
APIs.

@return an array containing all of the elements in this deque

Copyright 2008-present MongoDB, Inc.
Copyright (c) 2008-2014 Atlassian Pty Ltd

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
  jar/com/mongodb/assertions/Assertions.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2008-present MongoDB, Inc.
* Copyright 2010 The Guava Authors
* Copyright 2011 The Guava Authors
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1136447791_1613767063.08/0/mongodb-driver-core-3-12-4-sources-1-
  jar/com/mongodb/annotations/Beta.java

1.2950 json5 1.0.1 1.0.1

1.2950.1 Available under license :
MIT License

Copyright (c) 2012-2018 Aseem Kishore, and [others].

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[others]: https://github.com/json5/json5/contributors

1.2951 buffer-equal-constant-time 1.0.1
1.2951.1 Available under license :

Copyright (c) 2013, GoInstant Inc., a salesforce.com company
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of salesforce.com, nor GoInstant, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2952 json-schema-compatibility 1.1.0 1.1.0
1.2952.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2953 compression 1.6.2 1.6.2
1.2953.1 Available under license:
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
libjpeg-turbo is covered by three compatible BSD-style open source licenses:

- The IJG (Independent JPEG Group) License, which is listed in [README.ijg](README.ijg)

This license applies to the libjpeg API library and associated programs (any code inherited from libjpeg, and any modifications to that code.)

- The Modified (3-clause) BSD License, which is listed below

This license covers the TurboJPEG API library and associated programs, as well as the build system.

- The zlib License, which is listed below

This license is a subset of the other two, and it covers the libjpeg-turbo SIMD extensions.

Complying with the libjpeg-turbo Licenses
========================================

This section provides a roll-up of the libjpeg-turbo licensing terms, to the best of our understanding.

1. If you are distributing a modified version of the libjpeg-turbo source, then:

   1. You cannot alter or remove any existing copyright or license notices from the source.

      **Origin**
      - Clause 1 of the IJG License
      - Clause 1 of the Modified BSD License
      - Clauses 1 and 3 of the zlib License

   2. You must add your own copyright notice to the header of each source file you modified, so others can tell that you modified that file (if there is not an existing copyright header in that file, then you can simply add a notice stating that you modified the file.)
**Origin**
- Clause 1 of the IJG License
- Clause 2 of the zlib License

3. You must include the IJG README file, and you must not alter any of the copyright or license text in that file.

**Origin**
- Clause 1 of the IJG License

2. If you are distributing only libjpeg-turbo binaries without the source, or if you are distributing an application that statically links with libjpeg-turbo, then:

1. Your product documentation must include a message stating:

   This software is based in part on the work of the Independent JPEG Group.

**Origin**
- Clause 2 of the IJG license

2. If your binary distribution includes or uses the TurboJPEG API, then your product documentation must include the text of the Modified BSD License.

**Origin**
- Clause 2 of the Modified BSD License

3. You cannot use the name of the IJG or The libjpeg-turbo Project or the contributors thereof in advertising, publicity, etc.

**Origin**
- IJG License
- Clause 3 of the Modified BSD License

4. The IJG and The libjpeg-turbo Project do not warrant libjpeg-turbo to be free of defects, nor do we accept any liability for undesirable consequences resulting from your use of the software.

**Origin**
- IJG License
- Modified BSD License
- zlib License

The Modified (3-clause) BSD License
Copyright (C)\<YEAR\> \<AUTHOR\>. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the libjpeg-turbo Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS", AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The zlib License

Copyright (C) \<YEAR\>, \<AUTHOR\>.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Why Three Licenses?
===================

The zlib License could have been used instead of the Modified (3-clause) BSD License, and since the IJG License effectively subsumes the distribution conditions of the zlib License, this would have effectively placed libjpeg-turbo binary distributions under the IJG License. However, the IJG License specifically refers to the Independent JPEG Group and does not extend attribution and endorsement protections to other entities. Thus, it was desirable to choose a license that granted us the same protections for new code that were granted to the IJG for code derived from their software. This file is part of the Independent JPEG Group's software.

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1994-1996, Thomas G. Lane. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:
(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis
of commercial products, provided that all warranty or liability claims are
assumed by the product vendor.
Copyright (c) 2013 Jake Luer <jake@alogicalparadox.com> (http://alogicalparadox.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
/
This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or
distribute this software, either in source code form or as a compiled
binary, for any purpose, commercial or non-commercial, and by any
means.

In jurisdictions that recognize copyright laws, the author or authors
of this software dedicate any and all copyright interest in the
software to the public domain. We make this dedication for the benefit
of the public at large and to the detriment of our heirs and
successors. We intend this dedication to be an overt act of
relinquishment in perpetuity of all present and future rights to this
software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <http://unlicense.org/>
*/
Open Source Used In DNAC Platform Antelope  11478

Copyright (C) 2002, 2003  CodeFactory AB
Copyright (C) 2004, 2005 Red Hat, Inc.

Licensed under the Academic Free License version 2.1

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

---

The Academic Free License
v.2.1

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose
owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original
Work:

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual,
sublicenseable license to do the following:
   a) to reproduce the Original Work in copies;
   b) to prepare derivative works ("Derivative Works") based upon the Original Work;
   c) to distribute copies of the Original Work and Derivative Works to the public;
   d) to perform the Original Work publicly; and
   e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual,
sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original
Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for
making modifications to it and all available documentation describing how to modify the Original Work. Licensor
hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of
the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a
machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive
and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing
the address of that information repository in a notice immediately following the copyright notice that applies to the
Original Work.
4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately proceeding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations
Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved.
Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

---

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

  Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

  You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

     a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this

---

Open Source Used In DNAC Platform Antelope 11482
License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.
NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE
PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE
STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM
"AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTicular PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL
ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR
INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA
BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER
OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to
achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to
most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a
pointer to where the full notice is found.

<one line to give the program's name and an idea of what it does.>
Copyright (C) <yyyy> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public
License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later
version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the
Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO
WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

MIT License

Copyright (C) 2017 by Marijn Haverbeke <marijnh@gmail.com> and others

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2014-2018 Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014 Max Ogden and other contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Paul Hsieh OLD BSD license

Copyright (c) 2010, Paul Hsieh
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither my name, Paul Hsieh, nor the names of any other contributors to the code use may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright (c) 2009 The RE2 Authors. All rights reserved.

// Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

// * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
// * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
The JsonCpp library’s source code, including accompanying documentation, tests and demonstration applications, are licensed under the following conditions:

Baptiste Lepilleur and The JsonCpp Authors explicitly disclaim copyright in all jurisdictions which recognize such a disclaimer. In such jurisdictions, this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of 2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur and The JsonCpp Authors, and is released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this software may choose to accept it either as 1) Public Domain, 2) under the conditions of the MIT License (see below), or 3) under the terms of dual Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is described in clear, concise terms at:

http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

========================================================================
Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXpress OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

========================================================================

(END LICENSE TEXT)

The MIT license is compatible with both the GPL and commercial software, affording one all of the rights of Public Domain with the minor nuisance of being required to keep the above copyright notice and license text in the source code. Note also that by accepting the Public Domain "license" you can re-license your copy using whatever license you like.

Copyright (c) 2015-2018 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

MODIFICATIONS TO THIS FILE MAY MEAN IT NO LONGER ACCURATELY REFLECTS KHRONOS STANDARDS. THE UNMODIFIED, NORMATIVE VERSIONS OF KHRONOS SPECIFICATIONS AND HEADER INFORMATION ARE LOCATED AT https://www.khronos.org/registry/

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

The MIT License (MIT)

Copyright (c) 2014 Bitfighter developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
(The MIT License)

Copyright (c) 2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

GNU GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2015 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt GUI Toolkit under the terms of
GNU General Public License version 2, which is displayed below.

-------------------------------------------------------------------------
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software --to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents.
We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not
responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Program does not specify a version
number of this License, you may choose any version ever published by
the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the
author to ask for permission. For software which is copyrighted by
the Free Software Foundation, write to the Free Software Foundation;
we sometimes make exceptions for this. Our decision will be guided by
the two goals of preserving the free status of all derivatives of our
free software and of promoting the sharing and reuse of software
generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND,
EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH
YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL
NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY
MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE
TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS
The MIT License (MIT)

Copyright (c) 2011 Troy Goode <troygoode@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2012-2013 TJ Holowaychuk
Copyright (c) 2015 Andreas Lubbe
Copyright (c) 2015 Tiancheng "Timothy" Gu

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (C) 2012 Chris Jerdonek. All rights reserved.

Copyright (c) 2009 Chris Wanstrath

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Notice that the following BSD-style license applies to the Valgrind header files used by Chromium (valgrind.h and memcheck.h). However, the rest of Valgrind is licensed under the terms of the GNU General Public License, version 2, unless otherwise indicated.

----------------------------------------------------------------

Copyright (C) 2000-2008 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2012 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2011 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Open Asset Import Library (assimp)

Copyright (c) 2006-2016, assimp team
All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the assimp team, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of the assimp team.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**************************************************************************
AN EXCEPTION applies to all files in the ./test/models-nonbsd folder. These are 3d models for testing purposes, from various free sources on the internet. They are - unless otherwise stated - copyright of their respective creators, which may impose additional requirements on the use of their work. For any of these models, see <model-name>.source.txt for more legal information. Contact us if you have any questions.
are a copyright holder and believe that we credited you improperly or if you don't want your files to appear in the repository.

******************************************************************************

Poly2Tri Copyright (c) 2009-2010, Poly2Tri Contributors
http://code.google.com/p/poly2tri/

All rights reserved.
Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.
* Neither the name of Poly2Tri nor the names of its contributors may be
  used to endorse or promote products derived from this software without specific
  prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd
and Clark Cooper

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must
be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you
7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>
This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

Licence for libxslt except libexslt

-------------------------------------------------------------------------------
Copyright (C) 2001-2002 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

--------------------------------------------------------------------------------

Licence for libexslt

--------------------------------------------------------------------------------

Copyright (C) 2001-2002 Thomas Broyer, Charlie Bozeman and Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the authors shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

--------------------------------------------------------------------------------

(The MIT License)

Copyright (c) 2014 Nathan Rajlich <nathan@tootallnate.net>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
The MIT License (MIT)

Copyright (c) 2014, 2015, 2016, 2017, 2018 Simon Lydell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The MIT License

Copyright (c) 2010 Xtreme Labs and Pivotal Labs

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) Kevin Mrtensson <kevinmartensson@gmail.com> (github.com/kevva)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Most code has one of the following copyrights:

Copyright (C) 2002-2005 3Dlabs Inc. Ltd.
Copyright (C) 2012-2014 LunarG, Inc.
Copyright (C) 2002-2010 The ANGLE Project Authors.
Copyright (C) 2015-2016 Google, Inc.

and is covered under the following license (BSD):

//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//
// Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following
// disclaimer in the documentation and/or other materials provided
// with the distribution.
//
// Neither the name of 3Dlabs Inc. Ltd. nor the names of its
// contributors may be used to endorse or promote products derived
// from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
// FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
// COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
// BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
Files under glslang/MachineIndependent/preprocessor used under the following license (BSD-like):

****************************************************************************
Copyright (c) 2002, NVIDIA Corporation.

NVIDIA Corporation ("NVIDIA") supplies this software to you in consideration of your agreement to the following terms, and your use, installation, modification or redistribution of this NVIDIA software constitutes acceptance of these terms. If you do not agree with these terms, please do not use, install, modify or redistribute this NVIDIA software.

In consideration of your agreement to abide by the following terms, and subject to these terms, NVIDIA grants you a personal, non-exclusive license, under NVIDIA's copyrights in this original NVIDIA software (the "NVIDIA Software"), to use, reproduce, modify and redistribute the NVIDIA Software, with or without modifications, in source and/or binary forms; provided that if you redistribute the NVIDIA Software, you must retain the copyright notice of NVIDIA, this notice and the following text and disclaimers in all such redistributions of the NVIDIA Software. Neither the name, trademarks, service marks nor logos of NVIDIA Corporation may be used to endorse or promote products derived from the NVIDIA Software without specific prior written permission from NVIDIA.

Except as expressly stated in this notice, no other rights or licenses express or implied, are granted by NVIDIA herein, including but not limited to any patent rights that may be infringed by your derivative works or by other works in which the NVIDIA Software may be incorporated. No hardware is licensed hereunder.

THE NVIDIA SOFTWARE IS BEING PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR ITS USE AND OPERATION EITHER ALONE OR IN COMBINATION WITH OTHER PRODUCTS.

IN NO EVENT SHALL NVIDIA BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, EXEMPLARY, CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
TO, LOST PROFITS; PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) OR ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION OF THE NVIDIA SOFTWARE, HOWEVER CAUSED AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

****************************************************************************/

-----

The file glslang/MachineIndependent/gl_types.h has the following license (MIT):

/*
** Copyright (c) 2013 The Khronos Group Inc.
**
** Permission is hereby granted, free of charge, to any person obtaining a copy
** of this software and/or associated documentation files (the
** "Materials"), to deal in the Materials without restriction, including
** without limitation the rights to use, copy, modify, merge, publish,
** distribute, sublicense, and/or sell copies of the Materials, and to
** permit persons to whom the Materials are furnished to do so, subject to
** the following conditions:
**
** The above copyright notice and this permission notice shall be included
** in all copies or substantial portions of the Materials.
**
** THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
** EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
** MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
** IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
** CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
** TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
** MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.
*/

-----

The files glslang/SPIRV/GLSL.std.450.h and

    glslang/SPIRV/spirv.hpp have the following license.

/*
** Copyright (c) 2014-2016 The Khronos Group Inc.
**
** Permission is hereby granted, free of charge, to any person obtaining a copy
** of this software and/or associated documentation files (the "Materials"),
** to deal in the Materials without restriction, including without limitation
** the rights to use, copy, modify, merge, publish, distribute, sublicense,
** and/or sell copies of the Materials, and to permit persons to whom the
** Materials are furnished to do so, subject to the following conditions:
**The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.**

**MODIFICATIONS TO THIS FILE MAY MEAN IT NO LONGER ACCURATELY REFLECTS KHRONOS STANDARDS. THE UNMODIFIED, NORMATIVE VERSIONS OF KHRONOS SPECIFICATIONS AND HEADER INFORMATION ARE LOCATED AT https://www.khronos.org/registry/**

**THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.**

*/

-----

Files under tools/data contain portions of GNU Bison, and are used under the following license (GPL v3), with an exception as described at http://www.gnu.org/software/bison/manual/bison.html#Conditions. In particular, when parsers are generated from the template files, an exception clause is inserted into the generated source files which allows their use in non-free programs.


# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.

Copyright (c) 2010-2017 Benjamin Peterson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2013-2017 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contributions to the collaboration shall not be considered confidential.

Each contributor represents and warrants that it has the right and authority to license copyright in its contributions to the collaboration.

Each contributor agrees to license the copyright in the contributions under the Modified (2-clause or 3-clause) BSD License or the Clear BSD License.

Please see the IPR statements submitted to the IETF for the complete patent licensing details:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Microsoft Corporation:
https://datatracker.ietf.org/ipr/1914/

Skype Limited:
https://datatracker.ietf.org/ipr/1602/
Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/
The MIT License (MIT)

Copyright (c) 2016 crypto-browserify

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# https://developers.google.com/open-source/cla/individual
# https://developers.google.com/open-source/cla/corporate
#
# Names should be added to this file as:
# Name <email address>

Lei Zhang <antiagainst@google.com>
David Neto <dneto@google.com>
Andrew Woloszyn <awoloszyn@google.com>
Stefanus Du Toit <sdt@google.com>
Dejan Mircevski <deki@google.com>
Mark Adams <marka@nvidia.com>
Jason Ekstrand <jason.ekstrand@intel.com>
Damien Mabin <dmabin@google.com>
Qining Lu <qining@google.com>
Jakob Vogel <JakobpunktVogel@gmail.com>
David Yen <dyen@google.com>
Adam Chainz <adam@adamj.eu>
Copyright (c) 2010, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software is released under the MIT license:

Copyright (c) James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
The MIT License (MIT)

Copyright (c) 2015-2017, Brian Woodward.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2015 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2006-2013 James Graham and other contributors
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: perfetto
Source: https://android.googlesource.com/platform/external/perfetto/

Files: *
Copyright: Copyright (C) 2017 The Android Open Source Project
License: Apache 2
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*
 * SpanDSP - a series of DSP components for telephony
 *
 * g722_decode.c - The ITU G.722 codec, decode part.
 *
 * Written by Steve Underwood <steveu@coppice.org>
 *
 * Copyright (C) 2005 Steve Underwood
 *
 * Despite my general liking of the GPL, I place my own contributions
 * to this code in the public domain for the benefit of all mankind -
 * even the slimy ones who might try to proprietize my work and use it
* to my detriment.
* Based in part on a single channel G.722 codec which is:
* Copyright (c) CMU 1993
* Computer Science, Speech Group
* Chengxiang Lu and Alex Hauptmann
*/

Copyright 2015 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2000 Hans Petter Bieker. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2000 by Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2009, 2010, 2013-2016 by the Brotli Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
PCRE2 LICENCE
-------------

PCRE2 is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language.

Releases 10.00 and above of PCRE2 are distributed under the terms of the "BSD" licence, as specified below, with one exemption for certain binary redistributions. The documentation for PCRE2, supplied in the "doc" directory, is distributed under the same terms as the software itself. The data in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also included in the distribution is a just-in-time compiler that can be used to optimize pattern matching. This is an optional feature that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS
-----------------------------

Written by: Philip Hazel
Email local part: ph10
Email domain: cam.ac.uk

University of Cambridge Computing Service,

Copyright (c) 1997-2019 University of Cambridge
All rights reserved.

PCRE2 JUST-IN-TIME COMPILATION SUPPORT
---------------------------------------

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

Copyright(c) 2010-2019 Zoltan Herczeg
All rights reserved.

STACK-LESS JUST-IN-TIME COMPILER
----------------------------------

Written by: Zoltan Herczeg
Email local part: hzmester
THE "BSD" LICENCE
-------------

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notices,
  this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notices, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the names of any
  contributors may be used to endorse or promote products derived from this
  software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

EXEMPTION FOR BINARY LIBRARY-LIKE PACKAGES
--------------------------------------------

The second condition in the BSD licence (covering binary redistributions) does
not apply all the way down a chain of software. If binary package A includes
PCRE2, it must respect the condition, but if package B is software that
includes package A, the condition is not imposed on package B unless it uses
PCRE2 independently.

End
The MIT License (MIT)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2012, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

   "This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana Univeristy" and "Indiana Univeristy Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact http://www.extreme.indiana.edu/.

5. Products derived from this software may not use "Indiana Univeristy" name nor may "Indiana Univeristy" appear in their name, without prior written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

gestures/LICENSE

// Copyright (c) 2011 The Chromium OS Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
//   notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
//   copyright notice, this list of conditions and the following disclaimer
//   in the documentation and/or other materials provided with the
//   distribution.
// * Neither the name of Google Inc. nor the names of its
//   contributors may be used to endorse or promote products derived from
//   this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2014 Kim Kulling

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2011-2015 Paul Vorbach <paul@vorba.ch>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2016 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt Toolkit under the terms of GNU Lesser General Public License version 3, which is displayed below. This license makes reference to the version 3 of the GNU General Public License, which you can find in the LICENSE.GPL3 file.
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this
licensedocument, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, this License refers to version 3 of the GNU Lesser
General Public License, and the GNU GPL refers to version 3 of the
GNU General Public License.

The Library refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An Application is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A Combined Work is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the Linked
Version.

The Minimal Corresponding Source for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The Corresponding Application Code for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.
2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License or any later version applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy’s public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

(MIT License)

Copyright (c) 2013 [Ramesh Nair](http://www.hiddentao.com/)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CMake - Cross Platform Makefile Generator
Copyright 2000-2009 Kitware, Inc., Insight Software Consortium
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the names of Kitware, Inc., the Insight Software Consortium, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The above copyright and license notice applies to distributions of CMake in source and binary form. Some source files contain additional notices of original copyright by their contributors; see each source for details. Third-party software packages supplied with CMake under compatible licenses provide their own copyright notices documented in corresponding subdirectories.

CMake was initially developed by Kitware with the following sponsorship:

* National Library of Medicine at the National Institutes of Health as part of the Insight Segmentation and Registration Toolkit (ITK).

* US National Labs (Los Alamos, Livermore, Sandia) ASC Parallel Visualization Initiative.
* National Alliance for Medical Image Computing (NAMIC) is funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149.

* Kitware, Inc.
Copyright 2012-2015 The Dojo Foundation <http://dojofoundation.org/>
Based on Underscore.js, copyright 2009-2015 Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/*
* Copyright 2011 The WebRTC Project Authors. All rights reserved.
* 
* Use of this source code is governed by a BSD-style license
* that can be found in the LICENSE file in the root of the source
* tree. An additional intellectual property rights grant can be found
* in the file PATENTS. All contributing project authors may
* be found in the AUTHORS file in the root of the source tree.
*/

Copyright 2012 Square Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

Copyright 2002 USC/Information Sciences Institute

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Information Sciences Institute not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Information Sciences Institute makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

INFORMATION SCIENCES INSTITUTE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL INFORMATION SCIENCES INSTITUTE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (C) 1995-2002 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com
Copyright (c) 2016 Douglas Christopher Wilson doug@somethingdoug.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source“ form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Copyright 2017 Kat Marchn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright 2007, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-------------------------------------------------------------------------------

The file url_parse.cc is based on nsURLParsers.cc from Mozilla. This file is licensed separately as follows:

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is mozilla.org code.

The Initial Developer of the Original Code is Netscape Communications Corporation.
Portions created by the Initial Developer are Copyright (C) 1998 the Initial Developer. All Rights Reserved.

Contributor(s):
Darin Fisher (original author)

Alternatively, the contents of this file may be used under the terms of
either the GNU General Public License Version 2 or later (the “GPL”), or
the GNU Lesser General Public License Version 2.1 or later (the "LGPL"),
in which case the provisions of the GPL or the LGPL are applicable instead
of those above. If you wish to allow use of your version of this file only
under the terms of either the GPL or the LGPL, and not to allow others to
use your version of this file under the terms of the MPL, indicate your
decision by deleting the provisions above and replace them with the notice
and other provisions required by the GPL or the LGPL. If you do not delete
the provisions above, a recipient may use your version of this file under
the terms of any one of the MPL, the GPL or the LGPL.
Copyright (c) 2015, Wes Todd

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) Microsoft Corporation. All rights reserved.
Licensed under the Apache License, Version 2.0 (the "License"); you may not use
this file except in compliance with the License. You may obtain a copy of the
License at http://www.apache.org/licenses/LICENSE-2.0

THIS CODE IS PROVIDED ON AN *AS IS* BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED
WARRANTIES OR CONDITIONS OF TITLE, FITNESS FOR A PARTICULAR PURPOSE,
MERCHANTABILITY OR NON-INFRINGEMENT.

See the Apache Version 2.0 License for specific language governing permissions
and limitations under the License.

HarfBuzz was previously licensed under different licenses. This was
changed in January 2008. If you need to relicense your old copies,
consult the announcement of the license change on the internet.
Other than that, each copy of HarfBuzz is licensed under the COPYING
file included with it. The actual license follows:

Permission is hereby granted, without written agreement and without
license or royalty fees, to use, copy, modify, and distribute this
software and its documentation for any purpose, provided that the
above copyright notice and the following two paragraphs appear in
all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES
ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN
IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS
ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
Espree
Copyright JS Foundation and other contributors, https://js.foundation

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2015 Javier Blanco

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
Copyright (c) 2014 Andrew Kelley

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2011 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2006 Bob Ippolito

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Kevin Gravier

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013, Jim Higson
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (Expat)

Copyright (c) 2014 Andrew Kelley

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge,
publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-2016 Cory Benfield and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014, Eugene Ware

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of Eugene Ware nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY EUGENE WARE "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL EUGENE WARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Parts of jpeg-js-decoder.js is also licensed under Apache-2.0 license. See source file for license.

This license applies to all parts of V8 that are not externally maintained libraries. The externally maintained libraries used by V8 are:

- PCRE test suite, located in test/mjsunit/third_party/regexp-pcre/regexp-pcre.js. This is based on the test suite from PCRE-7.3, which is copyrighted by the University of Cambridge and Google, Inc. The copyright notice and license are embedded in regexp-pcre.js.

- Layout tests, located in test/mjsunit/third_party/object-keys. These are based on layout tests from webkit.org which are copyrighted by Apple Computer, Inc. and released under a 3-clause BSD license.

- Strongtalk assembler, the basis of the files assembler-arm-inl.h, assembler-arm.cc, assembler-arm.h, assembler-ia32-inl.h, assembler-ia32.cc, assembler-ia32.h, assembler-x64-inl.h, assembler-x64.cc, assembler-x64.h, assembler-mips-inl.h, assembler-mips.cc, assembler-mips.h, assembler.cc and assembler.h. This code is copyrighted by Sun Microsystems Inc. and released under a 3-clause BSD license.

- Valgrind client API header, located at src/third_party/valgrind/valgrind.h This is released under the BSD license.

- The Wasm C/C++ API headers, located at third_party/wasm-api/wasm.{h,hh} This is released under the Apache license. The API's upstream prototype implementation also formed the basis of V8's implementation in src/wasm/c-api.cc.
These libraries have their own licenses; we recommend you read them, as their terms may differ from the terms below.

Further license information can be found in LICENSE files located in sub-directories.

Copyright 2014, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

HarfBuzz is licensed under the so-called "Old MIT" license. Details follow. For parts of HarfBuzz that are licensed under different licenses see individual files names COPYING in subdirectories where applicable.

Copyright 2019 Facebook, Inc.
Copyright 2012 Mozilla Foundation
Copyright 2011 Codethink Limited
Copyright 2008,2010 Nokia Corporation and/or its subsidiary(-ies)
Copyright 2009 Keith Stribley
Copyright 2009 Martin Hosken and SIL International
Copyright 2007 Chris Wilson
Copyright 2006 Behdad Esfahbod
Copyright 2005 David Turner
Copyright 1998-2004 David Turner and Werner Lemberg

For full copyright notices consult the individual files in the package.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS. (The MIT License)

Copyright (C) 2012 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

qcms
Copyright (C) 2009 Mozilla Corporation
Copyright (C) 1998-2007 Marti Maria

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software
is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2017 yhirose

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Public domain
Copyright (C) 2001-2002 by Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*** Portions of the driver (that is, bdflib.c and bdf.h):

Copyright 2000 Computing Research Labs, New Mexico State University
Copyright 2001-2002, 2011 Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.
Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying

Open Source Used In DNAC Platform Antelope 11560
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs
If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year  name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Copyright 2015 Calvin Metcalf

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

# This is the official list of people who can contribute
# (and typically have contributed) code to the Wuffs repository.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# The submission process automatically checks to make sure
# that people submitting code are listed in this file (by email address).
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
# http://code.google.com/legal/individual-cla-v1.0.html
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate CLA was used.
#
# Names should be added to this file like so:
# Individual's name <submission email address>
# Individual's name <submission email address> <email2> <emailN>
#
# An entry with multiple email addresses specifies that the
# first address should be used in the submit logs and
# that the other addresses should be recognized as the
# same person when interacting with Gerrit.

# Please keep the list sorted.

Andrew Dassonville <dassonville.andrew@gmail.com>
Jimmy Casey <casey.jimmy@gmail.com>
Nigel Tao <nigeltao@golang.org>

Copyright (c) 2013 Christopher Thunes

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

MIT License

Copyright (c) 2014-2018 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Zeit, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2014 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# Contributors to the OpenH264 project

Patrick Ai
Sijia Chen
ZhaoZheng Chu
Paley Du
Martin Ettl
Andreas Gal
Xu Guang
Licai Guo
Yi Guo
Horace Huang
Steven Huang
Ethan Hugg
Cullen Jennings
ZhaoFeng Jia
Derrick Jin
Jesse Li
Jifei Li
Kai Li
Karina Li
Matt Li
Xiang Li
Bourne Ling
Alex Liu
Wayne Liu
Varun Patil
Eric Rescorla
Adam Roach
Sawyer Shan
Siping Tao
Martin Storsj
Brion Vibber
James Wang
Juanmy Wang
Zhiliang Wang
Herv Willems
Gregory J Wolfe
Katherine Wu
Guang Xu
Jeffery Xu
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* MODP_B64 - High performance base64 encoder/decoder
* Version 1.3 -- 17-Mar-2006
* http://modp.com/release/base64
* 
* Copyright (c) 2005, 2006 Nick Galbreath -- nickg [at] modp [dot] com
* All rights reserved.
* 
* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* 
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* 
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of the modp.com nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The project in general is under the following licenses:

================================================================================
Copyright 2010 Google Inc.
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

================================================================================
File dict_phrases.dat is built from libTabe; the licenses of libTabe is:

================================================================================
Copyright (c) 1999 TaBE Project.
Copyright (c) 1999 Pai-Hsiang Hsiao.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the TaBE Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1999 Computer Systems and Communication Lab, Institute of Information Science, Academia Sinica.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Computer Systems and Communication Lab nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 1996 Chih-Hao Tsai @ Beckman Institute, University of Illinois
c-tsai4@uiuc.edu http://casper.beckman.uiuc.edu/~c-tsai4
==========================================================================
Copyright (c) 2003-2006, Joe Walnes
Copyright (c) 2006-2015 XStream Committers
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of
   conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of
   conditions and the following disclaimer in the documentation and/or other materials provided
   with the distribution.

3. Neither the name of XStream nor the names of its contributors may be used to endorse
   or promote products derived from this software without specific prior written
   permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
Copyright (c) 2016, the R8 project authors.
All rights reserved.

Redistribution and use in source and binary forms, with or without
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Libevent is available for use under the following license, commonly known as the 3-clause (or "modified") BSD license:

============================================
Copyright (c) 2000-2007 Niels Provos <provos@citi.umich.edu>
Copyright (c) 2007-2010 Niels Provos and Nick Mathewson

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
Portions of Libevent are based on works by others, also made available by them under the three-clause BSD license above. The copyright notices are available in the corresponding source files; the license is as above. Here's a list:

log.c:
   Copyright (c) 2000 Dug Song <dugsong@monkey.org>
   Copyright (c) 1993 The Regents of the University of California.

strlcpy.c:
   Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

win32.c:
   Copyright (c) 2003 Michael A. Davis <mike@datanerds.net>

evport.c:
   Copyright (c) 2007 Sun Microsystems

min_heap.h:
   Copyright (c) 2006 Maxim Yegorushkin <maxim.yegorushkin@gmail.com>

tree.h:
   Copyright 2002 Niels Provos <provos@citi.umich.edu>
The MIT License (MIT)

Copyright (c) 2014 Daniel Cousens

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
xXHash Library
Copyright (c) 2012-2014, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper
Copyright (c) 2001-2017 Expat maintainers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Adobe Systems Incorporated nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the \"Software\"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED \"AS IS\", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Blaine Bublitz <blaine.bublitz@gmail.com>, Eric Schoffstall <yo@contra.io> and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

AutobahnAndroid
Copyright 2011,2012 Tavendo GmbH. Licensed under Apache 2.0
This product includes software developed at Tavendo GmbH http://www.tavendo.de
Copyright (c) 2012 Simon Boudrias

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Tencent is pleased to support the open source community by making RapidJSON available.

Copyright (C) 2015 THL A29 Limited, a Tencent company, and Milo Yip. All rights reserved.

If you have downloaded a copy of the RapidJSON binary from Tencent, please note that the RapidJSON binary is licensed under the MIT License.
If you have downloaded a copy of the RapidJSON source code from Tencent, please note that RapidJSON source
code is licensed under the MIT License, except for the third-party components listed below which are subject to different license terms. Your integration of RapidJSON into your own projects may require compliance with the MIT License, as well as the other licenses applicable to the third-party components included within RapidJSON. To avoid the problematic JSON license in your own projects, it's sufficient to exclude the bin/jsonchecker/ directory, as it's the only code under the JSON license.

A copy of the MIT License is included in this file.

Other dependencies and licenses:

Open Source Software Licensed Under the BSD License:
--------------------------------------------------------------------

The msinttypes r29
Copyright (c) 2006-2013 Alexander Chemeris
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Open Source Software Licensed Under the JSON License:
--------------------------------------------------------------------

json.org
Copyright (c) 2002 JSON.org
All Rights Reserved.

JSON_checker
Copyright (c) 2002 JSON.org
All Rights Reserved.
Terms of the JSON License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Terms of the MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2015 Devon Govett <devongovett@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This project contains parts of Node.js.
Node.js is licensed for use as follows:

===
Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===

This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

===
Copyright Joyent, Inc. and other Node contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===

The Ed25519 software is in the public domain.
Copyright (c) 2011 Alexander Shuchkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2009, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Software License Agreement (BSD License)

Copyright (c) 2009-2015, Kevin Decker <kpdecker@gmail.com>

All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Kevin Decker nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Javolution - Java(TM) Solution for Real-Time and Embedded Systems
Copyright (c) 2006, Javolution (http://javolution.org)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2017 crypto-browserify contributors
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

W3C 3-clause BSD License

http://www.w3.org/Consortium/Legal/2008/03-bsd-license.html

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of works must retain the original copyright notice,
  this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the original copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

* Neither the name of the W3C nor the names of its contributors may be
  used to endorse or promote products derived from this work without
  specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
(The MIT License)

Copyright (c) 2011-2015 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2011 Google Inc. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

pyelftools is in the public domain (see below if you need more details).

pyelftools uses the construct library for structured parsing of a binary stream. construct is packaged in pyelftools/construct - see its LICENSE file for the license.

------------------------------------------------------------------------

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors
of this software dedicate any and all copyright interest in the
software to the public domain. We make this dedication for the benefit
of the public at large and to the detriment of our heirs and
successors. We intend this dedication to be an overt act of
relinquishment in perpetuity of all present and future rights to this
software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <http://unlicense.org/>
GNU Free Documentation License
Version 1.3, 3 November 2008

<http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of "copyleft", which means that derivative
works of the document must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.
1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input.
to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History"). To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no
other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.
4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements.
and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retile any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and
multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole
aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently
reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.
The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.
Copyright (c) 2010-2014 Caolan McMahon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
3.1.0 / 2017-09-26
==================

* Add `DEBUG_HIDE_DATE` env var (#486)
* Remove ReDoS regexp in `%o` formatter (#504)
* Remove `"component"` from package.json
* Remove `"component.json"`
* Ignore package-lock.json
* Examples: fix colors printout
* Fix: browser detection
* Fix: spelling mistake (#496, @EdwardBetts)

3.0.1 / 2017-08-24
==================

* Fix: Disable colors in Edge and Internet Explorer (#489)

3.0.0 / 2017-08-08
==================

* Breaking: Remove DEBUG_FD (#406)
* Breaking: Use `Date#toISOString()` instead of `Date#toUTCString()` when output is not a TTY (#418)
* Breaking: Make millisecond timer namespace specific and allow 'always enabled' output (#408)
* Addition: document 'enabled' flag (#465)
* Addition: add 256 colors mode (#481)
* Addition: `enabled()`, updates existing debug instances, add `destroy()` function (#440)
* Update: component: update "ms" to v2.0.0
* Update: separate the Node and Browser tests in Travis-CI
* Update: refactor Readme, fixed documentation, added "Namespace Colors" section, redid screenshots
* Update: separate Node.js and web browser examples for organization
* Update: update "browserify" to v14.4.0
* Fix: fix Readme typo (#473)

2.6.9 / 2017-09-22
==================

* remove ReDoS regexp in `%o` formatter (#504)

2.6.8 / 2017-05-18
==================

* Fix: Check for undefined on browser globals (#462, @marbemac)
2.6.7 / 2017-05-16
==================
- Fix: Update ms to 2.0.0 to fix regular expression denial of service vulnerability (#458, @hubdotcom)
- Fix: Inline extend function in node implementation (#452, @dougwilson)
- Docs: Fix typo (#455, @msasad)

2.6.5 / 2017-04-27
==================
- Fix: null reference check on window.documentElement.style.WebkitAppearance (#447, @thebigredgeek)
- Misc: clean up browser reference checks (#447, @thebigredgeek)
- Misc: add npm-debug.log to .gitignore (@thebigredgeek)

2.6.4 / 2017-04-20
==================
- Fix: bug that would occur if process.env.DEBUG is a non-string value. (#444, @LucianBuzzo)
- Chore: ignore bower.json in npm installations. (#437, @joaovieira)
- Misc: update "ms" to v0.7.3 (@tootallnate)

2.6.3 / 2017-03-13
==================
- Fix: Electron reference to `process.env.DEBUG` (#431, @paulcbetts)
- Docs: Changelog fix (@thebigredgeek)

2.6.2 / 2017-03-10
==================
- Fix: DEBUG_MAX_ARRAY_LENGTH (#420, @slavaGanzin)
- Docs: Add backers and sponsors from Open Collective (#422, @piamancini)
- Docs: Add Slackin invite badge (@tootallnate)

2.6.1 / 2017-02-10
==================
- Fix: Module’s `export default` syntax fix for IE8 `Expected identifier` error
- Fix: Whitelist DEBUG_FD for values 1 and 2 only (#415, @pi0)
- Fix: IE8 "Expected identifier" error (#414, @vgoma)
- Fix: Namespaces would not disable once enabled (#409, @musikov)

2.6.0 / 2016-12-28
==================
- Fix: added better null pointer checks for browser useColors (@thebigredgeek)
* Improvement: removed explicit `window.debug` export (#404, @tootallnate)
* Improvement: deprecated `DEBUG_FD` environment variable (#405, @tootallnate)

2.5.2 / 2016-12-25
==================

* Fix: reference error on window within webworkers (#393, @KlausTrainer)
* Docs: fixed README typo (#391, @lurch)
* Docs: added notice about v3 api discussion (@thebigredgeek)

2.5.1 / 2016-12-20
==================

* Fix: babel-core compatibility

2.5.0 / 2016-12-20
==================

* Fix: wrong reference in bower file (@thebigredgeek)
* Fix: webworker compatibility (@thebigredgeek)
* Fix: output formatting issue (#388, @kribblo)
* Fix: babel-loader compatibility (#383, @escwald)
* Misc: removed built asset from repo and publications (@thebigredgeek)
* Misc: moved source files to /src (#378, @yamikuronue)
* Test: added karma integration and replaced babel with browserify for browser tests (#378, @yamikuronue)
* Test: coveralls integration (#378, @yamikuronue)
* Docs: simplified language in the opening paragraph (#373, @yamikuronue)

2.4.5 / 2016-12-17
==================

* Fix: `navigator` undefined in Rhino (#376, @jochenberger)
* Fix: custom log function (#379, @hsiliev)
* Improvement: bit of cleanup + linting fixes (@thebigredgeek)
* Improvement: rm non-maintained `dist/` dir (#375, @freewil)
* Docs: simplified language in the opening paragraph. (#373, @yamikuronue)

2.4.4 / 2016-12-14
==================

* Fix: work around debug being loaded in preload scripts for electron (#368, @paulcbetts)

2.4.3 / 2016-12-14
==================

* Fix: navigation.userAgent error for react native (#364, @escwald)

2.4.2 / 2016-12-14
* Fix: browser colors (#367, @tootallnate)
* Misc: travis ci integration (@thebigredgeek)
* Misc: added linting and testing boilerplate with sanity check (@thebigredgeek)

2.4.1 / 2016-12-13

* Fix: typo that broke the package (#356)

2.4.0 / 2016-12-13

* Fix: bower.json references unbuilt src entry point (#342, @justmatt)
* Fix: revert "handle regex special characters" (@tootallnate)
* Feature: configurable util.inspect() options for NodeJS (#327, @tootallnate)
* Feature: %O (big O) pretty-prints objects (#322, @tootallnate)
* Improvement: allow colors in workers (#335, @botverse)
* Improvement: use same color for same namespace. (#338, @lchenay)

2.3.3 / 2016-11-09

* Fix: Catch `JSON.stringify()` errors (#195, Jovan Alleyne)
* Fix: Returning `localStorage` saved values (#331, Levi Thomason)
* Improvement: Don't create an empty object when no `process` (Nathan Rajlich)

2.3.2 / 2016-11-09

* Fix: be super-safe in index.js as well (@TooTallNate)
* Fix: should check whether process exists (Tom Newby)

2.3.1 / 2016-11-09

* Fix: Added electron compatibility (#324, @paulcbetts)
* Improvement: Added performance optimizations (@tootallnate)
* Readme: Corrected PowerShell environment variable example (#252, @gimre)
* Misc: Removed yarn lock file from source control (#321, @fengmk2)

2.3.0 / 2016-11-07

* Fix: Consistent placement of ms diff at end of output (#215, @gorangajic)
* Fix: Escaping of regex special characters in namespace strings (#250, @zacronos)
* Fix: Fixed bug causing crash on react-native (#282, @vkarpov15)
* Feature: Enabled ES6+ compatible import via default export (#212 @bucaran)
* Feature: Added %O formatter to reflect Chrome's console.log capability (#279, @oncletom)
* Package: Update "ms" to 0.7.2 (#315, @DevSide)
* Package: removed superfluous version property from bower.json (#207 @kkirsche)
* Readme: fix USE_COLORS to DEBUG COLORS
* Readme: Doc fixes for format string sugar (#269, @mlucool)
* Readme: Updated docs for DEBUG_FD and DEBUG_COLORS environment variables (#232, @mattlyons0)
* Readme: doc fixes for PowerShell (#271 #243, @exoticknight @unreadable)
* Readme: better docs for browser support (#224, @matthewmueller)
* Tooling: Added yarn integration for development (#317, @thebigredgeek)
* Misc: Renamed History.md to CHANGELOG.md (@thebigredgeek)
* Misc: Added license file (#226 #274, @CantemoInternal @sdaitzman)
* Misc: Updated contributors (@thebigredgeek)

2.2.0 / 2015-05-09

* package: update "ms" to v0.7.1 (#202, @dougwilson)
* README: add logging to file example (#193, @DanielOchoa)
* README: fixed a typo (#191, @amir-s)
* browser: expose `storage` (#190, @stephenmathieson)
* Makefile: add a `distclean` target (#189, @stephenmathieson)

2.1.3 / 2015-03-13

* Updated stdout/stderr example (#186)
* Updated example/stdout.js to match debug current behaviour
* Renamed example/stderr.js to stdout.js
* Update Readme.md (#184)
* replace high intensity foreground color for bold (#182, #183)

2.1.2 / 2015-03-01

* dist: recompile
* update "ms" to v0.7.0
* package: update "browserify" to v9.0.3
* component: fix "ms.js" repo location
* changed bower package name
* updated documentation about using debug in a browser
* fix: security error on safari (#167, #168, @yields)

2.1.1 / 2014-12-29

* browser: use `typeof` to check for `console` existence
* browser: check for `console.log` truthiness (fix IE 8/9)
* browser: add support for Chrome apps
* Readme: added Windows usage remarks
* Add `bower.json` to properly support bower install

2.1.0 / 2014-10-15
==================

* node: implement `DEBUG_FD` env variable support
* package: update "browserify" to v6.1.0
* package: add "license" field to package.json (#135, @panuhorsmalait)

2.0.0 / 2014-09-01
==================

* package: update "browserify" to v5.11.0
* node: use stderr rather than stdout for logging (#29, @stephenmathieson)

1.0.4 / 2014-07-15
==================

* dist: recompile
* example: remove `console.info()` log usage
* example: add "Content-Type" UTF-8 header to browser example
* browser: place `%c` marker after the space character
* browser: reset the "content" color via `%color: inherit`
* browser: add colors support for Firefox >= v31
* debug: prefer an instance `log()` function over the global one (#119)
* Readme: update documentation about styled console logs for FF v31 (#116, @wryk)

1.0.3 / 2014-07-09
==================

* Add support for multiple wildcards in namespaces (#122, @seegno)
* browser: fix lint

1.0.2 / 2014-06-10
==================

* browser: update color palette (#113, @gscottolson)
* common: make console logging function configurable (#108, @timoxley)
* node: fix %o colors on old node <= 0.8.x
* Makefile: find node path using shell/which (#109, @timoxley)

1.0.1 / 2014-06-06
==================

* browser: use `removeItem()` to clear localStorage
* browser, node: don't set DEBUG if namespaces is undefined (#107, @leedm777)
* package: add "contributors" section
* node: fix comment typo
* README: list authors

1.0.0 / 2014-06-04
==================

* make ms diff be global, not be scope
* debug: ignore empty strings in enable()
* node: make DEBUG_COLORS able to disable coloring
* *: export the `colors` array
* npmignore: don't publish the `dist` dir
* Makefile: refactor to use browserify
* package: add "browserify" as a dev dependency
* Readme: add Web Inspector Colors section
* node: reset terminal color for the debug content
* node: map "%o" to `util.inspect()`
* browser: map "%j" to `JSON.stringify()`
* debug: add custom "formatters"
* debug: use "ms" module for humanizing the diff
* Readme: add "bash" syntax highlighting
* browser: add Firebug color support
* browser: add colors for WebKit browsers
* node: apply log to `console`
* rewrite: abstract common logic for Node & browsers
* add .jshintrc file

0.8.1 / 2014-04-14
==================

* package: re-add the "component" section

0.8.0 / 2014-03-30
==================

* add `enable()` method for nodejs. Closes #27
* change from stderr to stdout
* remove unnecessary index.js file

0.7.4 / 2013-11-13
==================

* remove "browserify" key from package.json (fixes something in browserify)

0.7.3 / 2013-10-30
==================

* fix: catch localStorage security error when cookies are blocked (Chrome)
* add debug(err) support. Closes #46
* add .browser prop to package.json. Closes #42

0.7.2 / 2013-02-06
==================================

* fix package.json
* fix: Mobile Safari (private mode) is broken with debug
* fix: Use unicode to send escape character to shell instead of octal to work with strict mode javascript

0.7.1 / 2013-02-05
==================================

* add repository URL to package.json
* add DEBUG_COLORED to force colored output
* add browserify support
* fix component. Closes #24

0.7.0 / 2012-05-04
==================================

* Added .component to package.json
* Added debug.component.js build

0.6.0 / 2012-03-16
==================================

* Added support for ":-" prefix in DEBUG [Vinay Pulim]
* Added `.enabled` flag to the node version [TooTallNate]

0.5.0 / 2012-02-02
==================================

* Added: humanize diffs. Closes #8
* Added `debug.disable()` to the CS variant
* Removed padding. Closes #10
* Fixed: persist client-side variant again. Closes #9

0.4.0 / 2012-02-01
==================================

* Added browser variant support for older browsers [TooTallNate]
* Added `debug.enable('project:*')` to browser variant [TooTallNate]
* Added padding to diff (moved it to the right)

0.3.0 / 2012-01-26
==================================
* Added millisecond diff when isatty, otherwise UTC string

0.2.0 / 2012-01-22

* Added wildcard support

0.1.0 / 2011-12-02

* Added: remove colors unless stderr isatty [TooTallNate]

0.0.1 / 2010-01-03

* Initial release

MIT License

Copyright (c) 2012-2019 Thorsten Lorenz, Paul Miller (https://paulmillr.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2005-2008, The Android Open Source Project
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designed in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
// Copyright 2014 The Chromium Authors. All rights reserved.
///< Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 1991-2017 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in http://www.unicode.org/copyright.html

Permission is hereby granted, free of charge, to any person obtaining
a copy of the Unicode data files and any associated documentation
(the "Data Files") or Unicode software and any associated documentation
(the "Software") to deal in the Data Files or Software
without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, and/or sell copies of
the Data Files or Software, and to permit persons to whom the Data Files
or Software are furnished to do so, provided that either
(a) this copyright and permission notice appear with all copies
of the Data Files or Software, or
(b) this copyright and permission notice appear in associated
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.
All trademarks and registered trademarks mentioned herein are the property of their respective owners.
Copyright 2008 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent
infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

For the file sRGB2014.icc:

Copyright International Color Consortium, 2015

This profile is made available by the International Color Consortium, and may be copied, distributed, embedded, made, used, and sold without restriction. Altered versions of this profile shall have the original identification and copyright information removed and shall not be misrepresented as the original profile.

(original source location: http://www.color.org/srgbprofiles.xalter)
Copyright 2007, 2008 The Python Markdown Project (v. 1.7 and later)
Copyright 2004, 2005, 2006 Yuri Takhteyev (v. 0.2-1.6b)
Copyright 2004 Manfred Stienstra (the original version)

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PYTHON MARKDOWN PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANY CONTRIBUTORS TO THE PYTHON MARKDOWN PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
MIT License

Copyright (c) Sam Verschueren <sam.verschueren@gmail.com> (github.com/SamVerschueren)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

musl as a whole is licensed under the following standard MIT license:

----------------------------------------------------------------------

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

----------------------------------------------------------------------

Authors/contributors include:
Valentin Ochs
William Haddon

Portions of this software are derived from third-party works licensed under terms compatible with the above MIT license:

The TRE regular expression implementation (src/regex/reg* and src/regex/tre*) is Copyright 2001-2008 Ville Laurikari and licensed under a 2-clause BSD license (license text in the source files). The included version has been heavily modified by Rich Felker in 2012, in the interests of size, simplicity, and namespace cleanliness.

Much of the math library code (src/math/* and src/complex/*) is Copyright 1993,2004 Sun Microsystems or Copyright 2003-2011 David Schultz or Copyright 2003-2009 Steven G. Kargl or Copyright 2003-2009 Bruce D. Evans or Copyright 2008 Stephen L. Moshier and labelled as such in comments in the individual source files. All have been licensed under extremely permissive terms.

The ARM memcpy code (src/string/arm/memcpy_el.S) is Copyright 2008 The Android Open Source Project and is licensed under a two-clause BSD license. It was taken from Bionic libc, used on Android.

The implementation of DES for crypt (src/crypt/crypt_des.c) is Copyright 1994 David Burren. It is licensed under a BSD license.

The implementation of blowfish crypt (src/crypt/crypt_blowfish.c) was originally written by Solar Designer and placed into the public domain. The code also comes with a fallback permissive license for use in jurisdictions that may not recognize the public domain.

The smoothsort implementation (src/stdlib/qsort.c) is Copyright 2011 Valentin Ochs and is licensed under an MIT-style license.

The BSD PRNG implementation (src/prng/random.c) and XSI search API (src/search/*.c) functions are Copyright 2011 Szabolcs Nagy and licensed under following terms: "Permission to use, copy, modify, and/or distribute this code for any purpose with or without fee is hereby granted. There is no warranty."

The x86_64 port was written by Nicholas J. Kain and is licensed under the standard MIT terms.

The mips and microblaze ports were originally written by Richard Pennington for use in the ellcc project. The original code was adapted by Rich Felker for build system and code conventions during upstream
integration. It is licensed under the standard MIT terms.

The mips64 port was contributed by Imagination Technologies and is licensed under the standard MIT terms.

The powerpc port was also originally written by Richard Pennington, and later supplemented and integrated by John Spencer. It is licensed under the standard MIT terms.

All other files which have no copyright comments are original works produced specifically for use as part of this library, written either by Rich Felker, the main author of the library, or by one or more contributors listed above. Details on authorship of individual files can be found in the git version control history of the project. The omission of copyright and license comments in each file is in the interest of source tree size.

In addition, permission is hereby granted for all public header files (include/* and arch/*/bits/*) and crt files intended to be linked into applications (crt/*, ldso/dlstart.c, and arch/*/crt_arch.h) to omit the copyright notice and permission notice otherwise required by the license, and to use these files without any requirement of attribution. These files include substantial contributions from:

Bobby Bingham
John Spencer
Nicholas J. Kain
Rich Felker
Richard Pennington
Stefan Kristiansson
Szabolcs Nagy

all of whom have explicitly granted such permission.

This file previously contained text expressing a belief that most of the files covered by the above exception were sufficiently trivial not to be subject to copyright, resulting in confusion over whether it negated the permissions granted in the license. In the spirit of permissive licensing, and of not having licensing issues being an obstacle to adoption, that text has been removed.

Copyright (c) 1994-2006 Sun Microsystems Inc.
All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Sun Microsystems or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (c) 2015 Kyle E. Mitchell & other authors listed in AUTHORS

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(WebKit doesn't distribute an explicit license. This LICENSE is derived from license text in the source.)

The terms and conditions vary from file to file, but are one of:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

*OR*

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of Apple Computer, Inc. ("Apple") nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY APPLE COMPUTER, INC. "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APPLE COMPUTER, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General
Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation
and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:
a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY
15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.
When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)
Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
END OF TERMS AND CONDITIONS
Copyright (C) 2012-2018 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(c) 2007-2009 Steven Levithan <stevenlevithan.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

BSD 3-Clause License

Copyright (c) 2005 - 2017, Jochen Kalmbach
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright 2015 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
//
// * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
// * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2010-2018 Caolan McMahon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Unshift.io, Arnout Kazemier, the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (C) 2013 Rase-

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2010, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional IP Rights Grant (Patents)
------------------------------------

“These implementations” means the copyrightable works that implement the WebM codecs distributed by Google as part of the WebM Project.

Google hereby grants to you a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, transfer, and otherwise run, modify and propagate the contents of these implementations of WebM, where such license applies only to those patent claims, both currently owned by Google and acquired in the future, licensable by Google that are necessarily infringed by these implementations of WebM. This grant does not include claims that would be infringed only as a consequence of further modification of these implementations. If you or your agent or exclusive licensee institute or order or agree to the institution of patent litigation or any other patent enforcement activity against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that any of these implementations of WebM or any code incorporated within any of these implementations of WebM constitute direct or contributory patent infringement, or inducement of patent infringement, then any patent rights granted to you under this License for these implementations of WebM shall terminate as of the date such litigation is filed.

The MIT License (MIT)

Copyright (c) 2013 Mikola Lysenko

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

Open Source Used In DNAC Platform Antelope 11658
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# License

Most files in FFmpeg are under the GNU Lesser General Public License version 2.1 or later (LGPL v2.1+). Read the file `COPYING.LGPLv2.1` for details. Some other files have MIT/X11/BSD-style licenses. In combination the LGPL v2.1+ applies to FFmpeg.

Some optional parts of FFmpeg are licensed under the GNU General Public License version 2 or later (GPL v2+). See the file `COPYING.GPLv2` for details. None of these parts are used by default, you have to explicitly pass `--enable-gpl` to configure to activate them. In this case, FFmpeg's license changes to GPL v2+.

Specifically, the GPL parts of FFmpeg are:

- libpostproc
- optional x86 optimization in the files
  - `libavcodec/x86/flac_dsp_gpl.asm`
  - `libavcodec/x86/idct_mmx.c`
  - `libavfilter/x86/vf_removegrain.asm`
- the following building and testing tools
  - `compat/solaris/make_sunver.pl`
  - `doc/t2h.pm`
  - `doc/tx2ipod.pl`
  - `libswresample/tests/swresample.c`
  - `tests/checkasm/*`
  - `tests/tiny_ssim.c`
- the following filters in libavfilter:
  - `signature_lookup.c`
  - `vf_blackframe.c`
  - `vf_boxblur.c`
  - `vf_colormatrix.c`
  - `vf_cover_rect.c`
  - `vf_cropdetect.c`
  - `vf_delogo.c`
  - `vf_eq.c`
  - `vf_find_rect.c`
  - `vf_fspp.c`
The following libraries are under GPL version 2:
- avisynth
- frei0r
- libcdio
- libdavs2
- librubberband
- libvidstab
- libx264
- libx265
- libxavs
- libxavs2
- libxvid

When combining them with FFmpeg, FFmpeg needs to be licensed as GPL as well by passing `--enable-gpl` to configure.

The following libraries are under LGPL version 3:
- gmp
- libaribb24
- liblensfun

When combining them with FFmpeg, use the configure option `--enable-version3` to upgrade FFmpeg to the LGPL v3.

The VMAF, mbedTLS, RK MPI, OpenCORE and VisualOn libraries are under the Apache License 2.0. That license is incompatible with the LGPL v2.1 and the GPL v2, but not with version 3 of those licenses. So to combine these libraries with FFmpeg, the license version needs to be upgraded by passing `--enable-version3` to configure.

The smbclient library is under the GPL v3, to combine it with FFmpeg, the options `--enable-gpl` and `--enable-version3` have to be passed to configure to upgrade FFmpeg to the GPL v3.

### Incompatible libraries

There are certain libraries you can combine with FFmpeg whose licenses are not compatible with the GPL and/or the LGPL. If you wish to enable these libraries, even in circumstances that their license may be incompatible, pass `--enable-nonfree` to configure. This will cause the resulting binary to be unredistributable.

The Fraunhofer FDK AAC and OpenSSL libraries are under licenses which are incompatible with the GPLv2 and v3. To the best of our knowledge, they are compatible with the LGPL.

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2014 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object...
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2017 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

GNU LESSER GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2015 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt GUI Toolkit under the terms of GNU Lesser General Public License version 3, which is displayed below. This license makes reference to the version 3 of the GNU General Public License, which you can find in the LICENSE.GPLv3 file.

REPUGUAN LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, this License refers to version 3 of the GNU Lesser General Public License, and the GNU GPL refers to version 3 of the GNU General Public License.

The Library refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.
An Application is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A Combined Work is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the Linked Version.

The Minimal Corresponding Source for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The Corresponding Application Code for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this
license document.


You may convey a Combined Work under terms of your choice that, taken
together, effectively do not restrict modification of the portions of
the Library contained in the Combined Work and reverse engineering for
debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this
license document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of
this License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with
the Library. A suitable mechanism is one that (a) uses at run
time a copy of the Library already present on the user's
computer system, and (b) will operate properly with a modified
version of the Library that is interface-compatible with the
Linked Version.

e) Provide Installation Information, but only if you would
otherwise be required to provide such information under section 6
of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

   b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License or any later version applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
The source code of UnRAR utility is freeware. This means:

1. All copyrights to RAR and the utility UnRAR are exclusively owned by the author - Alexander Roshal.

2. UnRAR source code may be used in any software to handle RAR archives without limitations free of charge, but cannot be used to develop RAR (WinRAR) compatible archiver and to re-create RAR compression algorithm, which is proprietary. Distribution of modified UnRAR source code in separate form or as a part of other software is permitted, provided that full text of this paragraph, starting from "UnRAR source code" words, is included in license, or in documentation if license is not available, and in source code comments of resulting package.

3. The UnRAR utility may be freely distributed. It is allowed to distribute UnRAR inside of other software packages.

4. THE RAR ARCHIVER AND THE UnRAR UTILITY ARE DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.

5. Installing and using the UnRAR utility signifies acceptance of these terms and conditions of the license.

6. If you don't agree with terms of the license you must remove UnRAR files from your storage devices and cease to use the utility.

Thank you for your interest in RAR and UnRAR.

Alexander L. Roshal

S2 Geometry Library
Copyright 2017 Google Inc. All Rights Reserved.

This product includes software developed at Google (https://www.google.com/).
Copyright 2019 web-platform-tests contributors

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2018 Dmitry Shirokov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2010-2017 Mike Bostock
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the author nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (c) 2011 Dominic Tarr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/* ***** BEGIN LICENSE BLOCK *****
 * Version: MPL 1.1/GPL 2.0/LGPL 2.1
 */
* The contents of this file are subject to the Mozilla Public License Version
* 1.1 (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
* http://www.mozilla.org/MPL/
*
* Software distributed under the License is distributed on an "AS IS" basis,
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
* for the specific language governing rights and limitations under the
* License.
*
* The Original Code is the Netscape Portable Runtime (NSPR).
*
* The Initial Developer of the Original Code is
* Netscape Communications Corporation.
* Portions created by the Initial Developer are Copyright (C) 1998-2000
* the Initial Developer. All Rights Reserved.
*
* Contributor(s):
*
* Alternatively, the contents of this file may be used under the terms of
* either the GNU General Public License Version 2 or later (the "GPL"), or
* the GNU Lesser General Public License Version 2.1 or later (the "LGPL"),
* in which case the provisions of the GPL or the LGPL are applicable instead
* of those above. If you wish to allow use of your version of this file only
* under the terms of either the GPL or the LGPL, and not to allow others to
* use your version of this file under the terms of the MPL, indicate your
* decision by deleting the provisions above and replace them with the notice
* and other provisions required by the GPL or the LGPL. If you do not delete
* the provisions above, a recipient may use your version of this file under
* the terms of any one of the MPL, the GPL or the LGPL.
*
* ***** END LICENSE BLOCK ***** */

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2014 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies of
the Software and that both the above copyright notice(s) and this
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF
THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS
INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT
OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall
not be used in advertising or otherwise to promote the sale, use or
other dealings in this Software without prior written authorization of
the copyright holder.

All trademarks and registered trademarks mentioned herein are the
property of their respective owners.

Third-Party Software Licenses

This section contains third-party software notices and/or additional
terms for licensed third-party software components included within ICU
libraries.

1. Unicode Data Files and Software

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2014 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in
http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining
a copy of the Unicode data files and any associated documentation
(the "Data Files") or Unicode software and any associated documentation
(the "Software") to deal in the Data Files or Software
without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, and/or sell copies of
the Data Files or Software, and to permit persons to whom the Data Files
or Software are furnished to do so, provided that
(a) this copyright and permission notice appear with all copies
of the Data Files or Software,
(b) this copyright and permission notice appear in associated
documentation, and
(c) there is clear notice in each modified Data File or in the Software
as well as in the documentation associated with the Data File(s) or
Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale,
use or other dealings in these Data Files or Software without prior
written authorization of the copyright holder.


# The Google Chrome software developed by Google is licensed under the BSD li-
cense. Other software included in this distribution is provided under other licen-
ses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without modifi-
cation, are permitted provided that the following conditions are met:
# Redistribution of source code must retain the above copyright notice, th
is list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.
# Neither the name of Google Inc. nor the names of its contributors may be
used to endorse or promote products derived from this software without specific
prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS I
S" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#
# The word list in cjdict.txt are generated by combining three word lists listed below with further processing for compound word breaking. The frequency is generated with an iterative training against Google web corpora.
#
# * Libtabe (Chinese)
#   - https://sourceforge.net/project/?group_id=1519
#   - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#   - Its license terms and conditions are shown below.
#
# --------COPYING.libtabe ---- BEGIN---------------------
#
# /*
# * Copyright (c) 1999 TaBE Project.
# * Copyright (c) 1999 Pai-Hsiang Hsiao.
# * All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * . Neither the name of the TaBE Project nor the names of its
# * contributors may be used to endorse or promote products derived
# * from this software without specific prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
*/

Copyright (c) 1999 Computer Systems and Communication Lab,
Institute of Information Science, Academia Sinica.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the
distribution.
. Neither the name of the Computer Systems and Communication Lab
nor the names of its contributors may be used to endorse or
promote products derived from this software without specific
prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Copyright 1996 Chih-Hao Tsai @ Beckman Institute, University of Illinois

Use, reproduction, and distribution of this software is permitted. Any copy of this software, whether in its original form or modified, must include both the above copyright notice and the following paragraphs.

Nara Institute of Science and Technology (NAIST), the copyright holders, disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall NAIST be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term “use” as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

---COPYING.ipadic---END------------------------------------

3. Lao Word Break Dictionary Data (laodict.txt)

Copyright (c) 2013 International Business Machines Corporation and others. All Rights Reserved.

Project: http://code.google.com/p/lao-dictionary/
License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt

(copied below)

This file is derived from the above dictionary, with slight modifications.

---COPYING.ipadic---END------------------------------------
Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

4. Burmese Word Break Dictionary Data (burmesedict.txt)
Copyright (c) 2014 International Business Machines Corporation and others. All Rights Reserved.
This list is part of a project hosted at: github.com/kanyawtech/myanmar-karen-word-lists

----------------------------------------
# Copyright (c) 2013, LeRoy Benjamin Sharon
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without modifi-
cation,
# are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this
# list of conditions and the following disclaimer.
#
# Redistributions in binary form must reproduce the above copyright notic-
e, this
# list of conditions and the following disclaimer in the documentation an-
d/or
# other materials provided with the distribution.
#
# Neither the name Myanmar Karen Word Lists, nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS I
S" AND
# ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMP
LIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LI-
ABLE FOR
# ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DA-
MAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVIC-
ES;
# LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED A
ND ON
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
# (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#

5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database
for its time zone support. The ownership of the TZ database is explained
in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

7. Database Ownership
The TZ database itself is not an IETF Contribution or an IETF document. Rather it is a pre-existing and regularly updated work that is in the public domain, and is intended to remain in the public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do not apply to the TZ Database or contributions that individuals make to it. Should any claims be made and substantiated against the TZ Database, the organization that is providing the IANA Considerations defined in this RFC, under the memorandum of understanding with the IETF, currently ICANN, may act in accordance with all competent court orders. No ownership claims will be made by ICANN or the IETF Trust on the database or the code. Any person making a contribution to the database or code waives all rights to future claims in that contribution or in the TZ Database.

Copyright (C) 1993-2004 by Sun Microsystems, Inc. All rights reserved.

Developed at SunSoft, a Sun Microsystems, Inc. business.
Permission to use, copy, modify, and distribute this software is freely granted, provided that this notice is preserved.

GNU GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2015 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt Toolkit under the terms of GNU Lesser General Public License version 3. That license references the General Public License version 3, that is displayed below. Other portions of the Qt Toolkit may be licensed directly under this license.

-------------------------------------------------------------------------

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could
make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its
content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.
You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.
When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a
If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

NAME: WebDriver
URL: http://selenium.googlecode.com/svn/trunk/py
LICENSE: Apache 2
The MIT License (MIT)

Copyright (c) 2013-2015 Roman Shtylman <shtylman@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
(The MIT License)

Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

****************************************************************************
** Copyright (C) YYYY The Qt Company Ltd.
** Contact: https://www.qt.io/licensing/
**
** This file is part of the FOO module of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:GPL$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms
** and conditions see https://www.qt.io/terms-conditions. For further
** information use the contact form at https://www.qt.io/contact-us.

Open Source Used In DNAC Platform Antelope 11701
Copyright (c) 2013 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2017 Made With MOXY Lda <hello@moxy.studio>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/* ***** BEGIN LICENSE BLOCK *****
* Version: MPL 1.1/GPL 2.0/LGPL 2.1
*
* The contents of this file are subject to the Mozilla Public License Version
* 1.1 (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
* http://www.mozilla.org/MPL/
*
* Software distributed under the License is distributed on an "AS IS" basis,
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
* for the specific language governing rights and limitations under the
* License.
*
* The Original Code is mozilla.org code.
*
* The Initial Developer of the Original Code is
* Netscape Communications Corporation.
* Portions created by the Initial Developer are Copyright (C) 2001
* the Initial Developer. All Rights Reserved.
*
* Contributor(s):
*
* Alternatively, the contents of this file may be used under the terms of
* either the GNU General Public License Version 2 or later (the "GPL"), or
* the GNU Lesser General Public License Version 2.1 or later (the "LGPL"),
* in which case the provisions of the GPL or the LGPL are applicable instead
* of those above. If you wish to allow use of your version of this file only
* under the terms of either the GPL or the LGPL, and not to allow others to
* use your version of this file under the terms of the MPL, indicate your
* decision by deleting the provisions above and replace them with the notice
* and other provisions required by the GPL or the LGPL. If you do not delete
* the provisions above, a recipient may use your version of this file under
* the terms of any one of the MPL, the GPL or the LGPL.
*
* ***** END LICENSE BLOCK ***** */
Copyright (c) 2015-2017 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and/or associated documentation files (the
"Materials"), to deal in the Materials without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Materials, and to
permit persons to whom the Materials are furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

The MIT License (MIT)

Copyright (c) 2014 Josh Wolfe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2003-2012, Michael Foord
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2017 Chai.js Assertion Library

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2003,2004 Colin Percival
All rights reserved

Redistribution and use in source and binary forms, with or without modification, are permitted providing that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR `AS IS` AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

//*****************************************************************************
//* Base64 - a simple base64 encoder and decoder.                          *
//*                                                                       *
//*   Copyright (c) 1999, Bob Withers - bwit@pobox.com                    *
//*                                                                       *
//* This code may be freely used for any purpose, either personal        *
//* or commercial, provided the authors copyright notice remains        *
//* intact.                                                             *
//*                                                                       *
//*    Enhancements by Stanley Yamane:                                   *
//*       o reverse lookup table for the decode function                 *
//*       o reserve string buffer space in advance                      *
//*                                                                       *
//*****************************************************************************

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License
treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the
application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.
If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work
Open Source Used In DNAC Platform Antelope  11712

during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

d) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the source code distributed need not include anything that is normally
distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing
NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Library General Public License for more details.

You should have received a copy of the GNU Library General Public
License along with this library; if not, write to the Free
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
wcwidth.js: JavaScript Porting of Markus Kuhn's wcwidth() Implementation
=======================================================================
Copyright (C) 2012 by Jun Woong.

This package is a JavaScript porting of `wcwidth()` implementation
[by Markus Kuhn][http://www.cl.cam.ac.uk/~mgk25/ucs/wcwidth.c].

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2009-2017 The OTS Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2013 Gareth Jones

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
LICENSE
=======
This is an installer created using CPack (http://www.cmake.org). No license provided.
// Copyright (c) 2006-2009 The Chromium OS Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//  * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//  * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
//  * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)
Copyright (c) 2012-2014 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2006-2019 the Mako authors and contributors <see AUTHORS file>.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# timers-browserify

This project uses the [MIT](http://jryans.mit-license.org/) license:

Copyright 2012 J. Ryan Stinnett <jryans@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# lib/node

The `lib/node` directory borrows files from joyent/node which uses the following license:

Copyright Joyent, Inc. and other Node contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

www.solarsystemscope.com/textures distributed under the terms of https://creativecommons.org/licenses/by/4.0/

Changes:
- Rings created from the reference rectangle file.
- earthcloudmapcolortrans.png was made from the earthcloudmapspec.jpg file
NSS is available under the Mozilla Public License, version 2, a copy of which is below.

Note on GPL Compatibility
----------------------

The MPL 2, section 3.3, permits you to combine NSS with code under the GNU
General Public License (GPL) version 2, or any later version of that license, to make a Larger Work, and distribute the result under the GPL. The only condition is that you must also make NSS, and any changes you have made to it, available to recipients under the terms of the MPL 2 also.

Anyone who receives the combined code from you does not have to continue to dual licence in this way, and may, if they wish, distribute under the terms of either of the two licences - either the MPL alone or the GPL alone. However, we discourage people from distributing copies of NSS under the GPL alone, because it means that any improvements they make cannot be reincorporated into the main version of NSS. There is never a need to do this for license compatibility reasons.

Note on LGPL Compatibility
--------------------------

The above also applies to combining MPLed code in a single library with code under the GNU Lesser General Public License (LGPL) version 2.1, or any later version of that license. If the LGPLed code and the MPLed code are not in the same library, then the copyleft coverage of the two licences does not overlap, so no issues arise.

Mozilla Public License Version 2.0
==================================

1. Definitions
---------------

1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means
(a) that the initial Contributor has attached the notice described 
in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of 
version 1.1 or earlier of the License, but not also under the 
terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in 
a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible, 
whether at the time of the initial grant or subsequently, any and 
all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to, 
deletion from, or modification of the contents of Covered 
Software; or

(b) any new file in Source Code Form that contains any Covered 
Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method, 
process, and apparatus claims, in any patent Licensable by such 
Contributor that would be infringed, but for the grant of the 
License, by the making, using, selling, offering for sale, having 
made, import, or transfer of either its Contributions or its 
Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU 
Lesser General Public License, Version 2.1, the GNU Affero General 
Public License, Version 3.0, or any later versions of those 
licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your") means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions
--------------------------------

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party’s
modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

-------------------

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.
3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.
4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.
Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

************************************************************************
************************************************************************

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous
This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.
If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

---------------------------------------------------------

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.
Copyright (c) 2017-2018 Advanced Micro Devices, Inc. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:

May you do good and not evil.
May you find forgiveness for yourself and forgive others.
May you share freely, never taking more than you give.

PEXPECT LICENSE

This license is approved by the OSI and FSF as GPL-compatible.
http://opensource.org/licenses/isc-license.txt

Copyright (c) 2012, Noah Spurrier <noah@noah.org>

PERMISSION TO USE, COPY, MODIFY, AND/OR DISTRIBUTE THIS SOFTWARE FOR ANY PURPOSE WITH OR WITHOUT FEE IS HEREBY GRANTED, PROVIDED THAT THE ABOVE COPYRIGHT NOTICE AND THIS PERMISSION NOTICE APPEAR IN ALL COPIES.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

Use, modification & distribution is subject to Boost Software License Ver 1. http://www.boost.org/LICENSE_1_0.txt

Attributions:
The code in this library is an extension of Bala Vatti's clipping algorithm:
"A generic solution to polygon clipping"
http://portal.acm.org/citation.cfm?id=129906

Computer graphics and geometric modeling: implementation and algorithms
By Max K. Agoston
Springer; 1 edition (January 4, 2005)
http://books.google.com/books?q=vatti+clipping+agoston

See also:
"Polygon Offsetting by Computing Winding Numbers"
Paper no. DETC2005-85513 pp. 565-575
ASME 2005 International Design Engineering Technical Conference
and Computers and Information in Engineering Conference (IDETC/CIE2005)
September 24-28, 2005 , Long Beach, California, USA
http://www.me.berkeley.edu/~mcmains/pubs/DAC05OffsetPolygon.pdf
Licenses of dependencies as produced by `cargo-license`:

MIT (1): synstructure
MIT OR Apache-2.0 (9): once_cell, failure, log, quote, proc-macro2, syn, failure_derive, unicode-xid, libc
Unlicense OR MIT (1): byteorder

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

HarfBuzz is licensed under the so-called "Old MIT" license. Details follow. For parts of HarfBuzz that are licensed under different licenses see individual files names COPYING in subdirectories where applicable.

Copyright 2012 Mozilla Foundation
Copyright 2011 Codethink Limited
Copyright 2008,2010 Nokia Corporation and/or its subsidiary(-ies)
Copyright 2009 Keith Stribley
Copyright 2009 Martin Hosken and SIL International
Copyright 2007 Chris Wilson
Copyright 2006 Behdad Esfahbod
Copyright 2005 David Turner
Copyright 1998-2004 David Turner and Werner Lemberg

For full copyright notices consult the individual files in the package.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Copyright (c) 2011 The Chromium OS Authors. All rights reserved.
Use of this source code is governed by a BSD-style license that can be found in the LICENSE file.
Copyright (c) 2015, Ilya Radchenko <ilya@burstcreations.com>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

---

Open Source Used In DNAC Platform Antelope 11736
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
// Copyright (c) 2011 Google Inc. All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.

// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright (c) 2018 Google Inc. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2015-2017 Rubn Norte <rubennorte@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

 OGRE (www.ogre3d.org) is made available under the MIT License.

Copyright (c) 2000-2013 Torus Knot Software Ltd

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

// Copyright (c) 2011 The Chromium OS Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// Copyright (c) 2012 The Chromium Authors. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
// * Redistributions of source code must retain the above copyright
//   notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
//   copyright notice, this list of conditions and the following disclaimer
//   in the documentation and/or other materials provided with the
//   distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*
 ** Copyright (c) 2007-2010 The Khronos Group Inc.
 **
 ** Permission is hereby granted, free of charge, to any person obtaining a
 ** copy of this software and/or associated documentation files (the
 ** "Materials"), to deal in the Materials without restriction, including
 ** without limitation the rights to use, copy, modify, merge, publish,
 ** distribute, sublicense, and/or sell copies of the Materials, and to
 ** permit persons to whom the Materials are furnished to do so, subject to
 ** the following conditions:
 **
 ** The above copyright notice and this permission notice shall be included
 ** in all copies or substantial portions of the Materials.
 **
 ** THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 ** EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 ** MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 ** IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
*/
The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Qt port of geosimplify.js, https://github.com/mapbox/geosimplify-js

Copyright (c) 2017, Daniel Patterson
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---------------------------------
Based on simplify-js by Vladimir Agafonkin - http://mourner.github.io/simplify-js/
ISC License

Copyright (c) 2015, Mapbox

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License

Copyright (C) 2011-2013 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
GNU GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2017 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt Toolkit under the terms of GNU Lesser General Public License version 3. That license references the General Public License version 3, that is displayed below. Other portions of the Qt Toolkit may be licensed directly under this license.

-------------------------------------------------------------------------

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.
The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.
No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties’ legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the
Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply
if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or
requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.
Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for
11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

A patent license is “discriminatory” if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License. You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all. For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work. The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

14. Revised Versions of this License.
The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.
If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Copyright (c) 2014-2019 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012, 2013 Thorsten Lorenz <thlorenz@gmx.de>
Copyright (c) 2012 James Halliday <mail@substack.net>
Copyright (c) 2009 Thomas Robinson <280north.com>

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2016 Google Inc. All rights reserved.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with the Licensor.
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2016, Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License (MIT)

Copyright (c) 2014 Cameron Lowell Palmer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Open Source Used In DNAC Platform Antelope 11768
Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===

Some of the benchmark data in testdata/ is licensed differently:

- fireworks.jpeg is Copyright 2013 Steinar H. Gunderson, and is licensed under the Creative Commons Attribution 3.0 license (CC-BY-3.0). See https://creativecommons.org/licenses/by/3.0/ for more information.

- kppkn.gtb is taken from the Gaviota chess tablebase set, and is licensed under the MIT License. See https://sites.google.com/site/gaviotachessengine/Home/endgame-tablebases-1 for more information.

- paper-100k.pdf is an excerpt (bytes 92160 to 194560) from the paper Combinatorial Modeling of Chromatin Features Quantitatively Predicts DNA Replication Timing in _Drosophila_ by Federico Comoglio and Renato Paro, which is licensed under the CC-BY license. See http://www.ploscompbiol.org/static/license for more information.
- Alice29.txt, asyoulik.txt, plrabin12.txt and lcet10.txt are from Project Gutenberg. The first three have expired copyrights and are in the public domain; the latter does not have expired copyright, but is still in the public domain according to the license information (http://www.gutenberg.org/ebooks/53).

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

qimagetransform.cpp was contributed by Daniel M. Duley based on code from Imlib2.

Copyright (C) 2004, 2005 Daniel M. Duley

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Imlib2 License

Copyright (C) 2000 Carsten Haitzler and various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies of the Software and its Copyright notices. In addition publicly documented acknowledgment must be given that this software has been used if no source code of this software is made available publicly. This includes acknowledgments in either Copyright notices, Manuals, Publicity and Marketing documents or any documentation provided with any product containing this software. This License does not apply to any software that links to the libraries provided by this software (statically or dynamically), but only to the software provided.

Please see the COPYING.PLAIN for a plain-english explanation of this notice and it's intent.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

language: node_js
node_js:
  - "6"
  - "8"
cache:
directories:
  - "node_modules"
MIT License

Copyright (c) 2014-2018 Suguru Motegi
Based on Async.js, Copyright Caolan McMahon
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Extracted from mac/include/DeckLinkAPI.h:

** Copyright (c) 2014 Blackmagic Design
**
** Permission is hereby granted, free of charge, to any person or organization
** obtaining a copy of the software and accompanying documentation covered by
** this license (the "Software") to use, reproduce, display, distribute,
** execute, and transmit the Software, and to prepare derivative works of the
** Software, and to permit third-parties to whom the Software is furnished to
** do so, all subject to the following:
**
** The copyright notices in the Software and this entire statement, including
** the above license grant, this restriction and the following disclaimer,
** must be included in all copies of the Software, in whole or in part, and
** all derivative works of the Software, unless such copies or derivative
** works are solely in the form of machine-executable object code generated by
** a source language processor.

**

** THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
** IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
** FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
** SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
** FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
** ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
** DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 Barnesandnoble.com, llc, Donavon West, and Domenic Denicola

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
Copyright (C) 1999-2000 Mizi Research Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

GNU LESSER GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2015 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt Toolkit under the terms of
GNU Lesser General Public License version 2.1, which is displayed below.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library...
with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above,

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:
Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

# debug


![image](https://user-images.githubusercontent.com/71256/29091486-fa38524c-7c37-11e7-895f-e7e8e1039b6.png)

A tiny JavaScript debugging utility modelled after Node.js core's debugging technique. Works in Node.js and web browsers.

## Installation

```bash
$ npm install debug
```

## Usage

`debug` exposes a function; simply pass this function the name of your module, and it will return a decorated version of `console.error` for you to pass debug statements to. This will allow you to toggle the debug output for different parts of your module as well as the module as a whole.

Example [_app.js_](./examples/node/app.js):

```js
var debug = require('debug')('http')
  , http = require('http')
  , name = 'My App';

// fake app

debug('booting %o', name);

http.createServer(function(req, res){
  debug(req.method + ' ' + req.url);
  // handle request
});
```
res.end('hello\n');
}).listen(3000, function(){
 debug('listening');
});

// fake worker of some kind

require('./worker');
```
```

Example [worker.js](./examples/node/worker.js):

```
```

var a = require('debug')('worker:a')
  , b = require('debug')('worker:b');

function work() {
  a('doing lots of uninteresting work');
  setTimeout(work, Math.random() * 1000);
}
work();

function workb() {
  b('doing some work');
  setTimeout(workb, Math.random() * 2000);
}
workb();
```

The `DEBUG` environment variable is then used to enable these based on space or comma-delimited names.

Here are some examples:

```
```

##### Windows command prompt notes

***** CMD

On Windows the environment variable is set using the `set` command.
Example:

```cmd
set DEBUG=*,-not_this
```

##### PowerShell (VS Code default)

PowerShell uses different syntax to set environment variables.

```cmd
$env:DEBUG = "*,-not_this"
```

Example:

```cmd
$env:DEBUG='app';node app.js
```

Then, run the program to be debugged as usual.

npm script example:

```js
"windowsDebug": "@powershell -Command $env:DEBUG='*';node app.js",
```

## Namespace Colors

Every debug instance has a color generated for it based on its namespace name. This helps when visually parsing the debug output to identify which debug instance a debug line belongs to.

#### Node.js

In Node.js, colors are enabled when stderr is a TTY. You also _should_ install the [supports-color](https://npmjs.org/supports-color) module alongside debug, otherwise debug will only use a small handful of basic colors.

<img width="521" src="https://user-images.githubusercontent.com/71256/29092181-47f6a9e6-7c3a-11e7-9a14-1928d8a711cd.png">

#### Web Browser
Colors are also enabled on "Web Inspectors" that understand the \`%c` formatting option. These are WebKit web inspectors, Firefox ([since version 31](https://hacks.mozilla.org/2014/05/editable-box-model-multiple-selection-sublime-text-keys-much-more-firefox-developer-tools-episode-31/)) and the Firebug plugin for Firefox (any version).

When stdout is not a TTY, `Date#toISOString()` is used, making it more useful for logging the debug information as shown below:

#### Millisecond diff

When actively developing an application it can be useful to see when the time spent between one `debug()` call and the next. Suppose for example you invoke `debug()` before requesting a resource, and after as well, the "+NNNms" will show you how much time was spent between calls.

When stdout is not a TTY, `Date#toISOString()` is used, making it more useful for logging the debug information as shown below:

#### Conventions

If you're using this in one or more of your libraries, you _should_ use the name of your library so that developers may toggle debugging as desired without guessing names. If you have more than one debuggers you _should_ prefix them with your library name and use ":" to separate features. For example "bodyParser" from Connect would then be "connect:bodyParser". If you append a "+" to the end of your name, it will always be enabled regardless of the setting of the DEBUG environment variable. You can then use it for normal output as well as debug output.

#### Wildcards

The "+" character may be used as a wildcard. Suppose for example your library has debuggers named "connect:bodyParser", "connect:compress", "connect:session", instead of listing all three with

```
'DEBUG=connect:bodyParser,connect:compress,connect:session'
```

you may simply do

```
'DEBUG=connect:*
```

to run everything using this module simply use 

```
'DEBUG=*
```

You can also exclude specific debuggers by prefixing them with a "-" character. For example, 

```
'DEBUG=*,-connect:*
```

would include all debuggers except those starting with "connect:".

#### Environment Variables
When running through Node.js, you can set a few environment variables that will change the behavior of the debug logging:

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DEBUG</code></td>
<td>Enables/disables specific debugging namespaces.</td>
</tr>
<tr>
<td><code>DEBUG_HIDE_DATE</code></td>
<td>Hide date from debug output (non-TTY).</td>
</tr>
<tr>
<td><code>DEBUG_COLORS</code></td>
<td>Whether or not to use colors in the debug output.</td>
</tr>
<tr>
<td><code>DEBUG_DEPTH</code></td>
<td>Object inspection depth.</td>
</tr>
<tr>
<td><code>DEBUG_SHOW_HIDDEN</code></td>
<td>Shows hidden properties on inspected objects.</td>
</tr>
</tbody>
</table>

__Note:_ The environment variables beginning with `DEBUG_` end up being converted into an Options object that gets used with ` `%o`/`%O` formatters. See the Node.js documentation for [``util.inspect()`](https://nodejs.org/api/util.html#util_util_inspect_object_options) for the complete list.

### Formatters

Debug uses [printf-style](https://wikipedia.org/wiki/Printf_format_string) formatting. Below are the officially supported formatters:

<table>
<thead>
<tr>
<th>Formatter</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>%O</code></td>
<td>Pretty-print an Object on multiple lines.</td>
</tr>
<tr>
<td><code>%o</code></td>
<td>Pretty-print an Object all on a single line.</td>
</tr>
<tr>
<td><code>%s</code></td>
<td>String.</td>
</tr>
<tr>
<td><code>%d</code></td>
<td>Number (both integer and float).</td>
</tr>
<tr>
<td><code>%j</code></td>
<td>JSON. Replaced with the string <code>[Circular]</code> if the argument contains circular references.</td>
</tr>
<tr>
<td><code>%%</code></td>
<td>Single percent sign (<code>%</code>). This does not consume an argument.</td>
</tr>
</tbody>
</table>

### Custom formatters

You can add custom formatters by extending the `debug.formatters` object. For example, if you wanted to add support for rendering a Buffer as hex with ` `%h``, you could do something like:

```js
const createDebug = require('debug')
createDebug.formatters.h = (v) => {
    return v.toString('hex')
}
```

```js
// elsewhere
cost debug = createDebug('foo')
```
debug('this is hex: %h', new Buffer('hello world'))

if foo this is hex: 68656c6c6f20776f726c6421 +0ms

```

## Browser Support

You can build a browser-ready script using [browserify](https://github.com/substack/node-browserify), or just use the [browserify-as-a-service](https://wzrd.in/) [build](https://wzrd.in/standalone/debug@latest), if you don't want to build it yourself.

Debug's enable state is currently persisted by `localStorage`.
Consider the situation shown below where you have `worker:a` and `worker:b`, and wish to debug both. You can enable this using `localStorage.debug`:

```
```js
localStorage.debug = 'worker:*'
```

And then refresh the page.

```
```js
a = debug('worker:a');
b = debug('worker:b');

setInterval(function()
  a('doing some work');
}, 1000);

setInterval(function()
  b('doing some work');
}, 1200);
```

## Output streams

By default `debug` will log to stderr, however this can be configured per-namespace by overriding the `log` method:

Example [stdout.js](./examples/node/stdout.js):

```
```js
var debug = require('debug');
var error = debug('app:error');

// by default stderr is used
error('goes to stderr!');
```
var log = debug('app:log');
// set this namespace to log via console.log
log.log = console.log.bind(console); // don't forget to bind to console!
log('goes to stdout');
error('still goes to stderr!');

// set all output to go via console.info
// overrides all per-namespace log settings
debug.log = console.info.bind(console);
error('now goes to stdout via console.info');
log('still goes to stdout, but via console.info now');
```

## Extend
You can simply extend debugger
```
const log = require('debug')('auth');

// creates new debug instance with extended namespace
const logSign = log.extend('sign');
const logLogin = log.extend('login');

log('hello'); // auth hello
logSign('hello'); // auth:sign hello
logLogin('hello'); // auth:login hello
```

## Set dynamically
You can also enable debug dynamically by calling the `enable()` method:
```
let debug = require('debug');

console.log(1, debug.enabled('test'));

debug.enable('test');
console.log(2, debug.enabled('test'));

debug.disable();
console.log(3, debug.enabled('test'));
```

print :
```
I false
Usage:
`
`enable(namespaces)`
`namespaces` can include modes separated by a colon and wildcards.

Note that calling `enable()` completely overrides previously set DEBUG variable:

```javascript
const debug = require('debug')('http');
if (debug.enabled) {
    // do stuff...
}
```

You can also manually toggle this property to force the debug instance to be enabled or disabled.

## Authors
- TJ Holowaychuk
- Nathan Rajlich
- Andrew Rhyne

## Backers
Support us with a monthly donation and help us continue our activities. [[Become a backer]](https://opencollective.com/debug#backer)
## Sponsors

Become a sponsor and get your logo on our README on Github with a link to your site. [[Become a sponsor](https://opencollective.com/debug#sponsor)]
## License

(The MIT License)

Copyright (c) 2014-2017 TJ Holowaychuk &lt;tj@vision-media.ca&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, 
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE 
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2013-2018 Franck34

Permission is hereby granted, free of charge, to any person obtaining a copy 
of this software and associated documentation files (the "Software"), to deal 
in the Software without restriction, including without limitation the rights 
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom the Software is 
 furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in 
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR 
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, 
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE 
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER 
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, 
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN 
THE SOFTWARE.

ISC License
Copyright (c) 2016 Mapbox

Permission to use, copy, modify, and/or distribute this software for any purpose 
with or without fee is hereby granted, provided that the above copyright notice 
and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND ISC DISCLAIMS ALL WARRANTIES WITH REGARD TO 
THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. 
IN NO EVENT SHALL ISC BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR 
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, 
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS 
ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS 
SOFTWARE.

MIT License

---------

Copyright (C) 2010-2019 by Philipp Dunkel, Ben Noordhuis, Elan Shankar, Paul Miller

Permission is hereby granted, free of charge, to any person obtaining a copy 
of this software and associated documentation files (the "Software"), to deal 
in the Software without restriction, including without limitation the rights 
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
Copyright (c) 2013 Thiago de Arruda

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2017, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright notice
and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF
THIS SOFTWARE.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

# The MIT License (MIT)

Copyright 2016-2019 Martin Mit

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the Software),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

The MIT License (MIT)

Copyright (C) 2013 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

APPLE PUBLIC SOURCE LICENSE
Version 2.0 - August 6, 2003

Please read this License carefully before downloading this software. By
downloading or using this software, you are agreeing to be bound by the terms of this License. If you do not or cannot agree to the terms of this License, please do not download or use the software.

Apple Note: In January 2007, Apple changed its corporate name from "Apple Computer, Inc." to "Apple Inc." This change has been reflected below and copyright years updated, but no other changes have been made to the APSL 2.0.

1. General; Definitions. This License applies to any program or other work which Apple Inc. ("Apple") makes publicly available and which contains a notice placed by Apple identifying such program or work as "Original Code" and stating that it is subject to the terms of this Apple Public Source License version 2.0 ("License"). As used in this License:

1.1 "Applicable Patent Rights" mean: (a) in the case where Apple is the grantor of rights, (i) claims of patents that are now or hereafter acquired, owned by or assigned to Apple and (ii) that cover subject matter contained in the Original Code, but only to the extent necessary to use, reproduce and/or distribute the Original Code without infringement; and (b) in the case where You are the grantor of rights, (i) claims of patents that are now or hereafter acquired, owned by or assigned to You and (ii) that cover subject matter in Your Modifications, taken alone or in combination with Original Code.

1.2 "Contributor" means any person or entity that creates or contributes to the creation of Modifications.

1.3 "Covered Code" means the Original Code, Modifications, the combination of Original Code and any Modifications, and/or any respective portions thereof.

1.4 "Externally Deploy" means: (a) to sublicense, distribute or otherwise make Covered Code available, directly or indirectly, to anyone other than You; and/or (b) to use Covered Code, alone or as part of a Larger Work, in any way to provide a service, including but not limited to delivery of content, through electronic communication with a client other than You.

1.5 "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.6 "Modifications" mean any addition to, deletion from, and/or change to, the substance and/or structure of the Original Code, any previous Modifications, the combination of Original Code and any previous Modifications, and/or any respective portions thereof. When code is released as a series of files, a Modification is: (a) any addition to or deletion from the contents of a file containing Covered Code; and/or (b) any new file or other representation of computer program statements that contains any part of Covered Code.

1.7 "Original Code" means (a) the Source Code of a program or other work as originally made available by Apple under this License, including the Source
Code of any updates or upgrades to such programs or works made available by Apple under this License, and that has been expressly identified by Apple as such in the header file(s) of such work; and (b) the object code compiled from such Source Code and originally made available by Apple under this License.

1.8 “Source Code” means the human readable form of a program or other work that is suitable for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an executable (object code).

1.9 “You” or "Your" means an individual or a legal entity exercising rights under this License. For legal entities, "You" or "Your" includes any entity which controls, is controlled by, or is under common control with, You, where "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

2. Permitted Uses; Conditions & Restrictions. Subject to the terms and conditions of this License, Apple hereby grants You, effective on the date You accept this License and download the Original Code, a world-wide, royalty-free, non-exclusive license, to the extent of Apple's Applicable Patent Rights and copyrights covering the Original Code, to do the following:

2.1 Unmodified Code. You may use, reproduce, display, perform, internally distribute within Your organization, and Externally Deploy verbatim, unmodified copies of the Original Code, for commercial or non-commercial purposes, provided that in each instance:

(a) You must retain and reproduce in all copies of Original Code the copyright and other proprietary notices and disclaimers of Apple as they appear in the Original Code, and keep intact all notices in the Original Code that refer to this License; and

(b) You must include a copy of this License with every copy of Source Code of Covered Code and documentation You distribute or Externally Deploy, and You may not offer or impose any terms on such Source Code that alter or restrict this License or the recipients' rights hereunder, except as permitted under Section 6.

2.2 Modified Code. You may modify Covered Code and use, reproduce, display, perform, internally distribute within Your organization, and Externally Deploy Your Modifications and Covered Code, for commercial or non-commercial purposes, provided that in each instance You also meet all of these conditions:

(a) You must satisfy all the conditions of Section 2.1 with respect to the Source Code of the Covered Code;
(b) You must duplicate, to the extent it does not already exist, the notice in Exhibit A in each file of the Source Code of all Your Modifications, and cause the modified files to carry prominent notices stating that You changed the files and the date of any change; and

(c) If You Externally Deploy Your Modifications, You must make Source Code of all Your Externally Deployed Modifications either available to those to whom You have Externally Deployed Your Modifications, or publicly available. Source Code of Your Externally Deployed Modifications must be released under the terms set forth in this License, including the license grants set forth in Section 3 below, for as long as you Externally Deploy the Covered Code or twelve (12) months from the date of initial External Deployment, whichever is longer. You should preferably distribute the Source Code of Your Externally Deployed Modifications electronically (e.g. download from a web site).

2.3 Distribution of Executable Versions. In addition, if You Externally Deploy Covered Code (Original Code and/or Modifications) in object code, executable form only, You must include a prominent notice, in the code itself as well as in related documentation, stating that Source Code of the Covered Code is available under the terms of this License with information on how and where to obtain such Source Code.

2.4 Third Party Rights. You expressly acknowledge and agree that although Apple and each Contributor grants the licenses to their respective portions of the Covered Code set forth herein, no assurances are provided by Apple or any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Apple and each Contributor disclaim any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow You to distribute the Covered Code, it is Your responsibility to acquire that license before distributing the Covered Code.

3. Your Grants. In consideration of, and as a condition to, the licenses granted to You under this License, You hereby grant to any person or entity receiving or distributing Covered Code under this License a non-exclusive, royalty-free, perpetual, irrevocable license, under Your Applicable Patent Rights and other intellectual property rights (other than patent) owned or controlled by You, to use, reproduce, display, perform, modify, sublicense, distribute and Externally Deploy Your Modifications of the same scope and extent as Apple's licenses under Sections 2.1 and 2.2 above.

4. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In each such instance, You must make sure the
requirements of this License are fulfilled for the Covered Code or any portion thereof.

5. Limitations on Patent License. Except as expressly stated in Section 2, no other patent rights, express or implied, are granted by Apple herein. Modifications and/or Larger Works may require additional patent licenses from Apple which Apple may grant in its sole discretion.

6. Additional Terms. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations and/or other rights consistent with the scope of the license granted herein (“Additional Terms”) to one or more recipients of Covered Code. However, You may do so only on Your own behalf and as Your sole responsibility, and not on behalf of Apple or any Contributor. You must obtain the recipient’s agreement that any such Additional Terms are offered by You alone, and You hereby agree to indemnify, defend and hold Apple and every Contributor harmless for any liability incurred by or claims asserted against Apple or such Contributor by reason of any such Additional Terms.

7. Versions of the License. Apple may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Once Original Code has been published under a particular version of this License, You may continue to use it under the terms of that version. You may also choose to use such Original Code under the terms of any subsequent version of this License published by Apple. No one other than Apple has the right to modify the terms applicable to Covered Code created under this License.

8. NO WARRANTY OR SUPPORT. The Covered Code may contain in whole or in part pre-release, untested, or not fully tested works. The Covered Code may contain errors that could cause failures or loss of data, and may be incomplete or contain inaccuracies. You expressly acknowledge and agree that use of the Covered Code, or any portion thereof, is at Your sole and entire risk. THE COVERED CODE IS PROVIDED "AS IS" AND WITHOUT WARRANTY, UPGRADES OR SUPPORT OF ANY KIND AND APPLE AND APPLE'S LICENSOR(S) (COLLECTIVELY REFERRED TO AS "APPLE" FOR THE PURPOSES OF SECTIONS 8 AND 9) AND ALL CONTRIBUTORS EXPRESSLY DISCLAIM ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY, OF SATISFACTORY QUALITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY, OF QUIET ENJOYMENT, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. APPLE AND EACH CONTRIBUTOR DOES NOT WARRANT AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE COVERED CODE, THAT THE FUNCTIONS CONTAINED IN THE COVERED CODE WILL MEET YOUR REQUIREMENTS, THAT THE OPERATION OF THE COVERED CODE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE COVERED CODE WILL BE CORRECTED. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY APPLE, AN APPLE AUTHORIZED REPRESENTATIVE OR ANY CONTRIBUTOR SHALL CREATE A WARRANTY. You acknowledge that the Covered Code is not intended for use in the operation of nuclear facilities, aircraft navigation, communication systems, or air traffic control.
machines in which case the failure of the Covered Code could lead to death, personal injury, or severe physical or environmental damage.

9. LIMITATION OF LIABILITY. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT SHALL APPLE OR ANY CONTRIBUTOR BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE OR YOUR USE OR INABILITY TO USE THE COVERED CODE, OR ANY PORTION THEREOF, WHETHER UNDER A THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCTS LIABILITY OR OTHERWISE, EVEN IF APPLE OR SUCH CONTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY REMEDY. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OF LIABILITY OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION MAY NOT APPLY TO YOU. In no event shall Apple's total liability to You for all damages (other than as may be required by applicable law) under this License exceed the amount of fifty dollars ($50.00).

10. Trademarks. This License does not grant any rights to use the trademarks or trade names "Apple", "Mac", "Mac OS", "QuickTime", "QuickTime Streaming Server" or any other trademarks, service marks, logos or trade names belonging to Apple (collectively "Apple Marks") or to any trademark, service mark, logo or trade name belonging to any Contributor. You agree not to use any Apple Marks in or as part of the name of products derived from the Original Code or to endorse or promote products derived from the Original Code other than as expressly permitted by and in strict compliance at all times with Apple's third party trademark usage guidelines which are posted at http://www.apple.com/legal/guidelinesfor3rdparties.html.

11. Ownership. Subject to the licenses granted under this License, each Contributor retains all rights, title and interest in and to any Modifications made by such Contributor. Apple retains all rights, title and interest in and to the Original Code and any Modifications made by or on behalf of Apple ("Apple Modifications"), and such Apple Modifications will not be automatically subject to this License. Apple may, at its sole discretion, choose to license such Apple Modifications under this License, or on different terms from those contained in this License or may choose not to license them at all.

12. Termination.

12.1 Termination. This License and the rights granted hereunder will terminate:

(a) automatically without notice from Apple if You fail to comply with any term(s) of this License and fail to cure such breach within 30 days of becoming aware of such breach;
(b) immediately in the event of the circumstances described in Section 13.5(b); or
(c) automatically without notice from Apple if You, at any time during the term of this License, commence an action for patent infringement against Apple;
providing that Apple did not first commence an action for patent infringement against You in that instance.

12.2 Effect of Termination. Upon termination, You agree to immediately stop any further use, reproduction, modification, sublicensing and distribution of the Covered Code. All sublicenses to the Covered Code which have been properly granted prior to termination shall survive any termination of this License. Provisions which, by their nature, should remain in effect beyond the termination of this License shall survive, including but not limited to Sections 3, 5, 8, 9, 10, 11, 12.2 and 13. No party will be liable to any other for compensation, indemnity or damages of any sort solely as a result of terminating this License in accordance with its terms, and termination of this License will be without prejudice to any other right or remedy of any party.


13.1 Government End Users. The Covered Code is a "commercial item" as defined in FAR 2.101. Government software and technical data rights in the Covered Code include only those rights customarily provided to the public as defined in this License. This customary commercial license in technical data and software is provided in accordance with FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for Department of Defense purchases, DFAR 252.227-7015 (Technical Data -- Commercial Items) and 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation). Accordingly, all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

13.2 Relationship of Parties. This License will not be construed as creating an agency, partnership, joint venture or any other form of legal association between or among You, Apple or any Contributor, and You will not represent to the contrary, whether expressly, by implication, appearance or otherwise.

13.3 Independent Development. Nothing in this License will impair Apple's right to acquire, license, develop, have others develop for it, market and/or distribute technology or products that perform the same or similar functions as, or otherwise compete with, Modifications, Larger Works, technology or products that You may develop, produce, market or distribute.

13.4 Waiver; Construction. Failure by Apple or any Contributor to enforce any provision of this License will not be deemed a waiver of future enforcement of that or any other provision. Any law or regulation which provides that the language of a contract shall be construed against the drafter will not apply to this License.

13.5 Severability. (a) If for any reason a court of competent jurisdiction finds any provision of this License, or portion thereof, to be unenforceable, that provision of the License will be enforced to the maximum extent
permissible so as to effect the economic benefits and intent of the parties, and the remainder of this License will continue in full force and effect. (b) Notwithstanding the foregoing, if applicable law prohibits or restricts You from fully and/or specifically complying with Sections 2 and/or 3 or prevents the enforceability of either of those Sections, this License will immediately terminate and You must immediately discontinue any use of the Covered Code and destroy all copies of it that are in your possession or control.

13.6 Dispute Resolution. Any litigation or other dispute resolution between You and Apple relating to this License shall take place in the Northern District of California, and You and Apple hereby consent to the personal jurisdiction of, and venue in, the state and federal courts within that District with respect to this License. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded.

13.7 Entire Agreement; Governing Law. This License constitutes the entire agreement between the parties with respect to the subject matter hereof. This License shall be governed by the laws of the United States and the State of California, except that body of California law concerning conflicts of law.

Where You are located in the province of Quebec, Canada, the following clause applies: The parties hereby confirm that they have requested that this License and all related documents be drafted in English. Les parties ont exig que le present contrat et tous les documents connexes soient rdigs en anglais.

EXHIBIT A.

"Portions Copyright (c) 1999-2007 Apple Inc. All Rights Reserved.

This file contains Original Code and/or Modifications of Original Code as defined in and that are subject to the Apple Public Source License Version 2.0 (the 'License'). You may not use this file except in compliance with the License. Please obtain a copy of the License at http://www.opensource.apple.com/apsl/ and read it before using this file.

The Original Code and all software distributed under the License are distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES, INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT. Please see the License for the specific language governing rights and limitations under the License."

Mozilla Public License Version 2.0

==================================

1. Definitions

--------------
1.1. "Contributor"
means each individual or legal entity that creates, contributes to
the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached
the notice in Exhibit A, the Executable Form of such Source Code
Form, and Modifications of such Source Code Form, in each case
including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means
(a) that the initial Contributor has attached the notice described
in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of
version 1.1 or earlier of the License, but not also under the
terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in
a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible,
whether at the time of the initial grant or subsequently, any and
all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to,
deletion from, or modification of the contents of Covered
Software; or
(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions
--------------------------------

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.
2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.
2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

-------------------

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices
You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after
Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.
8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to
create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice
---------------------------------------------------------------

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice
---------------------------------------------------------------

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

The MIT License (MIT)

Copyright (c) 2014 Simon Boudrias

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER 
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, 
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE 
SOFTWARE.

****************************************************************************
**
** Copyright (C) YYYY The Qt Company Ltd.
** Contact: https://www.qt.io/licensing/
**
** This file is part of the FOO module of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:LGPL$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms
** and conditions see https://www.qt.io/terms-conditions. For further
** information use the contact form at https://www.qt.io/contact-us.
**
** $QT_END_LICENSE$
** GNU Lesser General Public License Usage
** Alternatively, this file may be used under the terms of the GNU Lesser
** General Public License version 3 as published by the Free Software
** Foundation and appearing in the file LICENSE.LGPL3 included in the
** packaging of this file. Please review the following information to
** ensure the GNU Lesser General Public License version 3 requirements
** will be met: https://www.gnu.org/licenses/lgpl-3.0.html.
**
** GNU General Public License Usage
** Alternatively, this file may be used under the terms of the GNU
** General Public License version 2.0 or (at your option) the GNU General
** Public license version 3 or any later version approved by the KDE Free
** Qt Foundation. The licenses are as published by the Free Software
** Foundation and appearing in the file LICENSE.GPL2 and LICENSE.GPL3
** included in the packaging of this file. Please review the following
** information to ensure the GNU General Public License requirements will
** be met: https://www.gnu.org/licenses/gpl-2.0.html and
** https://www.gnu.org/licenses/gpl-3.0.html.
**
** $QT_END_LICENSE$
**
****************************************************************************/

All MurmurHash source files are placed in the public domain.

The license below applies to all other code in SMHasher:

Copyright (c) 2011 Google, Inc.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2017, Mozilla
Copyright (c) 2007-2017, Jean-Marc Valin
Copyright (c) 2005-2017, Xiph.Org Foundation
Copyright (c) 2003-2004, Mark Borgerding

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Xiph.Org Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
(The MIT License)

Copyright (c) 2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2011-2017 JP Richardson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files
(the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy,
modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the
Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS
OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

Apache License

Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor
for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other
commercial damages or losses), even if such Contributor has been advised
of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the
Work or Derivative Works thereof, You may choose to offer, and charge a
fee for, acceptance of support, warranty, indemnity, or other liability
obligations and/or rights consistent with this License. However, in accepting
such obligations, You may act only on Your own behalf and on Your sole
responsibility, not on behalf of any other Contributor, and only if You
agree to indemnify, defend, and hold each Contributor harmless for any
liability incurred by, or claims asserted against, such Contributor by
reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
# MIT LICENSED Copyright (c) 2013 Arnout Kazemier (http://3rd-Eden.com)
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
# THE SOFTWARE.
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

GNU Free Documentation License
Version 1.3, 3 November 2008

<http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of "copyleft", which means that derivative
works of the document must themselves be free in the same sense. It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does. But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book. We recommend this License
principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that
contains a notice placed by the copyright holder saying it can be
distributed under the terms of this License. Such a notice grants a
A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".
Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.
You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified
Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retile any existing section to be Entitled "Endorsements"
or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections orappendices that qualify as Secondary Sections and contain no materialcopied from the Document, you may at your option designate some or allof these sections as invariant. To do this, add their titles to thelist of Invariant Sections in the Modified Version's license notice.theses titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it containsnothing but endorsements of your Modified Version by variousparties--for example, statements of peer review or that the text hasbeen approved by an organization as the authoritative definition of astandard.

You may add a passage of up to five words as a Front-Cover Text, and apassage of up to 25 words as a Back-Cover Text, to the end of the list ofCover Texts in the Modified Version. Only one passage ofFront-Cover Text and one of Back-Cover Text may be added by (orthrough arrangements made by) any one entity. If the Document alreadyincludes a cover text for the same cover, previously added by you orby arrangement made by the same entity you are acting on behalf of,you may not add another; but you may replace the old one, on explicitpermission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this Licensegive permission to use their names for publicity for or to assert orimply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under thisLicense, under the terms defined in section 4 above for modifiedversions, provided that you include in the combination all of theInvariant Sections of all of the original documents, unmodified, andlist them all as Invariant Sections of your combined work in itslicense notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, andmultiple identical Invariant Sections may be replaced with a singlecopy. If there are multiple Invariant Sections with the same name butdifferent contents, make the title of each such section unique byadding at the end of it, in parentheses, the name of the originalauthor or publisher of that section if known, or else a unique number.Make the same adjustment to the section titles in the list of
Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may
distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE
The Free Software Foundation may publish new, revised versions of the
GNU Free Documentation License from time to time. Such new versions
will be similar in spirit to the present version, but may differ in
detail to address new problems or concerns. See

Each version of the License is given a distinguishing version number.
If the Document specifies that a particular numbered version of this
License "or any later version" applies to it, you have the option of
following the terms and conditions either of that specified version or
of any later version that has been published (not as a draft) by the
Free Software Foundation. If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation. If the Document
specifies that a proxy can decide which future versions of this
License can be used, that proxy's public statement of acceptance of a
version permanently authorizes you to choose that version for the
Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any
World Wide Web server that publishes copyrightable works and also
provides prominent facilities for anybody to edit those works. A
public wiki that anybody can edit is an example of such a server. A
"Massive Multiauthor Collaboration" (or "MMC") contained in the site
means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0
license published by Creative Commons Corporation, a not-for-profit
corporation with a principal place of business in San Francisco,
California, as well as future copyleft versions of that license
published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in
part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this
License, and if all works that were first published under this License
somewhere other than this MMC, and subsequently incorporated in whole or
in part into the MMC, (1) had no cover texts or invariant sections, and
(2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site
under CC-BY-SA on the same site at any time before August 1, 2009,
provided the MMC is eligible for relicensing.
ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

MIT License

Copyright (c) 2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Copyright (c) 2013 Julien Pommier (pommier@modartt.com)

Based on original fortran 77 code from FFTPACKv4 from NETLIB, authored by Dr Paul Swarztrauber of NCAR, in 1985.

As confirmed by the NCAR fftpack software curators, the following FFTPACKv5 license applies to FFTPACKv4 sources. My changes are released under the same terms.

FFTPACK license:

http://www.cisl.ucar.edu/css/software/fftpack5/ftp.html

Copyright (c) 2004 the University Corporation for Atmospheric Research ("UCAR"). All rights reserved. Developed by NCAR's Computational and Information Systems Laboratory, UCAR, www.cisl.ucar.edu.

Redistribution and use of the Software in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

- Neither the names of NCAR's Computational and Information Systems Laboratory, the University Corporation for Atmospheric Research, nor the names of its sponsors or contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

- Redistributions of source code must retain the above copyright notices, this list of conditions, and the disclaimer below.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer below in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

The MIT License (MIT)
Copyright (C) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS
"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as
distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS
Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

The FreeType 2 font engine is copyrighted work and cannot be used legally without a software license. In order to make this project usable to a vast majority of developers, we distribute it under two mutually exclusive open-source licenses.

This means that *you* must choose *one* of the two licenses described below, then obey all its terms and conditions when using FreeType 2 in any of your projects or products.

- The FreeType License, found in the file `FTL.TXT', which is similar to the original BSD license *with* an advertising clause that forces you to explicitly cite the FreeType project in your product's documentation. All details are in the license file. This license is suited to products which don't use the GNU General Public License.

  Note that this license is compatible to the GNU General Public License version 3, but not version 2.

- The GNU General Public License version 2, found in `GPLv2.TXT' (any later version can be used also), for programs which already use the GPL. Note that the FTL is incompatible with GPLv2 due to its advertisement clause.

The contributed BDF and PCF drivers come with a license similar to that of the X Window System. It is compatible to the above two licenses (see file src/bdf/README and src/pcf/README). The same holds for the files `fthash.c' and `fthash.h'; their code was part of the BDF driver in earlier FreeType versions.

The gzip module uses the zlib license (see src/gzip/zlib.h) which too is compatible to the above two licenses.

The MD5 checksum support (only used for debugging in development builds) is in the public domain.

--- end of LICENSE.TXT ---

Open Source Used In DNAC Platform Antelope 11849
/*
 * SpanDSP - a series of DSP components for telephony
 * 
 * g711.h - In line A-law and u-law conversion routines
 * 
 * Written by Steve Underwood <steveu@coppice.org>
 * 
 * Copyright (C) 2001 Steve Underwood
 * 
 * Despite my general liking of the GPL, I place this code in the
 * public domain for the benefit of all mankind - even the slimy
 * ones who might try to proprietize my work and use it to my
 * detriment.
 */

Copyright (c) 2006, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above
    copyright notice, this list of conditions and the following disclaimer
    in the documentation and/or other materials provided with the
    distribution.
  * Neither the name of Google Inc. nor the names of its
    contributors may be used to endorse or promote products derived from
    this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

--------------------------------------------------------------------

Copyright 2001-2004 Unicode, Inc.

Disclaimer
This source code is provided as is by Unicode, Inc. No claims are made as to fitness for any particular purpose. No warranties of any kind are expressed or implied. The recipient agrees to determine applicability of information provided. If this file has been purchased on magnetic or optical media from Unicode, Inc., the sole remedy for any claim will be exchange of defective media within 90 days of receipt.

Limitations on Rights to Redistribute This Code

Unicode, Inc. hereby grants the right to freely use the information supplied in this file in the creation of products supporting the Unicode Standard, and to make copies of this file in any form for internal or external distribution as long as this notice remains attached.

# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# https://developers.google.com/open-source/clia/individual
# https://developers.google.com/open-source/clia/corporate
#
# Names should be added to this file as:
# Name <email address>

Mark Mentovai <mark@chromium.org>
Robert Sesek <rsesek@chromium.org>
Scott Graham <scottmg@chromium.org>
Joshua Peraza <jperaza@chromium.org>

SMSLib Sudden Motion Sensor Access Library
Copyright (c) 2010 Suitable Systems
All rights reserved.

Developed by: Daniel Griscom
Suitable Systems
http://www.suitable.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimers.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

- Neither the names of Suitable Systems nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

For more information about SMSLib, see <http://www.suitable.com/tools/smslib.html> or contact

Daniel Griscom
Suitable Systems
1 Centre Street, Suite 204
Wakefield, MA 01880
(781) 665-0053

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2019 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in https://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale,
use or other dealings in these Data Files or Software without prior
written authorization of the copyright holder.

---------------------

Third-Party Software Licenses

This section contains third-party software notices and/or additional
terms for licensed third-party software components included within ICU
libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies of
the Software and that both the above copyright notice(s) and this
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.


# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
#   - https://sourceforge.net/project/?group_id=1519
#   - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#   - Its license terms and conditions are shown below.
#
# --------COPYING.libtabe ---- BEGIN-----------------
#
# /*
# * Copyright (c) 1999 TaBE Project.
# * Copyright (c) 1999 Pai-Hsiang Hsiao.
# * All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * . Neither the name of the TaBE Project nor the names of its
# * contributors may be used to endorse or promote products derived
# * from this software without specific prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# /*
# * Copyright (c) 1999 Computer Systems and Communication Lab,
# * Institute of Information Science, Academia
# * Sinica. All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * Neither the name of the Computer Systems and Communication Lab
# * nor the names of its contributors may be used to endorse or
# * promote products derived from this software without specific
# * prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */

# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-t sai4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-t sai4
#
# ----------COPYING.libtable----END-------------------------------
#
#
# ----------COPYING.ipadic----BEGIN-------------------------------
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise. The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
# project or thereafter. Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# ---------------COPYING.ipadic-----END----------------------------------

3. Lao Word Break Dictionary Data (laodict.txt)

# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
# (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# -----------------------------------------------
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
# 
# # Redistributions of source code must retain the above copyright notice, this
# # list of conditions and the following disclaimer. Redistributions in
# # binary form must reproduce the above copyright notice, this list of
# # conditions and the following disclaimer in the documentation and/or
# # other materials provided with the distribution.
# # 
# # THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
#

4. Burmese Word Break Dictionary Data (burmesedict.txt)

# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
# github.com/kanyawtech/myanmar-karen-word-lists
#
# -----------------------------------------------
# Copyright (c) 2013, LeRoy Benjamin Sharon
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
disclaimer. Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
# with the distribution.
#
# Neither the name Myanmar Karen Word Lists, nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
# BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
# TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

# 7. Database Ownership

# The TZ database itself is not an IETF Contribution or an IETF document. Rather it is a pre-existing and regularly updated work that is in the public domain, and is intended to remain in the public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do not apply to the TZ Database or contributions that individuals make to it. Should any claims be made and substantiated against the TZ Database, the organization that is providing the IANA Considerations defined in this RFC, under the memorandum of understanding with the IETF, currently ICANN, may act in accordance with all competent court orders. No ownership claims will be made by ICANN or the IETF Trust on the database or the code. Any person making a contribution to the database or code waives all rights to future claims in that contribution or in the TZ Database.

6. Google double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The FreeType 2 font engine is copyrighted work and cannot be used
legally without a software license. In order to make this project
usable to a vast majority of developers, we distribute it under two
mutually exclusive open-source licenses.

This means that *you* must choose *one* of the two licenses described
below, then obey all its terms and conditions when using FreeType 2 in
any of your projects or products.

- The FreeType License, found in the file `FTL.TXT', which is similar
to the original BSD license *with* an advertising clause that forces
you to explicitly cite the FreeType project in your product's
documentation. All details are in the license file. This license
is suited to products which don't use the GNU General Public License.

  Note that this license is compatible to the GNU General Public
License version 3, but not version 2.

- The GNU General Public License version 2, found in `GPLv2.TXT' (any
later version can be used also), for programs which already use the
GPL. Note that the FTL is incompatible with GPLv2 due to its
advertisement clause.

The contributed BDF and PCF drivers come with a license similar to that
of the X Window System. It is compatible to the above two licenses (see
file src/bdf/README and src/pcf/README).

The gzip module uses the zlib license (see src/gzip/zlib.h) which too is
compatible to the above two licenses.

The MD5 checksum support (only used for debugging in development builds)
is in the public domain.

--- FDL.TXT ---

The FreeType Project LICENSE
----------------------------
Introduction
============

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- We don’t promise that this software works. However, we will be interested in any kind of bug reports. (‘as is’ distribution)

- You can use this software for whatever you want, in parts or full form, without having to pay us. (‘royalty-free’ usage)

- You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (‘credits’)

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

"""

Portions of this software are copyright <year> The FreeType
Legal Terms
==============

0. Definitions

Throughout this license, the terms `package', `FreeType Project', and `FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project', be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

Please replace <year> with the value from the FreeType version you actually use.
This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- Redistribution of source code must retain this license file (`FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.

- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: `FreeType Project', `FreeType Engine', `FreeType library', or `FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts
There are two mailing lists related to FreeType:

- freetype@nongnu.org
  
  Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- freetype-devel@nongnu.org
  
  Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

http://www.freetype.org

--- end of FDL.TXT ---

--- GPLv2.TXT ---

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.
If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

--- end of GPLv2.TXT ---

Font license info

## Font Awesome

Copyright (C) 2016 by Dave Gandy

Author:    Dave Gandy
License:    SIL ()
Homepage:   http://fortawesome.github.com/Font-Awesome/
Copyright (c) 2015, Google Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

====================
Open Source Licenses
====================

This software may use portions of the following libraries subject to the accompanying licenses:

******************************************************************************
chromium_audio
******************************************************************************

// Copyright 2014 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

curl

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2014, Daniel Stenberg. <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.
Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

****************************

dynamic_annotations

****************************

Copyright (c) 2008-2009, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

****************************
eigen3

****************************

Eigen is primarily MPL2 licensed. See COPYING.MPL2 and these links:
http://www.mozilla.org/MPL/2.0/
http://www.mozilla.org/MPL/2.0/FAQ.html

Some files contain third-party code under BSD or LGPL licenses, whence the other COPYING.* files here.
All the LGPL code is either LGPL 2.1-only, or LGPL 2.1-or-later. For this reason, the COPYING.LGPL file contains the LGPL 2.1 text.

If you want to guarantee that the Eigen code that you are including is licensed under the MPL2 and possibly more permissive licenses (like BSD), define this preprocessor symbol: EIGEN_MPL2_ONLY

For example, with most compilers, you could add this to your project

    CXXFLAGS: -DEIGEN_MPL2_ONLY

This will cause a compilation error to be generated if you #include any code that is LGPL licensed.

Following applies to:
./test/mapstaticmethods.cpp
./test/schur_real.cpp
./test/prec_inverse_4x4.cpp
./test/smallvectors.cpp
./test/redux.cpp
./test/special_numbers.cpp
./test/adjoint.cpp
./test/resize.cpp
./test/mixingtypes.cpp
./test/product_trmv.cpp
./test/sparse_solvers.cpp
./test/cholesky.cpp
./test/geo_quaternion.cpp
./test/miscmatrices.cpp
./test/stddeque.cpp
./test/integer_types.cpp
./test/product_large.cpp
./test/eigensolver_generic.cpp
./test/householder.cpp
./test/geo_orthomethods.cpp
./test/array_for_matrix.cpp
./test/sparseLM.cpp
./test/upperbidiagonalization.cpp
./test/nomalloc.cpp
./test/packetmath.cpp
./test/jacobisvd.cpp
./test/geo_transformations.cpp
./test/swap.cpp
./test/eigensolver_selfadjoint.cpp
./test/inverse.cpp
./test/product_selfadjoint.cpp
./test/product_trsolve.cpp
./test/product_extra.cpp
./test/sparse_solver.h
./test/mapstride.cpp
Open Source Used In DNAC Platform Antelope 11880

/test/mapped_matrix.cpp
/test/geo_eulerangles.cpp
/test/eigen2support.cpp
/test/denseLM.cpp
/test/stdvector.cpp
/test/nesting_ops.cpp
/test/sparse_permutations.cpp
/test/zerosized.cpp
/test/exceptions.cpp
/test/vectorwiseop.cpp
/test/cwiseop.cpp
/test/basicstuff.cpp
/test/product_trmm.cpp
/test/linearstructure.cpp
/test/sparse_product.cpp
/test/stdvector_overload.cpp
/test/stable_norm.cpp
/test/umeyama.cpp
/test/unalignedcount.cpp
/test/triangular.cpp
/test/product_mmtr.cpp
/test/sparse_basic.cpp
/test/sparse_vector.cpp
/test/meta.cpp
/test/real_qz.cpp
/test/ref.cpp
/test/eigensolver_complex.cpp
/test/cholmod_support.cpp
/test/conjugate_gradient.cpp
/test/sparse.h
/test/simplicial_cholesky.cpp
/test/bicgstab.cpp
/test/dynalloc.cpp
/test/product_notemporary.cpp
/test/geo_hyperplane.cpp
/test/lu.cpp
/test/qr.cpp
/test/hessenberg.cpp
/test/sizeof.cpp
/test/main.h
/test/selfadjoint.cpp
/test/permutationmatrices.cpp
/test/superlu_support.cpp
/test/qtvector.cpp
/test/geo_homogeneous.cpp
/test/determinant.cpp
/test/array_reverse.cpp
/test/unalignedassert.cpp
Open Source Used In DNAC Platform Antelope 11884

Unsupported/Eigen/src/KroneckerProduct/KroneckerTensorProduct.h
Unsupported/Eigen/src/NonLinearOptimization/LevenbergMarquardt.h
Unsupported/Eigen/src/NonLinearOptimization/HybridNonLinearSolver.h
Unsupported/Eigen/src/BVH/BVAlgorithms.h
Unsupported/Eigen/src/BVH/KdBVH.h
Unsupported/Eigen/src/AutoDiff/AutoDiffScalar.h
Unsupported/Eigen/src/AutoDiff/AutoDiffJacobian.h
Unsupported/Eigen/src/AutoDiff/AutoDiffVector.h
Unsupported/Eigen/src/Splines/Spline.h
Unsupported/Eigen/src/Splines/SplineFitting.h
Unsupported/Eigen/src/SVD/JacobiSVD.h
Unsupported/Eigen/src/SVD/BDCSVD.h
Unsupported/Eigen/src/SVD/SVDBase.h
Unsupported/Eigen/src/MatrixFunctions/MatrixFunction.h
Unsupported/Eigen/src/MatrixFunctions/MatrixSquareRoot.h
Unsupported/Eigen/src/MatrixFunctions/MatrixLogarithm.h
Unsupported/Eigen/src/MatrixFunctions/StemFunction.h
Unsupported/Eigen/src/MatrixFunctions/MatrixPower.h
Unsupported/Eigen/src/MatrixFunctions/MatrixExponential.h
Unsupported/Eigen/src/MatrixFunctions/MatrixFunctionAtomic.h
Unsupported/Eigen/src/MoreVectorization/MathFunctions.h
Unsupported/Eigen/src/LevenbergMarquardt/LevenbergMarquardt.h
Unsupported/Eigen/src/FFT/ei_fftw_impl.h
Unsupported/Eigen/src/FFT/ei_kissfft_impl.h
 Unsupported/Eigen/src/Polynomials/PolynomialSolver.h
Unsupported/Eigen/src/Polynomials/Companion.h
Unsupported/Eigen/src/Polynomials/PolynomialUtils.h
Unsupported/Eigen/src/NumericalDiff/NumericalDiff.h
Unsupported/Eigen/src/Skyline/SkylineProduct.h
Unsupported/Eigen/src/Skyline/SkylineMatrixBase.h
Unsupported/Eigen/src/Skyline/SkylineStorage.h
Unsupported/Eigen/src/Skyline/SkylineUtil.h
Unsupported/Eigen/src/Skyline/SkylineInplaceLU.h
Unsupported/Eigen/src/Skyline/SkylineMatrix.h
Unsupported/Eigen/SparseExtra
Unsupported/Eigen/AdolcForward
Unsupported/Eigen/KroneckerProduct
Unsupported/Eigen/NonLinearOptimization
Unsupported/Eigen/BVH
Unsupported/Eigen/OpenGLSupport
Unsupported/Eigen/ArpackSupport
Unsupported/Eigen/AutoDiff
Unsupported/Eigen/Splines
Unsupported/Eigen/MPRealSupport
Unsupported/Eigen/MatrixFunctions
Unsupported/Eigen/MoreVectorization
Unsupported/Eigen/LevenbergMarquardt
/unsupported/Eigen/AlignedVector3
/unsupported/Eigen/FFT
/unsupported/Eigen/Polynomials
/unsupported/Eigen/NumericalDiff
/unsupported/Eigen/Skyline
/COPYING_README
/COPYING_README
/LICENSE
/LICENSE
/LICENSE
/Eigen/Eigen2Support
/Eigen/src/Eigen2Support/VectorBlock.h
/Eigen/src/Eigen2Support/Cwise.h
/Eigen/src/Eigen2Support/Minor.h
/Eigen/src/Eigen2Support/Lazy.h
/Eigen/src/Eigen2Support/Memory.h
/Eigen/src/Eigen2Support/MathFunctions.h
/Eigen/src/Eigen2Support/Geometry/AlignedBox.h
/Eigen/src/Eigen2Support/Geometry/Hyperplane.h
/Eigen/src/Eigen2Support/Geometry/Quaternion.h
/Eigen/src/Eigen2Support/Geometry/Rotation2D.h
/Eigen/src/Eigen2Support/Geometry/ParametrizedLine.h
/Eigen/src/Eigen2Support/Geometry/RotationBase.h
/Eigen/src/Eigen2Support/Geometry/Translation.h
/Eigen/src/Eigen2Support/Geometry/Scaling.h
/Eigen/src/Eigen2Support/Geometry/AngleAxis.h
/Eigen/src/Eigen2Support/Geometry/Transform.h
/Eigen/src/Eigen2Support/TriangularSolver.h
/Eigen/src/Eigen2Support/LU.h
/Eigen/src/Eigen2Support/QR.h
/Eigen/src/Eigen2Support/SVD.h
/Eigen/src/Eigen2Support/Meta.h
/Eigen/src/Eigen2Support/Block.h
/Eigen/src/Eigen2Support/Macros.h
/Eigen/src/Eigen2Support/LeastSquares.h
/Eigen/src/Eigen2Support/CwiseOperators.h
/Eigen/src/Jacobi/Jacobi.h
/Eigen/src/misc/Kernel.h
/Eigen/src/misc/SparseSolve.h
/Eigen/src/misc/Solve.h
/Eigen/src/misc/Image.h
/Eigen/src/SparseCore/SparseColEtree.h
/Eigen/src/SparseCore/SparseTranspose.h
/Eigen/src/SparseCore/SparseUtil.h
/Eigen/src/SparseCore/SparseCwiseBinaryOp.h
/Eigen/src/SparseCore/SparseDiagonalProduct.h
/Eigen/src/SparseCore/SparseProduct.h
/Eigen/src/SparseCore/SparseDot.h
Open Source Used In DNAC Platform Antelope 11887
/Eigen/src/Core/MatrixBase.h
/Eigen/src/Core/DenseStorage.h
/Eigen/src/Core/util/Memory.h
/Eigen/src/Core/util/StaticAssert.h
/Eigen/src/Core/util/BlasUtil.h
/Eigen/src/Core/util/MatrixMapper.h
/Eigen/src/Core/util/XprHelper.h
/Eigen/src/Core/util/ForwardDeclarations.h
/Eigen/src/Core/util/Meta.h
/Eigen/src/Core/util/Macros.h
/Eigen/src/Core/util/Constants.h
/Eigen/src/Core/CwiseNullaryOp.h
/Eigen/src/Core/Block.h
/Eigen/src/Core/GeneralProduct.h
/Eigen/src/Core/CommaInitializer.h
/Eigen/src/Core/ReturnByValue.h
/Eigen/src/Core/Stride.h
/Eigen/src/SPQRSupport/SuiteSparseQRSupport.h
/Eigen/src/SparseLU/SparseLU_column_dfs.h
/Eigen/src/SparseLU/SparseLU_panel_dfs.h
/Eigen/src/SparseLU/SparseLU_relax_snodoe.h
/Eigen/src/SparseLU/SparseLU_panel_bmod.h
/Eigen/src/SparseLU/SparseLU_SupernodalMatrix.h
/Eigen/src/SparseLU/SparseLU_Utils.h
/Eigen/src/SparseLU/SparseLU_gemm_kernel.h
/Eigen/src/SparseLU/SparseLU_kernel_bmod.h
/Eigen/src/SparseLU/SparseLU_pivotL.h
/Eigen/src/SparseLU/SparseLU_Memory.h
/Eigen/src/SparseLU/SparseLU_heap_relax_snodoe.h
/Eigen/src/SparseLU/SparseLUImpl.h
/Eigen/src/SparseLU/SparseLU_copy_to_uCol.h
/Eigen/src/SparseLU/SparseLU_Structs.h
/Eigen/src/SparseLU/SparseLU.h
/Eigen/src/SparseLU/SparseLU_column_bmod.h
/Eigen/src/SparseLU/SparseLU_pruneL.h
/Eigen/src/IterativeLinearSolvers/IncompleteLUT.h
/Eigen/src/IterativeLinearSolvers/BasicPreconditioners.h
/Eigen/src/IterativeLinearSolvers/IterativeSolverBase.h
/Eigen/src/IterativeLinearSolvers/ConjugateGradient.h
/Eigen/src/IterativeLinearSolvers/BiCGSTAB.h
/Eigen/src/SparseCholesky/SimplicialCholesky.h
/Eigen/src/Cholesky/LDLT.h
/Eigen/src/Cholesky/LLT.h
/Eigen/src/CholmodSupport/CholmodSupport.h
/Eigen/src/PaStiXSupport/PaStiXSupport.h
/Eigen/src/MetisSupport/MetisSupport.h
/Eigen/StdVector
/Eigen/Core
Mozilla Public License Version 2.0
==================================

1. Definitions
-------------

1.1. "Contributor"
means each individual or legal entity that creates, contributes to
the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached
the notice in Exhibit A, the Executable Form of such Source Code
Form, and Modifications of such Source Code Form, in each case
including portions thereof.
1.5. "Incompatible With Secondary Licenses" means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form" means any form of the work other than Source Code Form.

1.7. "Larger Work" means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License" means this document.

1.9. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications" means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License" means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.
1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions
--------------------------------

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or
(b) for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

-------------------

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code
3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional
disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

*************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*************************************************************************
*                                *  *
* Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.
*                                *

************************************************************************
************************************************************************
************************************************************************

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

************************************************************************

8. Litigation
-------------

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.
9. Miscellaneous
--------------

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License
---------------------------

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice
---------------------------------------------

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this
file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

Following applies to:
./doc/UsingIntelMKL.dox
./doc/UsingIntelMKL.dox
./Eigen/src/Eigenvalues/ComplexSchur_MKL.h
./Eigen/src/Eigenvalues/ComplexSchur_MKL.h
./Eigen/src/Eigenvalues/SelfAdjointEigenSolver_MKL.h
./Eigen/src/Eigenvalues/SelfAdjointEigenSolver_MKL.h
./Eigen/src/Eigenvalues/RealSchur_MKL.h
./Eigen/src/Eigenvalues/RealSchur_MKL.h
./Eigen/src/LU/arch/Inverse_SSE.h
./Eigen/src/LU/arch/Inverse_SSE.h
./Eigen/src/LU/PartialPivLU_MKL.h
./Eigen/src/LU/PartialPivLU_MKL.h
./Eigen/src/QR/HouseholderQR_MKL.h
./Eigen/src/QR/HouseholderQR_MKL.h
./Eigen/src/QR/ColPivHouseholderQR_MKL.h
./Eigen/src/QR/ColPivHouseholderQR_MKL.h
./Eigen/src/SVD/JacobiSVD_MKL.h
./Eigen/src/SVD/JacobiSVD_MKL.h
./Eigen/src/PardisoSupport/PardisoSupport.h
./Eigen/src/PardisoSupport/PardisoSupport.h
./Eigen/src/Core/Assign_MKL.h
./Eigen/src/Core/Assign_MKL.h
./Eigen/src/Core/products/SelfadjointMatrixVector_MKL.h
./Eigen/src/Core/products/SelfadjointMatrixVector_MKL.h
./Eigen/src/Core/products/GeneralMatrixVector_MKL.h
./Eigen/src/Core/products/GeneralMatrixVector_MKL.h
./Eigen/src/Core/products/SelfadjointMatrixMatrix_MKL.h
./Eigen/src/Core/products/SelfadjointMatrixMatrix_MKL.h
./Eigen/src/Core/products/TriangularMatrixMatrix_MKL.h
./Eigen/src/Core/products/TriangularMatrixMatrix_MKL.h
./Eigen/src/Core/products/GeneralMatrixMatrix_MKL.h

Open Source Used In DNAC Platform Antelope 11898
Copyright (c) 2011, Intel Corporation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. *
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Following applies to:
everything under ./bench/btl

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
The GNU General Public License is a free, copyleft license for software and other kinds of works.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.
To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.
The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other
6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no
charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.
Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions...
of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted...
by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software
Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if
necessary. For more information on this, and how to apply and follow
the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your
program into proprietary programs. If your program is a subroutine
library, you may consider it more useful to permit linking proprietary
applications with the library. If this is what you want to do, use
the GNU Lesser General Public License instead of this License. But

Following applies to:
./test/metis_support.cpp
./test/sparselu.cpp
./unsupported/test/mpreal/mpreal.h
./unsupported/Eigen/src/IterativeSolvers/IterationController.h
./unsupported/Eigen/src/IterativeSolvers/ConstrainedConjGrad.h
./unsupported/Eigen/src/Eigenvalues/ArpackSelfAdjointEigenSolver.h
./Eigen/src/OrderingMethods/Amd.h
./Eigen/src/SparseCholesky/SimplicialCholesky_impl.h

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the
GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.
A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

   a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

   b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

   a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany...
the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Following applies to:
./unsupported/Eigen/src/LevenbergMarquardt/LevenbergMarquardt.h
Minpack Copyright Notice (1999) University of Chicago. All rights reserved

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

   "This product includes software developed by the University of Chicago, as Operator of Argonne National Laboratory.

   Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.

5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT
HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF
ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT,
INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF
ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF
PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER
SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT
(INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE,
EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE
POSSIBILITY OF SUCH LOSS OR DAMAGES.

******************************************************************************
freeType2
******************************************************************************
FreeType

Quoth http://freetype.sourceforge.net/license.html:

FreeType comes with two licenses from which you can choose the one
which fits your needs best.

* The FreeType License is the most commonly used one.
  It is a BSD-style license with a credit clause (and thus not
  compatible with the GPL).

* The GNU General Public License (GPL).
  For all projects which use the GPL also or which need a license
  compatible to the GPL.

FTL.TXT:
---

The FreeType Project LICENSE
----------------------------

2006-Jan-27

Copyright 1996-2002, 2006 by
David Turner, Robert Wilhelm, and Werner Lemberg

Introduction
============

The FreeType Project is distributed in several archive packages;
some of them may contain, in addition to the FreeType font engine,
various tools and contributions which rely on, or relate to, the
FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- We don't promise that this software works. However, we will be interested in any kind of bug reports. (‘as is’ distribution)

- You can use this software for whatever you want, in parts or full form, without having to pay us. (‘royalty-free’ usage)

- You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (‘credits’)

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```plaintext
"""
Portions of this software are copyright (C) <year> The FreeType Project (www.freetype.org). All rights reserved.
"""

Please replace <year> with the value from the FreeType version you actually use.

Legal Terms
=============

0. Definitions
-------------
Throughout this license, the terms `package', `FreeType Project', and `FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project', be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty

-------------

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

-------------

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

o Redistribution of source code must retain this license file (`FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered,
original files must be preserved in all copies of source files.

- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising
-------------

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: 'FreeType Project', 'FreeType Engine', 'FreeType library', or 'FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts
------------

There are two mailing lists related to FreeType:

- freetype@nongnu.org

  Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- freetype-devel@nongnu.org
Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

http://www.freetype.org

--- end of FTL.TXT ---

******************************************************************************
GL
******************************************************************************

Mesa Component Licenses

<table>
<thead>
<tr>
<th>Component</th>
<th>Location</th>
<th>Primary Author</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Mesa code</td>
<td>src/mesa/</td>
<td>Brian Paul</td>
<td>Mesa (MIT)</td>
</tr>
<tr>
<td>Device drivers</td>
<td>src/mesa/drivers/*</td>
<td>See drivers</td>
<td>See drivers</td>
</tr>
<tr>
<td>Ext headers</td>
<td>include/GL/glext.h</td>
<td>SGI</td>
<td>SGI Free B</td>
</tr>
<tr>
<td></td>
<td>include/GL/glxext.h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLUT</td>
<td>src/glut/</td>
<td>Mark Kilgard</td>
<td>Mark's copyright</td>
</tr>
<tr>
<td>GLEW</td>
<td>src/glew-1.13.0</td>
<td>Nigel Stewart</td>
<td>Modified BSD</td>
</tr>
<tr>
<td>Mesa GLU library</td>
<td>src/glu/mesa/</td>
<td>Brian Paul</td>
<td>GNU-LGPL</td>
</tr>
<tr>
<td>SGI GLU library</td>
<td>src/glu/sgi/</td>
<td>SGI</td>
<td>SGI Free B</td>
</tr>
<tr>
<td>demo programs</td>
<td>progs/demos/</td>
<td>various</td>
<td>see source files</td>
</tr>
<tr>
<td>X demos</td>
<td>progs/xdemos/</td>
<td>Brian Paul</td>
<td>see source files</td>
</tr>
<tr>
<td>SGI demos</td>
<td>progs/samples/</td>
<td>SGI</td>
<td>SGI copyright</td>
</tr>
<tr>
<td>RedBook demos</td>
<td>progs/redbook/</td>
<td>SGI</td>
<td>SGI copyright</td>
</tr>
</tbody>
</table>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Alan Hourihane not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Alan Hourihane makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

ALAN HOURIHANE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL ALAN HOURIHANE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary
GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don’t assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

**GNU LIBRARY GENERAL PUBLIC LICENSE**
**TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
   a) The modified work must itself be a software library.
   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.
Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work
      based on the Library, uncombined with any other library
      facilities. This must be distributed under the terms of the
      Sections above.

   b) Give prominent notice with the combined library of the fact
      that part of it is a work based on the Library, and explaining
      where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).
To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

---------------------

The OpenGL Extension Wrangler Library
Copyright (C) 2002-2008, Milan Ikits <milan ikits@ieee org>
Copyright (C) 2002-2008, Marcelo E. Magallon <mmagallo@debian org>
Copyright (C) 2002, Lev Povalahev
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.
* The name of the author may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

****************************
gradle
****************************

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensors" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-----------------------------------------------
License for the slf4j package
-----------------------------------------------

SLF4J License

Copyright (c) 2004-2007 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

These terms are identical to those of the MIT License, also called the X License or the X11 License, which is a simple, permissive non-copyleft free software license. It is deemed compatible with virtually all types of licenses, commercial or otherwise. In particular, the Free Software Foundation has declared it compatible with GNU GPL. It is also known to be approved by the Apache Software Foundation as compatible with Apache Software License.

------------------------------------------------------------------------------
License for the JUnit package
------------------------------------------------------------------------------
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.
"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS
A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the
extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such
provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

License for the JCIFS package

JCIFS License
GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.
We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.
In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

**GNU LESSER GENERAL PUBLIC LICENSE**

**TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.
5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.
This library is distributed in the hope that it will be useful, 
but WITHOUT ANY WARRANTY; without even the implied warranty of 
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU 
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public 
License along with this library; if not, write to the Free Software 
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your 
school, if any, to sign a "copyright disclaimer" for the library, if 
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the 
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

*****************************************************************************
icu***************************************************************************** 
ICU

There are two licenses here:
- ICU license
- Unicode Terms of Use

ICU License - ICU 1.8.1 and later
From http://source.icu-project.org/repos/icu/icu/trunk/license.html
X License (old version). For license pedigree see the
ICU FAQ at http://icu-project.org/userguide/icufaq.html

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2006 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

=====================================================================
Unicode Terms of Use, from http://www.unicode.org/copyright.html

For the general privacy policy governing access to this site, see the Unicode Privacy Policy. For trademark usage, see the Unicode Consortium Trademarks and Logo Policy.

Notice to End User: Terms of Use

Carefully read the following legal agreement ("Agreement"). Use or copying of the software and/or codes provided with this agreement (The "Software") constitutes your acceptance of these terms

1. Unicode Copyright.
   1. Copyright 1991-2007 Unicode, Inc. All rights reserved.
   2. Certain documents and files on this website contain a legend indicating that "Modification is permitted." Any person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming to the Unicode Standard, subject to Terms and Conditions herein.
   3. Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files solely for informational purposes in the creation of products supporting the Unicode Standard, subject to the Terms and Conditions herein.
   4. Further specifications of rights and restrictions pertaining to the use of the particular set of data files known as the "Unicode Character Database" can be found in Exhibit 1.
   5. Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions, these are found on
the back of the title page. For the online edition, certain files (such as the PDF files for book chapters and code charts) carry specific restrictions. All other files are covered under these general Terms of Use. To request a permission to reproduce any part of the Unicode Standard, please contact the Unicode Consortium.

6. No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

7. Modification is not permitted with respect to this document. All copies of this document must be verbatim.

2. Restricted Rights Legend. Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

3. Warranties and Disclaimers.

1. This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

2. If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

3. EXCEPT AS PROVIDED IN SECTION C.2, THIS PUBLICATION AND/OR SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.

4. Waiver of Damages. In no event shall Unicode or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

5. Trademarks.

1. Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

2. This site contains product names and corporate names of other companies. All product names and company names and logos mentioned herein are the trademarks or registered trademarks of their respective owners. Other
products and corporate names mentioned herein which are trademarks of a third party are used only for explanation and for the owners' benefit and with no intent to infringe.

3. Use of third party products or information referred to herein is at the user's risk.

6. Miscellaneous.

1. Jurisdiction and Venue. This server is operated from a location in the State of California, United States of America. Unicode makes no representation that the materials are appropriate for use in other locations. If you access this server from other locations, you are responsible for compliance with local laws. This Agreement, all use of this site and any claims and damages resulting from use of this site are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this site shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.

2. Modification by Unicode Unicode shall have the right to modify this Agreement at any time by posting it to this site. The user may not assign any part of this Agreement without Unicode's prior written consent.

3. Taxes. The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.

4. Severability. If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.

5. Entire Agreement. This Agreement constitutes the entire agreement between the parties.

EXHIBIT 1
UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE


NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE
Copyright 1991-2007 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHERWISE TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions. All other trademarks and registered trademarks mentioned herein are the property of their respective owners.

*******************************************************************************
icu
*******************************************************************************

ICU

There are two licenses here:
- ICU license
- Unicode Terms of Use

*******************************************************************************
ICU License - ICU 1.8.1 and later
COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2006 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

Unicode Terms of Use, from http://www.unicode.org/copyright.html

For the general privacy policy governing access to this site, see the Unicode Privacy Policy. For trademark usage, see the Unicode Consortium Trademarks and Logo Policy.

Notice to End User: Terms of Use

Carefully read the following legal agreement ("Agreement"). Use or copying of the software and/or codes provided with this agreement (The "Software") constitutes your acceptance of these terms

1. Unicode Copyright.
   1. Copyright 1991-2007 Unicode, Inc. All rights reserved.
   2. Certain documents and files on this website contain a legend
indicating that "Modification is permitted." Any person is hereby authorized, without fee, to modify such documents and files to create derivative works conforming to the Unicode Standard, subject to Terms and Conditions herein.

3. Any person is hereby authorized, without fee, to view, use, reproduce, and distribute all documents and files solely for informational purposes in the creation of products supporting the Unicode Standard, subject to the Terms and Conditions herein.

4. Further specifications of rights and restrictions pertaining to the use of the particular set of data files known as the "Unicode Character Database" can be found in Exhibit 1.

5. Each version of the Unicode Standard has further specifications of rights and restrictions of use. For the book editions, these are found on the back of the title page. For the online edition, certain files (such as the PDF files for book chapters and code charts) carry specific restrictions. All other files are covered under these general Terms of Use. To request a permission to reproduce any part of the Unicode Standard, please contact the Unicode Consortium.

6. No license is granted to "mirror" the Unicode website where a fee is charged for access to the "mirror" site.

7. Modification is not permitted with respect to this document.

All copies of this document must be verbatim.

2. Restricted Rights Legend. Any technical data or software which is licensed to the United States of America, its agencies and/or instrumentalities under this Agreement is commercial technical data or commercial computer software developed exclusively at private expense as defined in FAR 2.101, or DFARS 252.227-7014 (June 1995), as applicable. For technical data, use, duplication, or disclosure by the Government is subject to restrictions as set forth in DFARS 202.227-7015 Technical Data, Commercial and Items (Nov 1995) and this Agreement. For Software, in accordance with FAR 12-212 or DFARS 227-7202, as applicable, use, duplication or disclosure by the Government is subject to the restrictions set forth in this Agreement.

3. Warranties and Disclaimers.

1. This publication and/or website may include technical or typographical errors or other inaccuracies. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication and/or website. Unicode may make improvements and/or changes in the product(s) and/or program(s) described in this publication and/or website at any time.

2. If this file has been purchased on magnetic or optical media from Unicode, Inc. the sole and exclusive remedy for any claim will be exchange of the defective media within ninety (90) days of original purchase.

3. EXCEPT AS PROVIDED IN SECTION C.2, THIS PUBLICATION AND/OR SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. UNICODE AND ITS LICENSORS ASSUME NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION AND/OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION OR THE UNICODE WEBSITE.
4. Waiver of Damages. In no event shall Unicode or its licensors be liable for any special, incidental, indirect or consequential damages of any kind, or any damages whatsoever, whether or not Unicode was advised of the possibility of the damage, including, without limitation, those resulting from the following: loss of use, data or profits, in connection with the use, modification or distribution of this information or its derivatives.

5. Trademarks.
   1. Unicode and the Unicode logo are registered trademarks of Unicode, Inc.
   2. This site contains product names and corporate names of other companies. All product names and company names and logos mentioned herein are the trademarks or registered trademarks of their respective owners. Other products and corporate names mentioned herein which are trademarks of a third party are used only for explanation and for the owners' benefit and with no intent to infringe.
   3. Use of third party products or information referred to herein is at the user's risk.

6. Miscellaneous.
   1. Jurisdiction and Venue. This server is operated from a location in the State of California, United States of America. Unicode makes no representation that the materials are appropriate for use in other locations. If you access this server from other locations, you are responsible for compliance with local laws. This Agreement, all use of this site and any claims and damages resulting from use of this site are governed solely by the laws of the State of California without regard to any principles which would apply the laws of a different jurisdiction. The user agrees that any disputes regarding this site shall be resolved solely in the courts located in Santa Clara County, California. The user agrees said courts have personal jurisdiction and agree to waive any right to transfer the dispute to any other forum.
   2. Modification by Unicode Unicode shall have the right to modify this Agreement at any time by posting it to this site. The user may not assign any part of this Agreement without Unicode's prior written consent.
   3. Taxes. The user agrees to pay any taxes arising from access to this website or use of the information herein, except for those based on Unicode's net income.
   4. Severability. If any provision of this Agreement is declared invalid or unenforceable, the remaining provisions of this Agreement shall remain in effect.
   5. Entire Agreement. This Agreement constitutes the entire agreement between the parties.

EXHIBIT 1
UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/. Unicode Software includes any source code
published in the Unicode Standard or under the directories
http://www.unicode.org/Public/, http://www.unicode.org/reports/, and
http://www.unicode.org/cldr/data/.

NOTICE TO USER: Carefully read the following legal agreement. BY
DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES
("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND
AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF
YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA
FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2007 Unicode, Inc. All rights reserved. Distributed under
the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a
copy of the Unicode data files and any associated documentation (the "Data
Files") or Unicode software and any associated documentation (the "Software")
to deal in the Data Files or Software without restriction, including without
limitation the rights to use, copy, modify, merge, publish, distribute, and/or
sell copies of the Data Files or Software, and to permit persons to whom the
Data Files or Software are furnished to do so, provided that (a) the above
copyright notice(s) and this permission notice appear with all copies of the
Data Files or Software, (b) both the above copyright notice(s) and this
permission notice appear in associated documentation, and (c) there is clear
notice in each modified Data File or in the Software as well as in the
documentation associated with the Data File(s) or Software that the data or
software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD
PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN
THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR
SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall
not be used in advertising or otherwise to promote the sale, use or other
dealings in these Data Files or Software without prior written authorization
of the copyright holder.

Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be
registered in some jurisdictions. All other trademarks and registered
trademarks mentioned herein are the property of their respective owners.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
   distribution, then any Derivative Works that You distribute must
   include a readable copy of the attribution notices contained
   within such NOTICE file, excluding those notices that do not
   pertain to any part of the Derivative Works, in at least one
   of the following places: within a NOTICE text file distributed
   as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
   within a display generated by the Derivative Works, if and
   wherever such third-party notices normally appear. The contents
   of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
   notices within Derivative Works that You distribute, alongside
   or as an addendum to the NOTICE text from the Work, provided
   that such additional attribution notices cannot be construed
   as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement You may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*******************************************************************************
java/android_libs/protobuf_nano
*******************************************************************************
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
     copyright notice, this list of conditions and the following disclaimer
     in the documentation and/or other materials provided with the
     distribution.
   * Neither the name of Google Inc. nor the names of its
     contributors may be used to endorse or promote products derived from
     this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

******************************
javascript/jquery_ui
******************************
The MIT License (MIT)

Copyright (c) 2015 jQuery Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

******************************
javascript/jquery\v2_0_1
******************************
Copyright 2013 jQuery Foundation and other contributors
http://jquery.com/

https://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt
https://github.com/jquery/sizzle/blob/master/LICENSE

jQuery and Sizzle are released under MIT Licence.

The text is provided below.

MIT License

****
Copyright 2013 jQuery Foundation and other contributors
http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

******************************
javascript/tracing_framework
******************************
Copyright 2012, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

************************************************
java_src/android_libs/exoplayer
************************************************

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*******************************************************************************
java_src/android_libs/protobuf_nano/v2
*******************************************************************************
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

************************
jpeg
************************
(extracted from src/README)

LEGAL ISSUES
============

In plain English:

1. We don't promise that this software works. (But if you find any bugs,
   please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a
   program, you must acknowledge somewhere in your documentation that
   you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied,
with respect to this software, its quality, accuracy, merchantability, or
fitness for a particular purpose. This software is provided "AS IS", and you,
its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this
software (or portions thereof) for any purpose, without fee, subject to these
conditions:
(1) If any part of the source code for this software is distributed, then this
README file must be included, with this copyright and no-warranty notice
unaltered; and any additions, deletions, or changes to the original files
must be clearly indicated in accompanying documentation.
(2) If only executable code is distributed, then the accompanying
The documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the
resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that
"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

********************************
libogg
********************************
Copyright (c) 2002, Xiph.org Foundation

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Xiph.org Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

********************************
libunwind
********************************
Copyright (c) 2002 Hewlett-Packard Co.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

************************
libvorbis
************************
Copyright (c) 2002-2008 Xiph.org Foundation

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Xiph.org Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**************************
libxcb
**************************
Copyright (C) 2001-2006 Bart Massey, Jamey Sharp, and Josh Triplett.
All Rights Reserved.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the
Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall
be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the names of the authors
or their institutions shall not be used in advertising or
otherwise to promote the sale, use or other dealings in this
Software without prior written authorization from the
authors.

**************************
libxml
**************************
Libxml2, an XML C Parser

Except where otherwise noted in the source code (e.g. the files hash.c,
list.c and the trio files, which are covered by a similar licence but
with different Copyright notices) all the files are:

Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

--------------------------------------------------------------------

Copyright (C) 2000,2012 Bjorn Reese and Daniel Veillard.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

Author: breese@users.sourceforge.net

(taken from hash.c)

--------------------------------------------------------------------

Copyright (C) 2000 Gary Pennington and Daniel Veillard.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.
Author: Gary.Pennington@uk.sun.com

(taken from list.c)

--------------------------------------------------------------------

Copyright (C) 1998 Bjorn Reese and Daniel Stenberg.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

(taken from trio.h and trio.c)

--------------------------------------------------------------------

Copyright (C) 2001 Bjorn Reese <breese@users.sourceforge.net>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

(taken from triodef.h, trionan.h, and trionan.c)

--------------------------------------------------------------------

Copyright (C) 2000 Bjorn Reese and Daniel Stenberg.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.
Copyright (C) 2001 Bjorn Reese and Daniel Stenberg.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

(lodpng)

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source.
Open Source Used In DNAC Platform Antelope 11984

distribution.

*******************************
minizip
*******************************
zlib

(extracted from README, except for match.S)

Copyright notice:

(C) 1995-2004 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu

(extracted from match.S, for match.S only)

Copyright (C) 1998, 2007 Brian Raiter <breadbox@muppetlabs.com>

This software is provided 'as-is', without any express or implied warranty. In no event will the author be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

mongoose

Copyright (c) 2004-2013 Sergey Lyubka

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

objective_c/gtm_session_fetcher

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
 incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

********************************************
openctm
Copyright (c) 2009-2010 Marcus Geelnard

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

OpenCV

IMPORTANT: READ BEFORE DOWNLOADING, COPYING, INSTALLING OR USING.

By downloading, copying, installing or using the software you agree to this license. If you do not agree to this license, do not download, install, copy or use the software.

Intel License Agreement
For Open Source Computer Vision Library

Copyright (C) 2000, 2001, Intel Corporation, all rights reserved.
Copyright (C) 2013, OpenCV Foundation, all rights reserved.
Third party copyrights are property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistribution's of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistribution's in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

* The name of Intel Corporation may not be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Intel Corporation or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

******************************
openssl
******************************

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium: https://cla.developers.google.com/clas

Some files from Intel are under yet another license, which is also included underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License
--------------

/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 */
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in
* the documentation and/or other materials provided with the
* distribution.
*
* 3. All advertising materials mentioning features or use of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit (http://www.openssl.org/)
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
*
*===================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
* /

Original SSLeay License
-----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) */
* All rights reserved.
* 
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
* 
* This library is free for commercial and non-commercial use as long as
* the following conditions are heared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
* 
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
* 
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* "This product includes cryptographic software written by
* Eric Young (eay@cryptsoft.com)"
* The word 'cryptographic' can be left out if the rouines from the library
* being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
* 
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.
 *
* Permission to use, copy, modify, and/or distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
* *
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
* OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

Some files from Intel carry the following license:

# Copyright (c) 2012, Intel Corporation
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#
# * Redistributions of source code must retain the above copyright
#   notice, this list of conditions and the following disclaimer.
# *
# * Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in the
#   documentation and/or other materials provided with the
#   distribution.
# *
# * Neither the name of the Intel Corporation nor the names of its
#   contributors may be used to endorse or promote products derived from
#   this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION "AS IS" AND ANY
# EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
# PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
# CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
# PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
# PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

openssl/boringssl

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL
licensing. Files that are completely new have a Google copyright and an ISC
license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium:
https://cla.developers.google.com/clas

Some files from Intel are under yet another license, which is also included
underneath.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the
OpenSSL License and the original SSLeay license apply to the toolkit. See below
for the actual license texts. Actually both licenses are BSD-style Open Source
licenses. In case of any license issues related to OpenSSL please contact
openssl-core@openssl.org.

OpenSSL License

/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 */
the documentation and/or other materials provided with the
distribution.

3. All advertising materials mentioning features or use of this
software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit. (http://www.openssl.org/)

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
endorse or promote products derived from this software without
prior written permission. For written permission, please contact
openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL"
nor may "OpenSSL" appear in their names without prior written
permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following
acknowledgment:
"This product includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit (http://www.openssl.org/)

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com). This product includes software written by Tim
Hudson (tjh@cryptsoft.com).

Original SSLeay License

/*

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.
*/
* This package is an SSL implementation written
  * by Eric Young (eay@cryptsoft.com).
  * The implementation was written so as to conform with Netscapes SSL.

* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).

* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
   * notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
   * notice, this list of conditions and the following disclaimer in the
   * documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
   * must display the following acknowledgement:
   * "This product includes cryptographic software written by
     * Eric Young (eay@cryptsoft.com)"
   * The word 'cryptographic' can be left out if the rouines from the library
   * being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
   * the apps directory (application code) you must include an acknowledgement:
   * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.
 *
* Permission to use, copy, modify, and/or distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
 *
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
* SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
* OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
* CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

Some files from Intel carry the following license:

# Copyright (c) 2012, Intel Corporation
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#
# * Redistributions of source code must retain the above copyright
#   notice, this list of conditions and the following disclaimer.
#
# * Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in the
#   documentation and/or other materials provided with the
#   distribution.
#
# * Neither the name of the Intel Corporation nor the names of its
#   contributors may be used to endorse or promote products derived from
#   this software without specific prior written permission.
#
#
# THIS SOFTWARE IS PROVIDED BY INTEL CORPORATION ""AS IS"" AND ANY
# EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
# PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL CORPORATION OR
# CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
# PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
# PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PCRE LICENCE
--------------

PCRE is a library of functions to support regular expressions whose syntax
and semantics are as close as possible to those of the Perl 5 language.

Release 8 of PCRE is distributed under the terms of the "BSD" licence, as
specified below. The documentation for PCRE, supplied in the "doc"
directory, is distributed under the same terms as the software itself. The data
in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also
included in the distribution is a set of C++ wrapper functions, and a
just-in-time compiler that can be used to optimize pattern matching. These
are both optional features that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS
-----------------------------

Written by: Philip Hazel
Email local part: ph10
Email domain: cam.ac.uk

University of Cambridge Computing Service,

Copyright (c) 1997-2015 University of Cambridge
All rights reserved.

PCRE JUST-IN-TIME COMPILATION SUPPORT
----------------------------------------

Open Source Used In DNAC Platform Antelope 11999
STACK-LESS JUST-IN-TIME COMPILER

Contributed by: Google Inc.

Copyright (c) 2007-2012, Google Inc.
All rights reserved.

THE "BSD" LICENCE

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

End

***************************************************************
pffft
***************************************************************

Copyright (c) 2013 Julien Pommier (pommier@modartt.com)

Based on original fortran 77 code from FFTPACKv4 from NETLIB,
authored by Dr Paul Swarztrauber of NCAR, in 1985.

As confirmed by the NCAR fftpack software curators, the following
FFTPACKv5 license applies to FFTPACKv4 sources. My changes are
released under the same terms.

FFTPACK license:

http://www.cisl.ucar.edu/css/software/fftpack5/ftp.html

Copyright (c) 2004 the University Corporation for Atmospheric
Research ("UCAR"). All rights reserved. Developed by NCAR's
Computational and Information Systems Laboratory, UCAR,
www.cisl.ucar.edu.

Redistribution and use of the Software in source and binary forms,
with or without modification, is permitted provided that the
following conditions are met:

- Neither the names of NCAR’s Computational and Information Systems
  Laboratory, the University Corporation for Atmospheric Research,
  nor the names of its sponsors or contributors may be used to
  endorse or promote products derived from this Software without
  specific prior written permission.

- Redistributions of source code must retain the above copyright
  notices, this list of conditions, and the disclaimer below.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer below in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

****************************

png

******************************

libpng

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

libpng versions 1.2.6, August 15, 2004, through 1.2.27, April 29, 2008, are Copyright (c) 2004, 2006-2008 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.2.5 with the following individual added to the list of Contributing Authors

Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5 - October 3, 2002, are Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors

Simon-Pierre Cadieux
Eric S. Raymond
Gilles Vollant

and with the following additions to the disclaimer:
There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997 Andreas Dilger Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.
Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

A "png_get_copyright" function is available, for convenient use in "about" boxes and the like:

    printf("%s",png_get_copyright(NULL));

Also, the PNG logo (in PNG format, of course) is supplied in the files "pngbar.png" and "pngbar.jpg" (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson
glennrp at users.sourceforge.net
April 29, 2008

********************************************************************************
protobuf
********************************************************************************

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

******************************
// Copyright (c) 2009 The RE2 Authors. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
// *
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
// *
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, 
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT 
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, 
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY 
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT 
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE 
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*****************************************

stlib
*****************************************

License for STLIB - A collection of public-domain single-file C/C++ libraries, primarily aimed at game developers.

The compilation and test files are licensed under the MIT license, but the single-file libraries themselves are in the public domain (free for use and modification for any purpose without legal friction).

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*****************************************

stl
*****************************************

SGI STL

The STL portion of GNU libstdc++ that is used with gcc3 and gcc4 is licensed under the GPL, with the following exception:
# As a special exception, you may use this file as part of a free software
# library without restriction. Specifically, if other files instantiate
# templates or use macros or inline functions from this file, or you compile
# this file and link it with other files to produce an executable, this
# file does not by itself cause the resulting executable to be covered by
# the GNU General Public License. This exception does not however
# invalidate any other reasons why the executable file might be covered by
# the GNU General Public License.

**************************
tinyxml
**************************

TinyXml is released under the zlib license:

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledge in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

**************************
tz
**************************

With a few exceptions, all files in the tz code and data (including
this one) are in the public domain. The exceptions are tzcode's
date.c, newstrtime.3, and strftime.c, which contain material derived
from BSD and which use the BSD 3-clause license.

**************************
utf
**************************
UTF-8 Library

The authors of this software are Rob Pike and Ken Thompson.
Copyright (c) 1998-2002 by Lucent Technologies.
Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.
THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR LUCENT TECHNOLOGIES MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

xmpmeta

xmpmeta. A fast XMP metadata parsing and writing library.
Copyright 2016 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Xorg

******************************************************
The following is the 'standard copyright' agreed upon by most contributors, and is currently the canonical license preferred by the X.Org Foundation. This is a slight variant of the common MIT license form published by the Open Source Initiative at http://www.opensource.org/licenses/mit-license.php

Copyright holders of new code should use this license statement where possible, and insert their name to this list. Please sort by surname for people, and by the full name for other entities (e.g. Juliusz Chroboczek sorts before Intel Corporation sorts before Daniel Stone).

See each individual source file or directory for the license that applies to that file.

Copyright (C) 2003-2006,2008 Jamey Sharp, Josh Triplett
Copyright 2009 Red Hat, Inc.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

******************************************************

The following licenses are 'legacy' - usually MIT/X11 licenses with the name of the copyright holder(s) in the license statement:

Copyright 1984-1994, 1998 The Open Group

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation.

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRES
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be
used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization from The Open Group.

X Window System is a trademark of The Open Group.

----------------------------------------

Copyright 2000 The XFree86 Project, Inc.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRES
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall
not be used in advertising or otherwise to promote the sale, use or
other dealings in this Software without prior written authorization
from the X Consortium.
of Xcms based on the TekColor Color Management System. TekColor is a trademark of Tektronix, Inc. The term “TekHVC” designates a particular color space that is the subject of U.S. Patent No. 4,985,853 (equivalent foreign patents pending). Permission is hereby granted to use, copy, modify, sell, and otherwise distribute this software and its documentation for any purpose and without fee, provided that:

1. This copyright, permission, and disclaimer notice is reproduced in all copies of this software and any modification thereof and in supporting documentation;
2. Any color-handling application which displays TekHVC color coordinates identifies these as TekHVC color coordinates in any interface that displays these coordinates and in any associated documentation;
3. The term "TekHVC" is always used, and is only used, in association with the mathematical derivations of the TekHVC Color Space, including those provided in this file and any equivalent pathways and mathematical derivations, regardless of digital (e.g., floating point or integer) representation.

Tektronix makes no representation about the suitability of this software for any purpose. It is provided "as is" and with all faults.

TEKTRONIX DISCLAIMS ALL WARRANTIES APPLICABLE TO THIS SOFTWARE, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL TEKTRONIX BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA, OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE, OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR THE PERFORMANCE OF THIS SOFTWARE.

(c) Copyright 1995 FUJITSU LIMITED
This is source code modified by FUJITSU LIMITED under the Joint Development Agreement for the CDE/Motif PST.

Copyright 1992 by Oki Technosystems Laboratory, Inc.
Copyright 1992 by Fuji Xerox Co., Ltd.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Oki Technosystems Laboratory and Fuji Xerox not be used in advertising or publicity.
pertaining to distribution of the software without specific, written
prior permission.
Oki Technosystems Laboratory and Fuji Xerox make no representations
about the suitability of this software for any purpose. It is provided
"as is" without express or implied warranty.

OKI TECHNO SYSTEMS LABORATORY AND FUJI XEROX DISCLAIM ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OKI TECHNO SYSTEMS
LABORATORY AND FUJI XEROX BE LIABLE FOR ANY SPECIAL, INDIRECT OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE
OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------


Permission to use, copy, modify, distribute, and sell this software
and its documentation for any purpose is hereby granted without fee,
provided that the above copyright notice appear in all copies and
that both that copyright notice and this permission notice appear
in supporting documentation, and that the name of FUJITSU LIMITED
not be used in advertising or publicity pertaining to distribution
of the software without specific, written prior permission.
FUJITSU LIMITED makes no representations about the suitability of
this software for any purpose.
It is provided "as is" without express or implied warranty.

FUJITSU LIMITED DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
EVENT SHALL FUJITSU LIMITED BE LIABLE FOR ANY SPECIAL, INDIRECT OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright (c) 1995 David E. Wexelblat. All rights reserved

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL DAVID E. WEXELBLAT BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of David E. Wexelblat shall
not be used in advertising or otherwise to promote the sale, use or
other dealings in this Software without prior written authorization
from David E. Wexelblat.

----------------------------------------
Copyright 1990, 1991 by OMRON Corporation

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the name OMRON not be used in
advertising or publicity pertaining to distribution of the software without
specific, written prior permission. OMRON makes no representations
about the suitability of this software for any purpose. It is provided
"as is" without express or implied warranty.

OMRON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
EVENT SHALL OMRON BE LIABLE FOR ANY SPECIAL, INDIRECT OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTUOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Digital Equipment Corporation

Portions Copyright 1990, 1991 by Tektronix, Inc
Rewritten for X.org by Chris Lee <clee@freedesktop.org>

Permission to use, copy, modify, distribute, and sell this documentation for any purpose and without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. Chris Lee makes no representations about the suitability for any purpose of the information in this document. It is provided "as-is" without express or implied warranty.

----------------------------------------

Copyright 1993 by Digital Equipment Corporation, Maynard, Massachusetts, Copyright 1994 by FUJITSU LIMITED Copyright 1994 by Sony Corporation

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Digital, FUJITSU LIMITED and Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL, FUJITSU LIMITED AND SONY CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL, FUJITSU LIMITED AND SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1991 by the Open Software Foundation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Open Software Foundation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Open Software Foundation makes no representations about the suitability of this
software for any purpose. It is provided "as is" without express or implied warranty.

OPEN SOFTWARE FOUNDATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OPEN SOFTWARE FOUNDATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1993, 1994 by Sony Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FUJITSU LIMITED and Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FUJITSU LIMITED and Sony Corporation makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

FUJITSU LIMITED AND SONY CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL FUJITSU LIMITED OR SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright (c) 1993, 1995 by Silicon Graphics Computer Systems, Inc.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Silicon Graphics not be used in advertising or publicity pertaining to distribution of the software without specific prior written permission. Silicon Graphics makes no representation about the suitability
of this software for any purpose. It is provided "as is"
without any express or implied warranty.

SILICON GRAPHICS DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS
SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SILICON
GRAPHICS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH
THE USE OR PERFORMANCE OF THIS SOFTWARE.

_____________________________________________________

Copyright 1993 by Digital Equipment Corporation

Permission to use, copy, modify, distribute, and sell this software
and its documentation for any purpose is hereby granted without fee,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name of FUJITSU LIMITED and
Digital Equipment Corporation not be used in advertising or publicity
pertaining to distribution of the software without specific, written
prior permission. FUJITSU LIMITED and Digital Equipment Corporation
makes no representations about the suitability of this software for
any purpose. It is provided "as is" without express or implied
warranty.

FUJITSU LIMITED AND DIGITAL EQUIPMENT CORPORATION DISCLAIM ALL
WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
FUJITSU LIMITED AND DIGITAL EQUIPMENT CORPORATION BE LIABLE FOR
ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF
THIS SOFTWARE.

_____________________________________________________

Copyright 1992, 1993 by FUJITSU LIMITED
Copyright 1993 by Fujitsu Open Systems Solutions, Inc.
Copyright 1994 by Sony Corporation

Permission to use, copy, modify, distribute and sell this software
and its documentation for any purpose is hereby granted without fee,
provided that the above copyright notice appear in all copies and
that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FUJITSU LIMITED, Fujitsu Open Systems Solutions, Inc. and Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FUJITSU LIMITED, Fujitsu Open Systems Solutions, Inc. and Sony Corporation make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

FUJITSU LIMITED, FUJITSU OPEN SYSTEMS SOLUTIONS, INC. AND SONY CORPORATION DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL FUJITSU OPEN SYSTEMS SOLUTIONS, INC., FUJITSU LIMITED AND SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1987, 1988, 1990, 1993 by Digital Equipment Corporation, Maynard, Massachusetts,

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Digital not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1993 by SunSoft, Inc.
Copyright 1999-2000 by Bruno Haible
Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of SunSoft, Inc. and Bruno Haible not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. SunSoft, Inc. and Bruno Haible make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

SunSoft Inc. AND Bruno Haible DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SunSoft, Inc. OR Bruno Haible BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 1991 by the Open Software Foundation
Copyright 1993 by the TOSHIBA Corp.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Open Software Foundation and TOSHIBA not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Open Software Foundation and TOSHIBA make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

OPEN SOFTWARE FOUNDATION AND TOSHIBA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OPEN SOFTWARE FOUNDATION OR TOSHIBA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 1988 by Wyse Technology, Inc., San Jose, Ca.,

All Rights Reserved
Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name Wyse not be
used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

WYSE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

----------------------------------------

Copyright 1991 by the Open Software Foundation
Copyright 1993, 1994 by the Sony Corporation

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the names of Open Software Foundation and
Sony Corporation not be used in advertising or publicity pertaining to
distribution of the software without specific, written prior permission.
Open Software Foundation and Sony Corporation make no
representations about the suitability of this software for any purpose.
It is provided "as is" without express or implied warranty.

OPEN SOFTWARE FOUNDATION AND SONY CORPORATION DISCLAIM ALL
WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OPEN
SOFTWARE FOUNDATIONN OR SONY CORPORATION BE LIABLE FOR ANY SPECIAL,
INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1992, 1993 by FUJITSU LIMITED
Copyright 1993 by Fujitsu Open Systems Solutions, Inc.
Open Source Used In DNAC Platform Antelope  12021

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FUJITSU LIMITED and Fujitsu Open Systems Solutions, Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
FUJITSU LIMITED and Fujitsu Open Systems Solutions, Inc. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

FUJITSU LIMITED AND FUJITSU OPEN SYSTEMS SOLUTIONS, INC. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL FUJITSU OPEN SYSTEMS SOLUTIONS, INC. AND FUJITSU LIMITED BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 1993, 1994 by Sony Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
Sony Corporation makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

SONY CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 1986, 1998  The Open Group
Copyright (c) 2000  The XFree86 Project, Inc.
Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation.

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
X CONSORTIUM OR THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Except as contained in this notice, the name of the X Consortium or of the
XFree86 Project shall not be used in advertising or otherwise to promote the
sale, use or other dealings in this Software without prior written
authorization from the X Consortium and the XFree86 Project.

----------------------------------------
Copyright 1990, 1991 by OMRON Corporation, NTT Software Corporation,
and Nippon Telegraph and Telephone Corporation
Copyright 1991 by the Open Software Foundation
Copyright 1993 by the FUJITSU LIMITED

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the names of OMRON, NTT Software, NTT, and
Open Software Foundation not be used in advertising or publicity
pertaining to distribution of the software without specific,
written prior permission. OMRON, NTT Software, NTT, and Open Software
Foundation make no representations about the suitability of this
software for any purpose. It is provided "as is" without express or
implied warranty.

OMRON, NTT SOFTWARE, NTT, AND OPEN SOFTWARE FOUNDATION
DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT
SHALL OMRON, NTT SOFTWARE, NTT, OR OPEN SOFTWARE FOUNDATION BE
LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1988 by Wyse Technology, Inc., San Jose, Ca,
Copyright 1987 by Digital Equipment Corporation, Maynard, Massachusetts,

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name Digital not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL AND WYSE DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL OR WYSE BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1991, 1992 by Fuji Xerox Co., Ltd.
Copyright 1992, 1993, 1994 by FUJITSU LIMITED

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Fuji Xerox, FUJITSU LIMITED not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Fuji Xerox, FUJITSU LIMITED make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

FUJI XEROX, FUJITSU LIMITED DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL FUJI XEROX, FUJITSU LIMITED BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA
Copyright 2006 Josh Triplett

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(c) Copyright 1996 by Sebastien Marineau and Holger Veit
<marineau@genie.uottawa.ca>
<Holger.Veit@gmd.de>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL HOLGER VEIT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF...
OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Sebastien Marineau or Holger Veit shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Holger Veit or Sebastien Marineau.

----------------------------------------
Copyright 1990, 1991 by OMRON Corporation, NTT Software Corporation, and Nippon Telegraph and Telephone Corporation
Copyright 1991 by the Open Software Foundation
Copyright 1993 by the TOSHIBA Corp.
Copyright 1993, 1994 by Sony Corporation
Copyright 1993, 1994 by the FUJITSU LIMITED

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of OMRON, NTT Software, NTT, Open Software Foundation, and Sony Corporation not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. OMRON, NTT Software, NTT, Open Software Foundation, and Sony Corporation make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

OMRON, NTT SOFTWARE, NTT, OPEN SOFTWARE FOUNDATION, AND SONY CORPORATION DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OMRON, NTT SOFTWARE, NTT, OPEN SOFTWARE FOUNDATION, OR SONY CORPORATION BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------
Copyright 2000 by Bruno Haible

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Bruno Haible not be used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission. Bruno Haible makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Bruno Haible DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL Bruno Haible BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 2003 Keith Packard

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Keith Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Keith Packard makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

KEITH PACKARD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL KEITH PACKARD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2007-2009, Troy D. Hanson
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS, IN NO EVENT SHALL IBM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------

Copyright 1990, 1991 by OMRON Corporation, NTT Software Corporation, and Nippon Telegraph and Telephone Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of OMRON, NTT Software, and NTT not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. OMRON, NTT Software, and NTT make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

OMRON, NTT SOFTWARE, AND NTT, DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL OMRON, NTT SOFTWARE, OR NTT, BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

************************
zlib
************************
(extracted from README, except for match.S)

Copyright notice:

(C) 1995-2013 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu

If you use the zlib library in a product, we would appreciate *not* receiving lengthy legal documents to sign. The sources are provided for free but without warranty of any kind. The library has been entirely written by Jean-loup Gailly and Mark Adler; it does not include third-party code.

If you redistribute modified sources, we would appreciate that you include in the file ChangeLog history information documenting your changes. Please read the FAQ for more information on the distribution of modified source versions.

(extracted from match.S, for match.S only)

Copyright (C) 1998, 2007 Brian Raiter <breadbox@muppetlabs.com>

This software is provided 'as-is', without any express or implied warranty. In no event will the author be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

************************************************
googleturl
************************************************

Copyright 2007, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The file url_parse.cc is based on nsURLParsers.cc from Mozilla. This file is licensed separately as follows:

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is mozilla.org code.

The Initial Developer of the Original Code is Netscape Communications Corporation.
Portions created by the Initial Developer are Copyright (C) 1998 the Initial Developer. All Rights Reserved.
Contributor(s):
Darin Fisher (original author)

Alternatively, the contents of this file may be used under the terms of either the GNU General Public License Version 2 or later (the "GPL"), or the GNU Lesser General Public License Version 2.1 or later (the "LGPL"), in which case the provisions of the GPL or the LGPL are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of either the GPL or the LGPL, and not to allow others to use your version of this file under the terms of the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL or the LGPL. If you do not delete the provisions above, a recipient may use your version of this file under the terms of any one of the MPL, the GPL or the LGPL.

The file icu_utf.cc is from IBM. This file is licensed separately as follows:

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2009 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

******************************

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of Your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

******************************************************************************
jsoncpp
******************************************************************************

The JsonCpp library's source code, including accompanying documentation, tests and demonstration applications, are licensed under the following conditions...

The author (Baptiste Lepilleur) explicitly disclaims copyright in all jurisdictions which recognize such a disclaimer. In such jurisdictions, this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of 2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur, and is released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this software may choose to accept it either as 1) Public Domain, 2) under the conditions of the MIT License (see below), or 3) under the terms of dual Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is described in clear, concise terms at:

http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

========================================================================
Copyright (c) 2007-2010 Baptiste Lepilleur
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

========================================================================
(END LICENSE TEXT)

The MIT license is compatible with both the GPL and commercial software, affording one all of the rights of Public Domain with the minor nuisance of being required to keep the above copyright notice and license text in the source code. Note also that by accepting the Public Domain "license" you can re-license your copy using whatever license you like.

**************************************************************************
libwebp
**************************************************************************
Copyright (c) 2010, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

QT AUTOMOTIVE SUITE LICENSE AGREEMENT
Agreement version 3.0

This Qt Automotive Suite License Agreement (Agreement) is a legal agreement between The Qt Company (as defined below) and the Licensee (as defined below) for the license of Licensed Software (as defined below). Capitalized terms used herein are defined in Section 1.

WHEREAS:

(A) Licensee wishes to use the Licensed Software for the purpose of developing and distributing Applications and/or Devices; and

(B) The Qt Company is willing to grant the Licensee a right to use Licensed Software for such purpose pursuant to term and conditions of this Agreement.

NOW, THEREFORE, THE PARTIES HEREBY AGREE AS FOLLOWS:

1. DEFINITIONS

"Affiliate" of a Party shall mean an entity (i) which is directly or indirectly controlling such Party; (ii) which is under the same direct or indirect ownership or control as such Party; or (iii) which is directly or indirectly owned or controlled by such Party. For these purposes, an entity shall be treated as being controlled by another if that other entity has fifty percent (50 %) or more of the votes in such entity, is able to direct its affairs and/or to control the composition of its board of directors or equivalent body.

"Applications" shall mean Licensee's software products created using the Licensed Software in connection with the Program, which may include the Redistributables, or part thereof.

"Contractor(s)" shall mean third party consultants, distributors and contractors...
performing services to a Party under applicable contractual arrangement.

"Customer(s)" shall mean Licensee's end users to whom Licensee, directly or indirectly, distributes copies of the Redistributables.

"Deployment Platforms" shall mean operating systems specified in the License Certificate, in which the Redistributables can be distributed pursuant to the terms and conditions of this Agreement.

"Designated User(s)" shall mean the employee(s) of Licensee or Licensee's Affiliates acting within the scope of their employment or Licensee's Contractors acting within the scope of their services for Licensee and on behalf of Licensee. Designated Users shall be named in the License Certificate.

"Development License" shall mean the license needed by the Licensee for each Designated User to use the Licensed Software under the license grant described in Section 3.1 of this Agreement.

"Development Platforms" shall mean those operating systems specified in the License Certificate, in which the Licensed Software can be used under the Development License, but not distributed in any form or used for any other purpose.

"Devices" shall mean hardware devices or products that 1) are manufactured and/or distributed by the Licensee or its Affiliates or Contractors in connection with the Program, and (2)(i) incorporate or integrate the Redistributables or parts thereof; or (ii) do not incorporate or integrate the Redistributables at the time of distribution, but where, when used by a Customer, the main user interface or substantial functionality of such device is provided by Application(s) or otherwise depends on the Licensed Software.

"Distribution License(s)" shall mean the license required for distribution of Redistributables in accordance with the license grant described in Section 3.2(ii)-(iii) of this Agreement.

"Distribution License Packs" shall mean set of prepaid Distribution Licenses for distribution of Redistributables, as defined in The Qt Company's standard price list, quote, Purchase Order confirmation or in an appendix hereto, as the case may be.

"Initial Support Term" shall mean a time period of twelve (12) months, calculated from the effective date of this Agreement.

"Intellectual Property Rights" shall mean patents (including utility models), design patents, and designs (whether or not capable of registration), chip topography rights and other like protection, copyrights, trademarks, service marks, trade names, logos or other words or symbols and any other form of
statutory protection of any kind and applications for any of the foregoing as well as any trade secrets.

"Licensee" shall mean the individual or legal entity that is party to this Agreement, as identified on the signature page hereof.

"License Certificate" shall mean a certificate accompanying the Licensed Software and generated for each Designated User respectively. License Certificate will specify the Designated User, the Development Platforms, Deployment Platforms, Program and the Term of this Agreement. The terms of the License Certificate are considered part of this Agreement and shall be updated from time to time to reflect any changes to the foregoing terms relating to Licensee's rights to the Licensed Software.

"Licensee's Records" shall mean books and records that are likely to contain information bearing on Licensee's compliance with this Agreement or the payments due to The Qt Company under this Agreement, including, but not limited to: assembly logs, sales records and distribution records.

"Licensees SDK Contractors" shall mean Contractors of Licensee, who have purchased or received SDK from the Licensee relating to the Program.

"License Fee" shall mean the fee charged to the Licensee for rights granted under the terms of this Agreement.

"Licensed Software" shall mean all versions of The Qt Company's computer software products, online or electronic documentation, associated media and printed materials, including the source code, example programs and the documentation, licensed to the Licensee under this Agreement. Licensed Software does not include Third Party Software (as defined in Section 4) or Open Source Qt.

"Modified Software" shall mean bug-fixes, error corrections, patches or modifications made to the Licensed Software by Licensee, including documentation related thereto.

"Online Services" shall mean any services or access to systems made available by The Qt Company to the Licensee over the Internet relating to the Licensed Software or for the purpose of use by the Licensee of the Licensed Software or Support. Use of any such Online Services is discretionary for the Licensee and some of them may be subject to additional fees.

"Open Source Qt" shall mean all versions of The Qt Company's Qt computer software products, online or electronic documentation, associated media and printed materials, including the source code, example programs and the documentation available under the terms of the GNU Lesser General Public License, version 2.1 or later ("LGPL") or the GNU General Public License, version 2.0 or later ("GPL").
"Party" or "Parties" shall mean Licensee and/or The Qt Company.

"Program" shall mean Licensees business program for which purpose the Licensee is entitled to use the Licensed Software and grant the Licensee’s SDK Contractors a right to use the Licensed Software as part of a SDK.

"Redistributables" shall mean the portions of the Licensed Software set forth in Appendix 1, Section 1 that may be distributed pursuant to the terms of this Agreement in object code form only, including any relevant documentation. Where relevant, any reference to Licensed Software in this Agreement shall include and refer also to Redistributables.

"SDK" or "Software Development Kit" shall mean a combination of software modules including Licensed Software intended to be utilized in connection with the Program.

"Submitted Modified Software" shall have the meaning as set forth in Section 2.3.

"Support" shall mean standard developer support that is provided by The Qt Company to assist Designated Users in using the Licensed Software in accordance with The Qt Company's standard support terms.

"Support Renewal Term" shall mean a time period of twelve (12) months, calculated from the end of the Initial Support Term or previous Support Renewal Term, as applicable.

"Support Term" shall mean the Initial Support Term and any possible Support Renewal Term(s) during which time the Licensee is eligible to receive for Support for the Licensed Software.

"Taxes" shall have the meaning set forth in Section 10.5.

"Term" shall mean the validity period of this Agreement, as set forth in the License Certificate.

The Qt Company shall mean:

(i) in the event Licensee is an individual residing in the United States or a legal entity incorporated in the United States or having its headquarters in the United States, The Qt Company Inc., a Delaware corporation with its office at 2350 Mission College Blvd., Suite 1020, Santa Clara, CA 95054, USA.; or

(ii) in the event the Licensee is an individual residing outside of the United States or a legal entity incorporated outside of the United States or having its registered office outside of the United States, The Qt Company Ltd., a Finnish company with its registered office at Bertel Jungin aukio D3A, 02600 Espoo, Finland.
“Updates” shall mean a release or version of the Licensed Software containing bug fixes, error corrections and other changes that are generally made available to users of the Licensed Software that have contracted for Support. Updates are generally depicted as a change to the digits following the decimal in the Licensed Software version number. The Qt Company shall make Updates available to the Licensee under the Support. Updates shall be considered as part of the Licensed Software hereunder.

“Upgrades” shall mean a release or version of the Licensed Software containing enhancements and new features and are generally depicted as a change to the first digit of the Licensed Software version number. In the event Upgrades are provided to the Licensee under this Agreement, they shall be considered as part of the Licensed Software hereunder.

2. OWNERSHIP

2.1 Ownership of The Qt Company

The Licensed Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Licensed Software is licensed, not sold.

All The Qt Company's Intellectual Property Rights are and shall remain the exclusive property of The Qt Company or its licensors respectively.

2.2 Ownership of Licensee

All the Licensee's Intellectual Property Rights are and shall remain the exclusive property of the Licensee or its licensors respectively.

All Intellectual Property Rights to the Modified Software, Applications and Devices shall remain with the Licensee and no rights thereto shall be granted by the Licensee to The Qt Company under this Agreement (except as set forth in Section 2.3 below).

2.3 Modified Software

Licensee may create Modified Software that breaks the source or binary compatibility with the Licensed Software. This includes, but is not limited to, changing the application programming interfaces ("API") by adding, changing or deleting any variable, method, or class signature in the Licensed Software and/or any inter-process protocols, services or standards in the Licensed Software libraries. To the extent that Licensee breaks source or binary compatibility with the Licensed Software, Licensee acknowledges that The Qt Company's ability to provide Support may be prevented or limited and Licensee's ability to make use of Updates may be restricted.
To the extent Licensee submits Modified Software to The Qt Company ("Submitted Modified Software"), Licensee hereby grants The Qt Company a sublicensable, assignable, irrevocable, perpetual, worldwide, non-exclusive, royalty-free and fully paid-up license, under all of Licensee's Intellectual Property Rights, to reproduce, adapt, translate, modify, and prepare derivative works of, publicly display, publicly perform, sublicense, make available and distribute such Submitted Modified Software as The Qt Company sees fit at its free and absolute discretion. For the sake of clarity, the Licensee shall have no obligation to provide Modified Software to The Qt Company.

3. LICENSES GRANTED

3.1 Development with Licensed Software

Subject to the terms of this Agreement, The Qt Company grants to Licensee a personal, worldwide, non-exclusive, non-transferable license, valid for the Term, to use, modify and copy the Licensed Software by Designated Users on the Development Platforms for the sole purposes of designing, developing, demonstrating and testing Application(s) and/or Devices, and to provide thereto related support and other services to end-user Customers.

Licensee may install copies of the Licensed Software on an unlimited number of computers provided that (i) only the Designated Users may use the Licensed Software, and (ii) all Designated Users must have a valid Development License to use Licensed Software.

Licensee may at any time designate another Designated User to replace a then-current Designated User by notifying The Qt Company in writing, provided that any Designated User may be replaced only once during any six-month period.

3.2 Distribution of Redistributables

Subject to the terms of this Agreement, The Qt Company grants to Licensee a personal, worldwide, non-exclusive, non-transferable license, valid for the Term, to (i) distribute, by itself or through its Contractors, Redistributables as installed, incorporated or integrated into Applications for execution on the Deployment Platforms, and (ii) distribute, by itself or through one or more tiers of Contractors, Redistributables as installed, incorporated or integrated, or intended to be installed, incorporated or integrated into Devices for execution on the Deployment Platforms, and (iii) grant sublicenses to Redistributables, as distributed hereunder, for Customers solely for Customer's internal use and to the extent necessary in order for the Customers to use the Applications and/or Devices for their respective intended purposes.

Right to distribute the Redistributables as provided herein is conditional upon the Licensee having purchased and paid the appropriate amount of Development and Distribution Licenses from The Qt Company before distributing any Redistributables to Customers.
For the avoidance of any doubt it is specifically acknowledged and agreed that distribution of Redistributables solely as installed, incorporated or integrated into Applications for execution on the Deployment Platform(s), as specified in (i) of the first paragraph of Section 3.2 above, i.e. with no connection to Devices or intention to use in connection therewith, shall not require a Distribution License.

3.3 SDK License

The Qt Company grants to Licensee a personal, worldwide, non-exclusive, non-transferable license, valid for the Term, to (i) distribute Licensed Software as a part of the SDK to Licensees SDK Contractors in connection with the Program and (ii) in connection with the Program, by itself or by Licensee's SDK Contractors, combine, incorporate or integrate Licensed Software with, or use Licensed Software for creation of, any software created with or incorporating Open Source Qt, provided, however, that:

(i) the Licensees SDK Contractors are only entitled to use the Licensed Software as part of SDK and for the sole purpose of developing software for Devices that are distributed under the Program; and

(ii) Licensees SDK Contractors shall not be entitled to distribute the SDK or any part thereof to any third parties.

For the avoidance of any doubt, the distribution of such software development tools that do not contain Licensed Software shall not be covered by this Agreement.

3.4 Further Requirements

The licenses granted above in this Section 3 by The Qt Company to Licensee are conditional and subject to Licensee's compliance with the following terms:

(i) Licensee shall not remove or alter any copyright, trademark or other proprietary rights notice contained in any portion of the Licensed Software;

(ii) Applications and SDKs must add primary and substantial functionality to the Licensed Software;

(iii) Applications may not pass on functionality which in any way makes it possible for others to create software with the Licensed Software; provided however that Licensee may use the Licensed Software's scripting and QML ("Qt Quick") functionality solely in order to enable scripting, themes and styles that augment the functionality and appearance of the Application(s) without adding primary and substantial functionality to the Application(s);

(iv) Applications and SDKs must not compete with the Licensed Software;
(v) Licensee shall not use The Qt Company's or any of its suppliers' names, logos, or trademarks to market Applications or SDKs, except that Licensee may use "Built with Qt" logo to indicate that Application(s) was developed using the Licensed Software;

(vi) Except as expressly provided in Section 3.3, Licensee shall not distribute, sublicense or disclose source code of Licensed Software to any third party (provided however that Licensee may appoint employee(s) of Contractors as Designated Users to use Licensed Software pursuant to this Agreement);

(vii) Licensee shall not grant the Customers a right to (i) make copies of the Redistributables except when and to the extent required to use the Applications and/or Devices for their intended purpose, (ii) modify the Redistributables or create derivative works thereof, (iii) decompile, disassemble or otherwise reverse engineer Redistributables, or (iv) redistribute any copy or portion of the Redistributables to any third party, except as part of the onward sale of the Device on which the Redistributables are installed;

(viii) Except as expressly provided in Section 3.3, Licensee shall not and shall cause that its Affiliates, Contractors and Licensee's SDK Contractors shall not a) in any way, combine, incorporate or integrate Licensed Software with, or use Licensed Software for creation of, any software created with or incorporating Open Source Qt or b) incorporate or integrate Applications into a hardware device or product other than a Device, unless Licensee has received an advance written permission from The Qt Company to do so. Unless specifically otherwise agreed, any and all distribution by the Licensee during the Term of a hardware device or product a) which incorporate or integrate any part of Licensed Software or Open Source Qt; or b) where the main user interface or substantial functionality is provided by software build with Licensed Software or Open Source Qt or otherwise depends on the Licensed Software or Open Source Qt, shall be considered as distribution under this Agreement and dependent on compliance thereof (including but not limited to obligation to pay applicable License Fees for such distribution);

(ix) Licensee shall cause all of its Affiliates and Contractors entitled to make use of the licenses granted under this Agreement, to be contractually bound to comply with the relevant terms of this Agreement and not to use the Licensed Software beyond the terms hereof and for any purposes other than operating within the scope of their services for Licensee. Licensee shall be responsible for any and all actions and omissions of its Affiliates and Contractors relating to the Licensed Software and use thereof (including but not limited to payment of all applicable License Fees);

(x) Except when and to the extent explicitly provided in this Section 3, Licensee shall not transfer, publish, disclose, display or otherwise make available the Licensed Software;
(xi) Licensee shall not take any action inconsistent with The Qt Company's Intellectual Property Rights; and

(xii) Attempt or enlist a third party to conduct or attempt to conduct any of the above.

Above terms shall not be applicable if and to the extent they conflict with any mandatory provisions of any applicable laws.

Any use of Licensed Software beyond the provisions of this Agreement is strictly prohibited and requires an additional license from The Qt Company.

4. THIRD PARTY SOFTWARE

The Licensed Software may provide links to third party libraries or code (collectively "Third Party Software") to implement various functions. Third Party Software does not comprise part of the Licensed Software. In some cases, access to Third Party Software may be included in the Licensed Software. Such Third Party Software will be listed in the ".../src/3rdparty" source tree delivered with the Licensed Software or documented in the Licensed Software, as such may be amended from time to time. Licensee acknowledges that use or distribution of Third Party Software is in all respects subject to applicable license terms of applicable third party right holders.

5. PRE-RELEASE CODE

The Licensed Software may contain pre-release code and functionality marked or otherwise stated as "Technology Preview", "Alpha", "Beta" or similar designation. Such pre-release code may be present in order to provide experimental support for new platforms or preliminary versions of one or more new functionalities. The pre-release code may not be at the level of performance and compatibility of a final, generally available, product offering of the Licensed Software. The pre-release parts of the Licensed Software may not operate correctly, may contain errors and may be substantially modified by The Qt Company prior to the first commercial product release, if any. The Qt Company is under no obligation to make pre-release code commercially available, or provide any Support or Updates relating thereto. The Qt Company assumes no liability whatsoever regarding any pre-release code, but any use thereof is exclusively at Licensee's own risk and expense.

6. LIMITED WARRANTY AND WARRANTY DISCLAIMER

The Qt Company hereby represents and warrants that it has the power and authority to grant the rights and licenses granted to Licensee under this Agreement.

Except as set forth above, the Licensed Software is licensed to Licensee "as is".

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE QT COMPANY ON BEHALF OF
ITSELF AND ITS LICENSORS, SUPPLIERS AND AFFILIATES, DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT WITH REGARD TO THE LICENSED SOFTWARE. THE QT COMPANY DOES NOT WARRANT THAT THE LICENSED SOFTWARE WILL SATISFY LICENSEE'S REQUIREMENTS OR THAT IT WILL OPERATE WITHOUT DEFECT OR ERROR OR THAT THE OPERATION THEREOF WILL BE UNINTERRUPTED. ALL USE OF AND RELIANCE ON THE LICENSED SOFTWARE IS AT THE SOLE RISK OF AND RESPONSIBILITY OF LICENSEE.

Licensee’s exclusive remedy and The Qt Company's entire liability for Licensed Software shall be limited, at The Qt Company’s option, to correction of the error, replacement of the Licensed Software or return of the applicable fees paid for the defective Licensed Software for the time period during which the License is not able to utilize the Licensed Software under the terms of this Agreement.

7. INDEMNIFICATION AND LIMITATION OF LIABILITY

7.1 Limitation of Liability

EXCEPT FOR (I) CASES OF GROSS NEGLIGENCE OR INTENTIONAL MISCONDUCT, (II) LICENSEE’S DUTY TO PAY ALL APPLICABLE LICENSE FEES AND COMPENSATIONS, AND (III) BREACH OF CONFIDENTIALITY, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL EITHER PARTY BE LIABLE TO THE OTHER PARTY FOR ANY LOSS OF PROFIT, LOSS OF DATA, LOSS OF BUSINESS OR GOODWILL OR ANY OTHER INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE COST, DAMAGES OR EXPENSE OF ANY KIND, HOWSOEVER ARISING UNDER OR IN CONNECTION WITH THIS AGREEMENT.

EXCEPT FOR (I) CASES OF GROSS NEGLIGENCE OR INTENTIONAL MISCONDUCT, (II) LICENSEE’S DUTY TO PAY ALL APPLICABLE LICENSE FEES AND COMPENSATIONS, AND (III) BREACH OF CONFIDENTIALITY, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL EITHER PARTY’S TOTAL AGGREGATE LIABILITY UNDER THIS AGREEMENT EXCEED THE AGGREGATE LICENSE FEES RECEIVED BY THE QT COMPANY FROM LICENSEE DURING THE PERIOD OF TWELVE (12) MONTHS IMMEDIATELY PRECEDING THE EVENT RESULTING IN SUCH LIABILITY.

THE PROVISIONS OF THIS SECTION 7 ALLOCATE THE RISKS UNDER THIS AGREEMENT BETWEEN THE QT COMPANY AND LICENSEE AND THE PARTIES HAVE RELIED UPON THE LIMITATIONS SET FORTH HEREIN IN DETERMINING WHETHER TO ENTER INTO THIS AGREEMENT.

7.2 Licensees Indemnification

Licensee shall indemnify and hold harmless The Qt Company from and against any claim, injury, judgment, settlement, loss or expense, including attorneys’ fees related to: (a) Licensee's misrepresentation in connection with The Qt Company or the Licensed Software or breach of this Agreement, (b) the Application or Device (except where such cause of liability is solely attributable to the Licensed Software).
8. SUPPORT, UPDATES AND ONLINE SERVICES

Licensee will be eligible to receive Support and Updates and to use the Online Services during the Support Term. Unless otherwise decided by The Company at its free and absolute discretion, Upgrades will not be included in the Support but may be available subject to additional fees.

Licenses granted under this Agreement shall include a prepaid Initial Support Term.

Initial Support Term shall be automatically extended to one or more Support Renewal Term(s), unless and until either Party notifies the other Party in writing that it does not wish to continue the Support, such notification to be provided to the other Party no less than ninety (90) days before expiry of the Initial Support Term or respective Support Renewal Term. During any such Support Renewal Term Support shall be available subject to prices and terms agreed between the Parties or, if no advance agreement exists, subject to The Qt Company's standard pricing applicable at the commencement date of any such Support Renewal Term. From time to time The Qt Company may change Support provided within each Support plan; provided that during the respective Initial Support Term or Support Renewal Term (as the case may be), the level of Support provided by The Qt Company may not be reduced without the consent of the Licensee.

Unless otherwise agreed, The Qt Company shall not be responsible for providing any service or support to the Customers.

9. CONFIDENTIALITY

Each Party acknowledges that during the Term of this Agreement each Party may receive information about the other Party's business, business methods, business plans, customers, business relations, technology, and other information, including the terms of this Agreement, that is confidential and of great value to the other Party, and the value of which would be significantly reduced if disclosed to third parties ("Confidential Information"). Accordingly, when a Party (the "Receiving Party") receives Confidential Information from the other Party (the "Disclosing Party"), the Receiving Party shall only disclose such information to employees and Contractors on a need to know basis, and shall cause its employees and employees of its Affiliates to: (i) maintain any and all Confidential Information in confidence; (ii) not disclose the Confidential Information to a third party without the Disclosing Party's prior written approval; and (iii) not, directly or indirectly, use the Confidential Information for any purpose other than for exercising its rights and fulfilling its responsibilities pursuant to this Agreement. Each Party shall take reasonable measures to protect the Confidential Information of the other Party, which measures shall not be less than the measures taken by such Party to protect its own confidential and proprietary information.
Obligation of confidentiality shall not apply to information that (i) is or becomes generally known to the public through no act or omission of the Receiving Party; (ii) was in the Receiving Party’s lawful possession prior to the disclosure hereunder and was not subject to limitations on disclosure or use; (iii) is developed independently by employees or Contractors of the Receiving Party or other persons working for the Receiving Party who have not had access to the Confidential Information of the Disclosing Party, as proven by the written records of the Receiving Party; (iv) is lawfully disclosed to the Receiving Party without restrictions, by a third party not under an obligation of confidentiality; or (v) the Receiving Party is legally compelled to disclose, in which case the Receiving Party shall notify the Disclosing Party of such compelled disclosure and assert the privileged and confidential nature of the information and cooperate fully with the Disclosing Party to limit the scope of disclosure and the dissemination of disclosed Confidential Information to the minimum extent necessary.

The obligations under this Section 9 shall continue to remain in force for a period of five (5) years after the last disclosure, and, with respect to trade secrets, for so long as such trade secrets are protected under applicable trade secret laws.

10. FEES, DELIVERY AND PAYMENT

10.1 License Fees

License Fees are described in The Qt Company’s standard price list, quote or Purchase Order confirmation or in an appendix hereto, as the case may be. The License Fees shall not be refunded or claimed as a credit, even on the ground that Distribution Licenses are not used, i.e. Redistributables are not actually distributed corresponding to the Distribution Licenses purchased, or for any other reason.

10.2 Ordering Licenses

Licensee may purchase Development Licenses and Distribution Licenses pursuant to agreed pricing terms or, if no specific pricing terms have been agreed upon, at The Qt Company’s standard pricing terms applicable at the time of purchase.

Licensee shall submit all purchase orders for Development Licenses and Distribution Licenses to The Qt Company by email or any other method acceptable to The Qt Company (each such order is referred to herein as a “Purchase Order”) for confirmation, whereupon the Purchase Order shall become binding between the Parties.

10.3 Distribution

License Packs Unless otherwise agreed, the Distribution Licenses are bought by
way of Distribution License Packs.

Upon due payment of the ordered Distribution License Pack(s), the Licensee will have an account of Distribution Licenses available for installing, bundling or integrating (all jointly "installing") the Redistributables with the Devices or for otherwise distributing the Redistributables in accordance with this Agreement.

Each time Licensee "installs” or distributes a copy of Redistributables, then one Distribution License is used, and Licensee's account of available Distribution Licenses is decreased accordingly.

Licensee may "install” copies of the Redistributables so long as Licensee has Distribution Licenses remaining on its account.

Redistributables will be deemed to have been "installed” into a Device when one of the following circumstances shall have occurred: a) the Redistributables have been loaded onto the Device and used outside of the Licensee's premises or b) the Device has been fully tested and placed into Licensee's inventory (or sold) for the first time (i.e., Licensee will not be required to use (or pay for) more than one Distribution License for each individual Device, e.g. in a situation where a Device is returned to Licensee's inventory after delivery to a distributor or sale to a Customer). In addition, if Licensee includes a back-up copy of the Redistributables on a CD-ROM or other storage medium along with the product, that backup copy of the Redistributables will not be deemed to have been "installed” and will not require an additional Distribution License.

10.4 Payment Terms
License Fees and any other charges under this Agreement shall be paid by Licensee no later than thirty (30) days from the date of the applicable invoice from The Qt Company.

The Qt Company will submit an invoice to Licensee after the date of this Agreement and/or after The Qt Company receives a Purchase Order from Licensee. A late payment charge of the lower of (a) one percent per month; or (b) the interest rate stipulated by applicable law, shall be charged on any unpaid balances that remain past due.

The Qt Company shall have the right to suspend, terminate or withhold grants of all rights to the Licensed Software hereunder, including but not limited to the Developer License, Distribution License, and Support, should Licensee fail to make payment in a timely fashion.

10.5 Taxes
All License Fees and other charges payable hereunder are gross amounts but exclusive of any value added tax, use tax, sales tax and other taxes, duties or tariffs ("Taxes"). Such applicable Taxes shall be paid by Licensee, or, where
applicable, in lieu of payment of such Taxes, Licensee shall provide an exemption certificate to The Qt Company and any applicable authority.

11 RECORD-KEEPING AND REPORTING OBLIGATIONS; AUDIT RIGHTS

11.1 Licensee's Record-keeping

Licensee shall at all times maintain accurate and up-to-date written records of Licensee's activities related to the use of Licensed Software and distribution of Redistributables. The records shall be adequate to determine Licensee's compliance with the provisions of this Agreement and to demonstrate the number of Designated Users and Redistributables distributed by Licensee. The records shall conform to good accounting practices reasonably acceptable to The Qt Company.

Licensee shall, within thirty (30) days from the end of each calendar quarter, deliver to The Qt Company a report detailing the number of Designated Users and copies of Redistributables distributed by Licensee during that calendar quarter, and also detailing the number of undistributed copies of Redistributables made by Licensee and remaining in its account (i.e., undistributed copies for which Distribution Licenses have been or need to be obtained from The Qt Company). Such report shall contain such other information as The Qt Company shall reasonably require from time to time.

11.2. The Qt Company's Audit Rights

The Qt Company or an independent auditor acting on behalf of The Qt Company's, may, upon at least five (5) business days' prior written notice and at its expense, audit Licensee with respect to the use of the Redistributables, but not more frequently than once during each 6-month period. Such audit may be conducted by mail, electronic means or through an in-person visit to Licensee's place of business. Any such in-person audit shall be conducted during regular business hours at Licensee's facilities and shall not unreasonably interfere with Licensee's business activities. The Qt Company or the independent auditor acting on behalf of The Qt Company shall be entitled to inspect Licensee's Records. All such Licensee's Records and use thereof shall be subject to an obligation of confidentiality under this Agreement.

If an audit reveals that Licensee is using the Licensed Software beyond scope of the licenses Licensee has paid for, Licensee agrees to immediately pay The Qt Company any amounts owed for such unauthorized use.

In addition, in the event the audit reveals a material violation of the terms of this Agreement (underpayment of more than 5% of License Fees shall always be deemed a material violation for purposes of this section), then the Licensee shall pay The Qt Company's reasonable cost of conducting such audit.

12 TERM AND TERMINATION
12.1 Term

This Agreement shall enter into force upon due acceptance by both Parties and remain in force for the Term, unless and until terminated pursuant to the terms of this Section 12.

12.2 Termination by The Qt Company

The Qt Company shall have the right to terminate this Agreement upon thirty (30) days prior written notice if (i) the Licensee is in material breach of any obligation of this Agreement and fails to remedy such breach within such notice period; (ii) or Licensee or any of its Affiliates bring a suit before any court or administrative agency or otherwise assert a claim against The Qt Company's or any of its Affiliates' Intellectual Property Rights or validity thereof.

12.3 Mutual Right to Terminate

Either Party shall have the right to terminate this Agreement immediately upon written notice in the event that the other Party becomes insolvent, files for any form of bankruptcy, makes any assignment for the benefit of creditors, has a receiver, administrative receiver or officer appointed over the whole or a substantial part of its assets, ceases to conduct business, or an act equivalent to any of the above occurs under the laws of the jurisdiction of the other Party.

12.4 Parties Rights and Duties upon Termination

Upon expiry or termination of the Agreement for any reason, Licensee shall, within 30 days after such termination, cease and shall cause all Designated Users (including those of its Affiliates' and Contractors') and Licensee's SDK Contractors to cease using the Licensed Software and distribution of the Redistributables under this Agreement. Notwithstanding the above, in the event the Agreement expires or is terminated for reason other than by The Qt Company pursuant to Section 12.2, the Licensee is entitled, for a period of six (6) months after the effective date of termination, to continue distribution of Devices under the Distribution Licenses paid but unused at such effective date of termination.

Upon any such termination the Licensee shall destroy or return to The Qt Company all copies of the Licensed Software and all related materials and will certify the same to The Qt Company upon its request, provided however that Licensee may retain and exploit such copies of the Licensed Software as it may reasonably require in providing continued support to Customers.

Expiry or termination of this Agreement for any reason whatsoever shall not relieve Licensee of its obligation to pay any License Fees accrued or payable to The Qt Company prior to the effective date of termination, and Licensee shall
immediately pay to The Qt Company all such fees upon the effective date of termination. Termination of this Agreement shall not affect any rights of Customers to continue use of Applications and Devices (and therein incorporated Redistributables).

13. GOVERNING LAW AND LEGAL VENUE

In the event this Agreement is in the name of The Qt Company Inc., a Delaware Corporation, then:

(i) this Agreement shall be construed and interpreted in accordance with the laws of the State of California, USA, excluding its choice of law provisions;

(ii) the United Nations Convention on Contracts for the International Sale of Goods will not apply to this Agreement; and

(iii) any dispute, claim or controversy arising out of or relating to this Agreement or the breach, termination, enforcement, interpretation or validity thereof, including the determination of the scope or applicability of this Agreement to arbitrate, shall be determined by arbitration in San Francisco, USA, before one arbitrator. The arbitration shall be administered by JAMS pursuant to JAMS' Streamlined Arbitration Rules and Procedures. Judgment on the Award may be entered in any court having jurisdiction. This Section shall not preclude parties from seeking provisional remedies in aid of arbitration from a court of appropriate jurisdiction.

In the event this Agreement is in the name of The Qt Company Ltd., a Finnish Company, then:

(i) this Agreement shall be construed and interpreted in accordance with the laws of Finland, excluding its choice of law provisions;

(ii) the United Nations Convention on Contracts for the International Sale of Goods will not apply to this Agreement; and

(iii) any disputes, controversy or claim arising out of or relating to this Agreement, or the breach, termination or validity thereof shall be finally settled by arbitration in accordance with the Arbitration Rules of Finland Chamber of Commerce. The arbitration tribunal shall consist of one (1), or if either Party so requires, of three (3), arbitrators. The award shall be final and binding and enforceable in any court of competent jurisdiction. The arbitration shall be held in Helsinki, Finland and the process shall be conducted in the English language. This Section shall not preclude parties from seeking provisional remedies in aid of arbitration from a court of appropriate jurisdiction.

14. GENERAL PROVISIONS
14.1 No Assignment Licensee

shall not be entitled to assign or transfer all or any of its rights, benefits and obligations under this Agreement without the prior written consent of The Qt Company, which shall not be unreasonably withheld or delayed. The Qt Company shall be entitled to freely assign or transfer any of its rights, benefits or obligations under this Agreement.

14.2 No Third Party Representations

Licensee shall make no representations or warranties concerning the Licensed Software on behalf of The Qt Company. Any representation or warranty Licensee makes or purports to make on The Qt Company's behalf shall be void as to The Qt Company.

14.3 Surviving Sections

Any terms and conditions that by their nature or otherwise reasonably should survive termination of this Agreement shall so be deemed to survive.

14.4 Entire Agreement

This Agreement, the exhibits hereto, the License Certificate and any applicable Purchase Order constitute the complete agreement between the Parties and supersedes all prior or contemporaneous discussions, representations, and proposals, written or oral, with respect to the subject matters discussed herein. In the event of any conflict or inconsistency between this Agreement and any Purchase Order, the terms of this Agreement will prevail over the terms of the Purchase Order with respect to such conflict or inconsistency.

14.5 Modifications

No modification of this Agreement shall be effective unless contained in a writing executed by an authorized representative of each Party. No term or condition contained in Licensee's Purchase Order shall apply unless expressly accepted by The Qt Company in writing.

14.6 Force Majeure

Except for the payment obligations hereunder, neither Party shall be liable to the other for any delay or non-performance of its obligations hereunder in the event and to the extent that such delay or non-performance is due to an event of act of God, terrorist attack or other similar unforeseeable catastrophic event that prevents either Party for fulfilling its obligations under this Agreement and which such Party cannot avoid or circumvent ("Force Majeure Event"). If the Force Majeure Event results in a delay or non-performance of a Party for a period of three (3) months or longer, then either Party shall have the right to terminate this Agreement with immediate effect without any
liability (except for the obligations of payment arising prior to the event of Force Majeure) towards the other Party.

14.7 Notices

Any notice given by one Party to the other shall be deemed properly given and deemed received if specifically acknowledged by the receiving Party in writing or when successfully delivered to the recipient by hand, fax, or special courier during normal business hours on a business day to the addresses specified for The Qt Company in the beginning of this Agreement, and for the Licensee in the Licensees account profile. Each communication and document made or delivered by one Party to the other Party pursuant to this Agreement shall be in the English language.

14.8 Export Control

Licensee acknowledges that the Redistributables may be subject to export control restrictions under the applicable laws of respective countries. Licensee shall fully comply with all applicable export license restrictions and requirements as well as with all lawses hereunder and shall procure all necessary governmental authorizations, including without limitation, all necessary licenses, approvals, permissions or consents, where necessary for the re-exportation of the Redistributables, Applications and/or Devices.

14.9 No Implied License

There are no implied licenses or other implied rights granted under this Agreement, and all rights, save for those expressly granted hereunder, shall remain with The Qt Company and its licensors. In addition, no licenses or immunities are granted to the combination of the Licensed Software with any other software or hardware not delivered by The Qt Company under this Agreement.

14.10 Attorney Fees

The prevailing Party in any action to enforce this Agreement shall be entitled to recover its attorney's fees and costs in connection with such action.

14.11 Severability

If any provision of this Agreement shall be adjudged by any court of competent jurisdiction to be unenforceable or invalid, that provision shall be limited or eliminated to the minimum extent necessary so that this Agreement shall otherwise remain in full force and effect and enforceable.

IN WITNESS WHEREOF, the Parties hereto, intending to be legally bound hereby, have caused this Agreement to be executed by Licensees authorized representative installing the Licensed Software and accepting the terms hereof in connection therewith.
Appendix 1

1. Parts of the Licensed Software that are permitted for distribution in object code form only ("Redistributables") under this Agreement:

- The Licensed Software's essential and add-on libraries

- The Licensed Software's configuration tool ("qtconfig")

- The Licensed Software's help tool ("Qt Assistant")

- The Licensed Software's internationalization tools ("Qt Linguist", "lupdate", "lrelease")

- The Licensed Software's QML ("Qt Quick") launcher tool ("qmlscene" and "qmlviewer")

- The Licensed Software's installer framework

2. Parts of the Licensed Software that are not permitted for distribution include, but are not limited to:

- The Licensed Software's source code and header files

- The Licensed Software's documentation

- The Licensed Software's documentation generation tool ("qdoc")

- The Licensed Software's tool for writing makefiles ("qmake")

- The Licensed Software's Meta Object Compiler ("moc")

- The Licensed Software's User Interface Compiler ("uic" or in the case of Qt Jambi: "juic")

- The Licensed Software's Resource Compiler ("rcc")

- The Licensed Software's generator (only in the case of Qt Jambi if applicable)

- The Licensed Software's parts of the IDE tool ("Qt Creator")

- The Licensed Software's Emulator

- Build scripts, recipes and other material for creating the configuration of Licensed Software and/or 3rd party components, including the reference operating system configuration delivered in conjunction with the

The ISC License

Copyright (c) 2015, Contributors
Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2015, Rebecca Turner

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

NPM License Checker

[!]Build Status](https://www.travis-ci.org/davglass/license-checker.svg?branch=master)

*As of v17.0.0 the `failOn` and `onlyAllow` arguments take semicolons as delimeters instead of commas. Some license names contain commas and it messed with the parsing*

Ever needed to see all the license info for a module and its dependencies?

It's this easy:

```shell
cmpi -g license-checker
mkdir foo
cd foo
npm install yui-lint
license-checker
```
You should see something like this:

```
...
cli@0.4.3
   repository: http://github.com/chriso/cli
   licenses: MIT
glob@3.1.14
   repository: https://github.com/isaacs/node-glob
   licenses: UNKNOWN
graceful-fs@1.1.14
   repository: https://github.com/isaacs/node-graceful-fs
   licenses: UNKNOWN
inherits@1.0.0
   repository: https://github.com/isaacs/inherits
   licenses: UNKNOWN
jshint@0.9.1
   licenses: MIT
lru-cache@1.0.6
   repository: https://github.com/isaacs/node-lru-cache
   licenses: MIT
lru-cache@2.0.4
   repository: https://github.com/isaacs/node-lru-cache
   licenses: MIT
minimatch@0.0.5
   repository: https://github.com/isaacs/minimatch
   licenses: MIT
minimatch@0.2.9
   repository: https://github.com/isaacs/minimatch
   licenses: MIT
sigmund@1.0.0
   repository: https://github.com/isaacs/sigmund
   licenses: UNKNOWN
yui-lint@0.1.1
   licenses: BSD
   repository: http://github.com/yui/yui-lint
...
```

An asterisk next to a license name means that it was deduced from an other file than package.json (README, LICENSE, COPYING, ...)

You could see something like this:

```
...
debug@2.0.0
   repository: https://github.com/visionmedia/debug
   licenses: MIT*
```
Options
-------
* `--production` only show production dependencies.
* `--development` only show development dependencies.
* `--start [path of the initial json to look for]`
* `--unknown` report guessed licenses as unknown licenses.
* `--onlyunknown` only list packages with unknown or guessed licenses.
* `--json` output in json format.
* `--csv` output in csv format.
* `--csvComponentPrefix` prefix column for component in csv format.
* `--out [filepath]` write the data to a specific file.
* `--customPath` to add a custom Format file in JSON
* `--exclude [list]` exclude modules which licenses are in the comma-separated list from the output
* `--relativeLicensePath` output the location of the license files as relative paths
* `--summary` output a summary of the license usage.
* `--failOn [list]` fail (exit with code 1) on the first occurrence of the licenses of the semicolon-separated list
* `--onlyAllow [list]` fail (exit with code 1) on the first occurrence of the licenses not in the semicolon-seperated list
* `--packages [list]` restrict output to the packages (package@version) in the semicolon-seperated list
* `--excludePackages [list]` restrict output to the packages (package@version) not in the semicolon-seperated list
* `--excludePrivatePackages` restrict output to not include any package marked as private
* `--direct look for direct dependencies only`

Exclusions
----------
A list of licenses is the simplest way to describe what you want to exclude.

You can use valid [SPDX identifiers](https://spdx.org/licenses/).
You can use valid SPDX expressions like `MIT OR X11`.
You can use non-valid SPDX identifiers, like `Public Domain`, since `npm` does support some license strings that are not SPDX identifiers.

Examples
--------
```
license-checker --json > /path/to/licenses.json
license-checker --csv --out /path/to/licenses.csv
license-checker --unknown
license-checker --customPath customFormatExample.json
license-checker --exclude 'MIT, MIT OR X11, BSD, ISC'
license-checker --packages 'react@16.3.0;react-dom@16.3.0;lodash@4.3.1'
license-checker --excludePackages 'internal-1;internal-2'
license-checker --onlyunknown
```

Custom format
--------------
The `--customPath` option can be used with CSV to specify the columns. Note that the first column, `module_name`, will always be used.

When used with JSON format, it will add the specified items to the usual ones.

The available items are the following:
- name
- version
- description
- repository
- publisher
- email
- url
- licenses
- licenseFile
- licenseText
- licenseModified

You can also give default values for each item.
See an example in [customFormatExample.json](customFormatExample.json).

Requiring
---------
```js
var checker = require('license-checker');

checker.init({
  start: '/path/to/start/looking'
}, function(err, packages) {
  if (err) {
    //Handle error
  } else {
    //The sorted package data
    //as an Object
  }
});
```

Debugging
---------

license-checker uses [debug](https://www.npmjs.com/package/debug) for internal logging. There are two internal markers:

* `license-checker:error` for errors
* `license-checker:log` for non-errors

Set the `DEBUG` environment variable to one of these to see debug output:

```
```
```
$ export DEBUG=license-checker*; license-checker
scanning ./yui-lint
cli@0.4.3
  repository: http://github.com/chriso/cli
  licenses: MIT
# ...
```

How Licenses are Found
----------------------

We walk through the `node_modules` directory with the 
`[read-installed]`(https://www.npmjs.org/package/read-installed) module. Once we gathered a list of modules we walk through them and look at all of their `package.json`’s, We try to identify the license with the 
`[spdx]`(https://www.npmjs.com/package/spdx) module to see if it has a valid SPDX license attached. If that fails, we then look into the module for the following files: `LICENSE`, `LICENCE`, `COPYING`, & `README`.

If one of the those files are found (in that order) we will attempt to parse the license data from it with a list of known license texts. This will be shown with the `*` next to the name of the license to show that we "guessed" at it.

The Mesa header files use the following licenses.

================================================================================

The default Mesa license is as follows:

Copyright (C) 1999-2007 Brian Paul All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
GLES/glext.h, GLES/gl.h and GLES/glplatform.h use the following license:

SGI FREE SOFTWARE LICENSE B
(Version 2.0, Sept. 18, 2008)

Copyright (C) [dates of first publication] Silicon Graphics, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice including the dates of first publication and either this permission notice or a reference to http://oss.sgi.com/projects/FreeB/ shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL SILICON GRAPHICS, INC. BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Silicon Graphics, Inc. shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Silicon Graphics, Inc.

Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the names of the Mozilla Foundation nor the names of project contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*************************************************************************/

* IAccessible2 IDL Specification

* Copyright (c) 2007, 2010 Linux Foundation
* Copyright (c) 2006 IBM Corporation
* Copyright (c) 2000, 2006 Sun Microsystems, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above
*    copyright notice, this list of conditions and the following
*    disclaimer in the documentation and/or other materials
*    provided with the distribution.
*
* 3. Neither the name of the Linux Foundation nor the names of its
*    contributors may be used to endorse or promote products
*    derived from this software without specific prior written
*    permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
* CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
* INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
* OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
This BSD License conforms to the Open Source Initiative "Simplified BSD License" as published at:
http://www.opensource.org/licenses/bsd-license.php

IAccessible2 is a trademark of the Linux Foundation. The IAccessible2 mark may be used in accordance with the Linux Foundation Trademark Policy to indicate compliance with the IAccessible2 specification.

************************************************************************/
<!-- LICENSEFILE/ -->

<h1>License</h1>

Unless stated otherwise all works are:

<ul><li>Copyright &copy; 2013+ <a href="http://bevry.me">Bevry Pty Ltd</a></li></ul>

and licensed under:

<ul><li><a href="http://spdx.org/licenses/MIT.html">MIT License</a></li></ul>

<h2>MIT License</h2>

<pre>
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
</pre>

<!-- /LICENSEFILE -->

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and
attrition notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
The MIT License (MIT)

Copyright (c) 2014-2015, 2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)
Copyright (c) 2016 Peter Goldsborough

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Font license info

## Elusive

Copyright (C) 2013 by Aristeides Stathopoulos

Author: Aristeides Stathopoulos
License: SIL (http://scripts.sil.org/OFL)
Homepage: http://aristeides.com/
/*
 * $Id: pa_memorybarrier.h 1240 2007-07-17 13:05:07Z bjornroche $
 * Portable Audio I/O Library
 * Memory barrier utilities
 *
 * Author: Bjorn Roche, XO Audio, LLC
 *
 * This program uses the PortAudio Portable Audio Library.
 * For more information see: http://www.portaudio.com
 * Copyright (c) 1999-2000 Ross Bencina and Phil Burk
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files
 * (the "Software"), to deal in the Software without restriction,
 * including without limitation the rights to use, copy, modify, merge,
 * publish, distribute, sublicense, and/or sell copies of the Software,
 * and to permit persons to whom the Software is furnished to do so,
 * subject to the following conditions:
 *
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
* ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
* CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

/*
 * The text above constitutes the entire PortAudio license; however,
 * the PortAudio community also makes the following non-binding requests:
 *
 * Any person wishing to distribute modifications to the Software is
 * requested to send the modifications to the original developer so that
 * they can be incorporated into the canonical version. It is also
 * requested that these non-binding requests be included along with the
 * license above.
 */

/*
 * $Id: pa_ringbuffer.c 1421 2009-11-18 16:09:05Z bjornroche $
 * Portable Audio I/O Library
 * Ring Buffer utility.
 *
 * Author: Phil Burk, http://www.softsynth.com
 * modified for SMP safety on Mac OS X by Bjorn Roche
 * modified for SMP safety on Linux by Leland Lucius
 * also, allowed for const where possible
 * modified for multiple-byte-sized data elements by Sven Fischer
 *
 * Note that this is safe only for a single-thread reader and a
 * single-thread writer.
 *
 * This program uses the PortAudio Portable Audio Library.
 * For more information see: http://www.portaudio.com
 * Copyright (c) 1999-2000 Ross Bencina and Phil Burk
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files
 * (the "Software"), to deal in the Software without restriction,
 * including without limitation the rights to use, copy, modify, merge,
 * publish, distribute, sublicense, and/or sell copies of the Software,
 * and to permit persons to whom the Software is furnished to do so,
 * subject to the following conditions:
The Catharon Open Source LICENSE
---------------------------------

2000-Jul-04

Copyright (C) 2000 by Catharon Productions, Inc.

Introduction
-------------

This license applies to source files distributed by Catharon Productions, Inc. in several archive packages. This license applies to all files found in such packages which do not fall under their own explicit license.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- We don't promise that this software works. However, we are interested in any kind of bug reports. (as is' distribution)
You can use this software for whatever you want, in parts or full form, without having to pay us. (‘royalty-free’ usage)

You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the Catharon Code. (‘credits’)

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering the packages distributed by Catharon Productions, Inc. and assume no liability related to their use.

Legal Terms
===========

0. Definitions
--------------

Throughout this license, the terms `Catharon Package’, `package’, and `Catharon Code’ refer to the set of files originally distributed by Catharon Productions, Inc.

`You’ refers to the licensee, or person using the project, where `using’ is a generic term including compiling the project’s source code as well as linking it to form a `program’ or `executable’. This program is referred to as `a program using one of the Catharon Packages’.

This license applies to all files distributed in the original Catharon Package(s), including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The Catharon Packages are copyright (C) 2000 by Catharon Productions, Inc. All rights reserved except as specified below.

1. No Warranty
--------------

THE CATHARON PACKAGES ARE PROVIDED `AS IS’ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OF OR THE INABILITY TO USE THE CATHARON PACKAGE.

2. Redistribution

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the Catharon Packages (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

  o Redistribution of source code must retain this license file ('license.txt') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.

  o Redistribution in binary form must provide a disclaimer that states that the software is based in part on the work of Catharon Productions, Inc. in the distribution documentation.

These conditions apply to any software derived from or based on the Catharon Packages, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising

Neither Catharon Productions, Inc. and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use the following phrase to refer to this software in your documentation: ‘this software is based in part on the Catharon Typography Project’.

As you have not signed this license, you are not required to accept it. However, as the Catharon Packages are copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the Catharon Packages, you indicate that you understand and accept all the terms of this license.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright contributors
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

PCRE is a library of functions to support regular expressions whose syntax
and semantics are as close as possible to those of the Perl 5 language.

This is JavaScriptCore's variant of the PCRE library. While this library
started out as a copy of PCRE, many of the features of PCRE have been
removed.

Copyright (c) 1997-2005 University of Cambridge. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,
    this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright
    notice, this list of conditions and the following disclaimer in the
    documentation and/or other materials provided with the distribution.

  * Neither the name of the University of Cambridge nor the name of Apple
    Inc. nor the names of their contributors may be used to endorse or
    promote products derived from this software without specific prior
    written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2016, Vladimir Agafonkin
Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The XML testsuite available here is a copy of the Extensible Markup Language (XML) Conformance Test Suites provided by W3C. Please see http://www.w3.org/XML/Test/ for updates and other information.

These files are licensed under the W3C Software License (19980720), reproduced below:

---

W3C SOFTWARE NOTICE AND LICENSE


This W3C work (including software, documents, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code:

3. Notice of any changes or modifications to the W3C files, including
the date changes were made. (We recommend you provide URIs to the
location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT
HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR
DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS,
TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL
OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in
advertising or publicity pertaining to the software without specific,
written prior permission. Title to copyright in this software and any
associated documentation will at all times remain with copyright
holders.
The MIT License

Copyright (c) 2015 Kyle E. Mitchell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2006 Hewlett-Packard Development Company, L.P.
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the
Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Fonts are (c) Bitstream (see below). DejaVu changes are in public domain.
Glyphs imported from Arev fonts are (c) Tavmjong Bah (see below)

Bitstream Vera Fonts Copyright

----------------------------------------

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT,
TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOme
FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING
ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE
FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome
Foundation, and Bitstream Inc., shall not be used in advertising or
otherwise to promote the sale, use or other dealings in this Font Software
without prior written authorization from the Gnome Foundation or Bitstream
Inc., respectively. For further information, contact: fonts at gnome dot
org.

Arev Fonts Copyright

Copyright (c) 2006 by Tavmjong Bah. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of the fonts accompanying this license ("Fonts") and
associated documentation files (the "Font Software"), to reproduce
and distribute the modifications to the Bitstream Vera Font Software,
including without limitation the rights to use, copy, merge, publish,
distribute, and/or sell copies of the Font Software, and to permit
persons to whom the Font Software is furnished to do so, subject to
the following conditions:

The above copyright and trademark notices and this permission notice
shall be included in all copies of one or more of the Font Software
typefaces.

The Font Software may be modified, altered, or added to, and in
particular the designs of glyphs or characters in the Fonts may be
modified and additional glyphs or characters may be added to the
Fonts, only if the fonts are renamed to names not containing either
the words "Tavmjong Bah" or the word "Arev".

This License becomes null and void to the extent applicable to Fonts
or Font Software that has been modified and is distributed under the
"Tavmjong Bah Arev" names.

The Font Software may be sold as part of a larger software package but
no copy of one or more of the Font Software typefaces may be sold by
itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESSION OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL TAVMJONG BAH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the name of Tavmjong Bah shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from Tavmjong Bah. For further information, contact: tavmjong @ free.fr.

$Id: LICENSE 2133 2007-11-28 02:46:28Z lechimp $
Copyright (c) 2011 The LevelDB Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2015 Nathan Rajlich

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

See Terms of Use for definitions of Unicode Inc.’s Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.’S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2019 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in https://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale,
use or other dealings in these Data Files or Software without prior
written authorization of the copyright holder.
// Copyright 2019 The Chromium OS Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012-2017 by Ingvar Stepanyan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013 Kael Zhang <i@kael.me>, contributors http://kael.me/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2010 James Halliday (mail@substack.net)
Modified work Copyright 2014 Contributors (ben@npmjs.com)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2018, Gajus Kuizinas (http://gajus.com/)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Gajus Kuizinas (http://gajus.com/) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANUARY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The OpenGL Extension Wrangler Library
Copyright (C) 2002-2007, Milan Ikits <milan ikits@ieee org>
Copyright (C) 2002-2007, Marcelo E. Magallon <mmagallo@debian org>
Copyright (C) 2002, Lev Povalahev
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The name of the author may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Mesa 3-D graphics library
Version: 7.0

Copyright (C) 1999-2007 Brian Paul All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2007 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and/or associated documentation files (the
"Materials"), to deal in the Materials without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Materials, and to
permit persons to whom the Materials are furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

MIT License

Copyright (c) 2013 Julian Gruber <julian@juliangruber.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

MIT License

Copyright (c) 2017 Gary Hsu

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (C) 2007, 2013 Apple Inc. All rights reserved.
Copyright (C) Rich Moore. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY APPLE COMPUTER, INC. "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APPLE COMPUTER, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

string-unpack-code.js contains code from jQuery 1.2.1 - New Wave Javascript:

Copyright (c) 2007 John Resig (jquery.com)
Dual licensed under the MIT (MIT-LICENSE.txt)
and GPL (GPL-LICENSE.txt) licenses.

and

Copyright (c) 2004-2007, The Dojo Foundation
All Rights Reserved.

Licensed under the Academic Free License version 2.1 or above OR the modified BSD license. For more information on Dojo licensing, see:
http://dojotoolkit.org/community/licensing.shtml

and

Prototype 1.5 rc0
- Adapted from Ruby on Rails - http://dev.rubyonrails.org/browser/spinoffs/prototype/src
- By Lunarmedia, 06 August, 2006
- Available at (and packed with) JavascriptCompressor.com

string-tagcloud.js contains code from json.js 2007-10-10, placed in the Public Domain

string-base64.js contains code licensed under MPL 1.1 OR GPL 2.0 OR LGPL 2.1
The Initial Developer of the Original Code is Digital Creations 2, Inc.
Portions created by the Initial Developer are Copyright (C) 2000
the Initial Developer. All Rights Reserved.

Contributor(s):
* Martijn Pieters <mj@digicool.com> (original author)
* Samuel Sieb <samuel@sieb.net>

date-format-xparb.js is licensed under the LGPL v2.1.
Eclipse Public License - v 1.0
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.
"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special,
incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is
solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient’s rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient’s rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.
Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
Copyright (c) 2011-2015 Jake Luer jake@alogicalparadox.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2009 by the Jinja Team, see AUTHORS for more details.

Some rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* The names of the contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2013-2018 Petka Antonov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)
Copyright (c) 2015 Dmitry Ivanov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// Copyright (C) 2002-2013 The ANGLE Project Authors.
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of TransGaming Inc., Google Inc., 3DLabs Inc. Ltd., nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This Font Software is licensed under the SIL Open Font License, Version 1.1.

This license is copied below, and is also available with a FAQ at:
http://scripts.sil.org/OFL

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.
DEFINITIONS
"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS
Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be
distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

`cssmin.py` - A Python port of the YUI CSS compressor.

Copyright (c) 2010 Zachary Voase

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-------------------------------------------------------------------------------

This software contains portions of the YUI CSS Compressor, notably some regular expressions for reducing the size of CSS. The YUI Compressor source code can be found at <http://github.com/yui/yuicompressor>, and is licensed as follows:

> YUI Compressor Copyright License Agreement (BSD License)
> 
> Copyright (c) 2009, Yahoo! Inc.
> All rights reserved.
> 
> Redistribution and use of this software in source and binary forms,
> with or without modification, are permitted provided that the following
> conditions are met:
> 
> * Redistributions of source code must retain the above
> copyright notice, this list of conditions and the
> following disclaimer.
> 
> * Redistributions in binary form must reproduce the above
> copyright notice, this list of conditions and the
> following disclaimer in the documentation and/or other
> materials provided with the distribution.
> 
> * Neither the name of Yahoo! Inc. nor the names of its
> contributors may be used to endorse or promote products
> derived from this software without specific prior
> written permission of Yahoo! Inc.
> 
> THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
> AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
> IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.
Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.
MIT License

Copyright (c) 2017 itmeo

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
// Copyright (c) 2011 Google, Inc.

// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to deal
// in the Software without restriction, including without limitation the rights
// to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

Open Source Used In DNAC Platform Antelope 12111
Copyright 2018 The Immersive Web Community Group

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2000 - 2015 The Legion of the Bouncy Castle Inc. (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter, Ben Noordhuis, and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1992, 1993
The Regents of the University of California. All rights reserved.

Copyright (c) 2011 The FreeBSD Foundation
All rights reserved.
Portions of this software were developed by David Chisnall under sponsorship from the FreeBSD Foundation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS `AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The ISC License

Copyright (c) 2019 Elan Shanker, Paul Miller (https://paulmillr.com)

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Node.js is licensed for use as follows:

====

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====
This license applies to parts of Node.js originating from the https://github.com/joyent/node repository:

****

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

****

Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----------------------------------------------------------------

Notice that the following BSD-style license applies to this one file (valgrind.h) only. The rest of Valgrind is licensed under the terms of the GNU General Public License, version 2, unless otherwise indicated. See the COPYING file in the source distribution for details.

----------------------------------------------------------------
This file is part of Valgrind, a dynamic binary instrumentation framework.

Copyright (C) 2000-2010 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Notice that the above BSD-style license applies to this one file (valgrind.h) only. The entire rest of Valgrind is licensed under the terms of the GNU General Public License, version 2. See the COPYING file in the source distribution for details.

The MIT License (MIT)

Copyright (c) 2018 Tobias Reich
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This package was debianized by gperftools Contributors <google-perftools@googlegroups.com> on Sat, 20 Jul 2013 14:21:10 -0700.

It was downloaded from http://code.google.com/p/gperftools/downloads/list

Upstream Author: google-perftools@googlegroups.com

Copyright (c) 2005, Google Inc.
All rights reserved.

Copyright (c) 2013, gperftools Contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a
covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.
5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by
When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.
Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all. For example, if you agree to terms that oblige you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work. The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy’s
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.

For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Copyright (c) 2013, Joel Feenstra
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the ESQuery nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JOEL FEENSTRA BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

--------------------------------------------------------------------------------
Copyright (c) 2011-2015 Jake Luer jake@alogicalparadox.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

----

This library bundles a version of the `fs.realpath` and `fs.realpathSync`
methods from Node.js v0.10 under the terms of the Node.js MIT license.

Node's license follows, also included at the header of `old.js` which contains the licensed code:

Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at 

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software 
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.

#!/usr/bin/env node

/*
Copyright (c) 2013, Yahoo! Inc. All rights reserved.
Code licensed under the BSD License:
http://yuilibrary.com/license/
*/

var checker = require('../lib/index');

var args = require('../lib/args').parse();
var mkdirp = require('mkdirp');
var path = require('path');
var chalk = require('chalk');
var fs = require('fs');

if (args.help) {
  console.error('license-checker@' + require('../package.json').version);
  var usage = [
    '',
    '  --production only show production dependencies.',
    '  --development only show development dependencies.',
    '  --unknown report guessed licenses as unknown licenses.',
    '  --start [path of the initial json to look for]',
    '  --onlyunknown only list packages with unknown or guessed licenses.',
    '  --json output in json format.',
    '  --csv output in csv format.',
    '  --csvComponentPrefix column prefix for components in csv file',
    '  --out [filepath] write the data to a specific file.',
    '  --customPath to add a custom Format file in JSON',
    '  --exclude [list] exclude modules which licenses are in the comma-separated list from the output',
    '  --relativeLicensePath output the location of the license files as relative paths',
    '  --summary output a summary of the license usage',
    '  --failOn [list] fail (exit with code 1) on the first occurrence of the licenses of the semicolon-separated list',
    '  --onlyAllow [list] fail (exit with code 1) on the first occurrence of the licenses not in the semicolon-seperated list',
    '  --direct look for direct dependencies only',
    '  --packages [list] restrict output to the packages (package@version) in the semicolon-seperated list',
  ];
  console.error(usage.join('
    '));
} else {
  console.error(checker(args));
}
' --excludePackages [list] restrict output to the packages (package@version) not in the semicolon-separated list',
' --excludePrivatePackages restrict output to not include any package marked as private',
' --version The current version',
' --help The text you are reading right now :)',
''
];
console.error(usage.join('
'), '
');
process.exit(0);

if (args.version) {
    console.error(require('../package.json').version);
    process.exit(1);
}

if (args.failOn && args.onlyAllow) {
    console.error('--failOn and --onlyAllow can not be used at the same time. Choose one or the other.');
    process.exit(1);
} else {
    var argValue = args.failOn || args.onlyAllow;
    if (argValue && argValue.indexOf(',') >= 0) {
        var argName = args.failOn ? 'failOn' : 'onlyAllow';
        console.warn('Warning: As of v17 the --' + argName + ' argument takes semicolons as delimiters instead of commas (some license names can contain commas));
    }
}

checker.init(args, function(err, json) {

    var formattedOutput = "";

    if (!err) {
        console.error('Found error');
        console.error(err);
    }

    if (shouldColorizeOutput(args)) {
        var keys = Object.keys(json);
        keys.forEach(function(key) {
            var keyParts = key.split('@');
            var colorizedKey = chalk.blue(keyParts[0]) + chalk.dim('@') + chalk.green(keyParts[1]);
            json[colorizedKey] = json[key];
            delete json[key];
        });
    }
}

Open Source Used In DNAC Platform Antelope  12140
if (args.json) {
    formattedOutput = JSON.stringify(json, null, 2) + '\n';
} else if (args.csv) {
    formattedOutput = checker.asCSV(json, args.customFormat, args.csvComponentPrefix);
} else if (args.markdown) {
    formattedOutput = checker.asMarkDown(json, args.customFormat); + '\n';
} else if (args.summary) {
    formattedOutput = checker.asSummary(json);
} else {
    formattedOutput = checker.asTree(json);
}

if (args.files) {
    checker.asFiles(json, args.files);
} else if (args.out) {
    var dir = path.dirname(args.out);
    mkdirp.sync(dir);
    fs.writeFileSync(args.out, formattedOutput, 'utf8');
} else {
    console.log(formattedOutput);
}

function shouldColorizeOutput(args) {
    return args.color && !args.out && !(args.csv || args.json || args.markdown);
}

The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Uranus Ring:
Used under the terms of the license at http://www.jpl.nasa.gov/imagepolicy/
MIT License

Copyright (c) 2017 crypto-browserify
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2013 Roman Shtylman <shtylman@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Daniel Frey

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Copyright (c) 2014 Yehuda Katz, Tom Dale, Stefan Penner and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
The MIT License (MIT)

Copyright (c) 2014 Jameson Little

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright 2007-2009 Google Inc.
Copyright 2007-2009 WebDriver committers

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
The text and information contained in this file may be freely used,
copied, or distributed without compensation or licensing restrictions.

// Copyright (c) 2014 The Polymer Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2017, Google Inc.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
Unless required by applicable law or agreed to in writing, software
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.
You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

// Copyright 2016 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
//   notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
//   copyright notice, this list of conditions and the following disclaimer
//   in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENSE

Lipi core toolkit is licensed under the MIT license.

- MIT-Style License
  http://www.opensource.org/licenses/mit-license.php

This is the weakest license, meaning it applies the least restrictions. It is used for most parts of the Lipi Toolkit, and especially for libraries which are potentially included (by copying) into other libraries (as sub-components). The intention is to make sure the covered software can be used really everywhere and for every reason without any
real restrictions.
Copyright (c) 2007-2014, Alexandru Marasteanu <hello[at]alexei[dot]ro>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
* Neither the name of this software nor the names of its contributors may be
  used to endorse or promote products derived from this software without
  specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
MPL-1.1 / GPL-2.0 / LGPL-2.1
================================

SkGifImageReader.cpp and SkGifImageReader.h:

***** BEGIN LICENSE BLOCK *****
Version: MPL 1.1/GPL 2.0/LGPL 2.1

The contents of this file are subject to the Mozilla Public License Version
1.1 (the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis,
WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
for the specific language governing rights and limitations under the
License.

The Original Code is mozilla.org code.

The Initial Developer of the Original Code is
Netscape Communications Corporation.
Portions created by the Initial Developer are Copyright (C) 1998
the Initial Developer. All Rights Reserved.

Contributor(s):
Chris Saari <saari@netscape.com>
Apple Computer

Alternatively, the contents of this file may be used under the terms of either the GNU General Public License Version 2 or later (the "GPL"), or the GNU Lesser General Public License Version 2.1 or later (the "LGPL"), in which case the provisions of the GPL or the LGPL are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of either the GPL or the LGPL, and not to allow others to use your version of this file under the terms of the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL or the LGPL. If you do not delete the provisions above, a recipient may use your version of this file under the terms of any one of the MPL, the GPL or the LGPL.

***** END LICENSE BLOCK ***** */

BSD-3-Clause
============

libgifcodec.gni, SkGifCodec.h, SkLibGifCodec.cpp, SkLibGifCodec.h:

Copyright 2019 Google LLC. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
BSD-2-Clause
============

SkLibGifCodec.cpp:

Copyright (C) 2006 Apple Computer, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY APPLE COMPUTER, INC. "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APPLE COMPUTER, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The BSD License)

Copyright (c) 2010-2014, Christian Johansen, christian@cjohansen.no
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Christian Johansen nor the names of his contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright 2013 The Chromium Authors. All rights reserved.
/
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2004, 2005, 2006 Ero Carrera <ero@dkbza.org>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/* ***** BEGIN LICENSE BLOCK *****
* Version: MPL 1.1/GPL 2.0/LGPL 2.1
*
* The contents of this file are subject to the Mozilla Public License Version
* 1.1 (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
* http://www.mozilla.org/MPL/
*
* Software distributed under the License is distributed on an "AS IS" basis,
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
* for the specific language governing rights and limitations under the
* License.
*
* The Original Code is mozilla.org code.
*
* The Initial Developer of the Original Code is
* Netscape Communications Corporation.
* Portions created by the Initial Developer are Copyright (C) 2002
* the Initial Developer. All Rights Reserved.
*
* Contributor(s):
*
* Alternatively, the contents of this file may be used under the terms of
* either the GNU General Public License Version 2 or later (the "GPL"), or
* the GNU Lesser General Public License Version 2.1 or later (the "LGPL"),
* in which case the provisions of the GPL or the LGPL are applicable instead
* of those above. If you wish to allow use of your version of this file only
* under the terms of either the GPL or the LGPL, and not to allow others to
* use your version of this file under the terms of the MPL, indicate your
* decision by deleting the provisions above and replace them with the notice
* and other provisions required by the GPL or the LGPL. If you do not delete
* the provisions above, a recipient may use your version of this file under
* the terms of any one of the MPL, the GPL or the LGPL.
*
* ***** END LICENSE BLOCK ***** */

QT LICENSE AGREEMENT

Agreement version 4.2.3

This Qt License Agreement ("Agreement") is a legal agreement for the licensing
of Licensed Software (as defined below) between The Qt Company (as defined
below) and the Licensee who has accepted the terms of this Agreement by
downloading or using the Licensed Software and/or as defined herein:

Capitalized terms used herein are defined in Section 1.

WHEREAS:

(A) Licensee wishes to use the Licensed Software for the purpose of
developing and distributing Applications and/or Devices (each as
defined below); and

(B) The Qt Company is willing to grant the Licensee a right to use
Licensed Software for such a purpose pursuant to term and conditions
of this Agreement.

NOW, THEREFORE, THE PARTIES HEREBY AGREE AS FOLLOWS:

1. DEFINITIONS

"Affiliate" of a Party shall mean an entity (i) which is directly
or indirectly controlling such Party; (ii) which is under the same direct or
indirect ownership or control as such Party; or (iii) which is directly or
indirectly owned or controlled by such Party. For these purposes, an entity
shall be treated as being controlled by another if that other entity has fifty
percent (50 %) or more of the votes in such entity, is able to direct its
affairs and/or to control the composition of its board of directors or
equivalent body.

"Add-on Products" shall mean The Qt Company's specific add-on software products
(for example Qt Safe Renderer, Qt for Automation, Qt Application Manager), which
are not licensed as part of The Qt Company's standard offering, but shall be
included into the scope of Licensed Software only if so specifically agreed
between the Parties.

"Applications" shall mean Licensee's software products created using the
Licensed Software, which may include the Redistributables, or part thereof.
"Contractor(s)" shall mean third party consultants, distributors and contractors performing services to the Licensee under applicable contractual arrangement.

"Customer(s)" shall mean Licensee's end users to whom Licensee, directly or indirectly, distributes copies of the Redistributables.

"Data Protection Legislation" shall mean the General Data Protection Regulation (EU 2016/679) (GDPR) and any national implementing laws, regulations and secondary legislation, as may be amended or updated from time to time, as well as any other data protection laws or regulations applicable in relevant territory.

"Deployment Platforms" shall mean operating systems and/or hardware specified in the License Certificate, on which the Redistributables can be distributed pursuant to the terms and conditions of this Agreement.

"Designated User(s)" shall mean the employee(s) of Licensee or Licensee's Affiliates acting within the scope of their employment or Licensee's Contractors acting within the scope of their services for Licensee and on behalf of Licensee. Designated Users shall be named in the License Certificate.

"Development License" shall mean the license needed by the Licensee for each Designated User to use the Licensed Software under the license grant described in Section 3.1 of this Agreement. Development Licenses are available per respective Licensed Software products, each product having its designated scope and purpose of use. Distribution Licenses are always connected to Qt for Device Creation product and Qt for MCUs product only.

"Development Platforms" shall mean those operating systems specified in the License Certificate, in which the Licensed Software can be used under the Development License, but not distributed in any form or used for any other purpose.

"Devices" shall mean hardware devices or products that 1) are manufactured and/or distributed by the Licensee or its Affiliates or Contractors, and (2)(i) incorporate or integrate the Redistributables or parts thereof; or (ii) where the main user interface or substantial functionality of such unit, when used by a Customer, is provided by Application(s) or otherwise depends on the Licensed Software, regardless of whether the Redistributables are distributed together with the hardware or not. Devices covered with this Agreement shall be specified in Appendix 2 or in a quote.

"Distribution License(s)" shall mean the license required for any kind of sale, trade, exchange, loan, lease, rental or other distribution by or on behalf of Licensee to a third party of Redistributables in connection with Devices pursuant to license grant described in Section 3.3 of this Agreement.

"Distribution License Packs" shall mean set of prepaid Distribution Licenses for distribution of Redistributables, as defined in The Qt Company's standard price...
"Intellectual Property Rights" shall mean patents (including utility models), design patents, and designs (whether or not capable of registration), chip topography rights and other like protection, copyrights, trademarks, service marks, trade names, logos or other words or symbols and any other form of statutory protection of any kind and applications for any of the foregoing as well as any trade secrets.

"License Certificate" shall mean a certificate generated by The Qt Company for each Designated User respectively upon them downloading the Licensed Software, which will be available under respective Designated User's Qt Account at account.qt.io. License Certificates will specify the Designated User, the Development Platforms, Deployment Platforms and the License Term. Such terms are considered part of the licenses granted hereunder and shall be updated from time to time to reflect any agreed changes to the foregoing terms relating to Designated User's rights to the Licensed Software.

"License Fee" shall mean the fee charged to the Licensee for rights granted under the terms of this Agreement.

"License Term" shall mean the agreed validity period of the Development License of the respective Designated User, during which time the Designated User is entitled to use the Licensed Software, as set forth in the respective License Certificate.

"Licensed Software" shall mean either

(i) Qt for Application Development or
(ii) Qt for Device Creation, and/or
(iii) Qt 3D Studio, and/or
(iv) Qt Design Studio, and/or
(v) Qt for MCUs, and/or
(vi) selected Add-on Products, if any,

depending on which product(s) the Licensee has purchased under this Agreement, as well as corresponding online or electronic documentation, associated media and printed materials, including the source code (where applicable), example programs and the documentation, licensed to the Licensee under this Agreement. Licensed Software does not include Third Party Software (as defined in Section 4) or Open Source Qt. The Qt Company may, in the course of its development activities, at its free and absolute discretion and without any obligation to send or publish any notifications to the Licensee or in general, make changes, additions or deletions in the components and functionalities of the Licensed Software, provided that no such changes, additions or deletions will affect the already released version of the Licensed Software, but only upcoming version(s).
"Licensee" shall mean the individual or legal entity that is party to this Agreement, as identified on the signature page hereof.

"Licensee's Records" shall mean books and records that are likely to contain information bearing on Licensee's compliance with this Agreement, Licensee's use of Open Source Qt and/or the payments due to The Qt Company under this Agreement, including, but not limited to user information, assembly logs, sales records and distribution records.

"Modified Software" shall have the meaning as set forth in Section 2.3.

"Online Services" shall mean any services or access to systems made available by The Qt Company to the Licensee over the Internet relating to the Licensed Software or for the purpose of use by the Licensee of the Licensed Software or Support. Use of any such Online Services is discretionary for the Licensee and some of them may be subject to additional fees.

"Open Source Qt" shall mean the non-commercial Qt computer software products, licensed under the terms of the GNU Lesser General Public License, version 2.1 or later ("LGPL") or the GNU General Public License, version 2.0 or later ("GPL"). For clarity, Open Source Qt shall not be provided nor governed under this Agreement.

"Party" or "Parties" shall mean Licensee and/or The Qt Company.

"Permitted Software" shall mean any (i) open source software (excluding Open Source Qt) that is a) generally available for public in source code form without additional cost under any of the licenses approved by Open Source Initiative as listed on https://opensource.org/licenses; and (b) is in no way, directly or indirectly, developed by or for or otherwise related to or in the interest of the Licensee or its Affiliates, or (ii) software The Qt Company has made available via its Qt Marketplace online distribution channel.

"Pre-Release Code" shall have the meaning as set forth in Section 4.

"Prohibited Combination" shall mean any means to (i) use, combine, incorporate, link or integrate Licensed Software with any software created with or incorporating Open Source Qt, (ii) use Licensed Software for creation of any software created with or incorporating Open Source Qt, or (iii) incorporate or integrate Applications into a hardware device or product other than a Device.

"Qt 3D Studio" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.

"Qt Design Studio" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.
"Qt for Application Development" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.

"Qt for Device Creation" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.

"Qt for MCUs" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.

"Redistributables" shall mean the portions of the Licensed Software set forth in Appendix 1 that may be distributed pursuant to the terms of this Agreement in object code form only, including any relevant documentation. Where relevant, any reference to Licensed Software in this Agreement shall include and refer also to Redistributables.

"Renewal Term" shall mean an extension of previous License Term as agreed between the Parties.

"Submitted Modified Software" shall have the meaning as set forth in Section 2.3.

"Support" shall mean standard developer support that is provided by The Qt Company to assist Designated Users in using the Licensed Software in accordance with The Qt Company's standard support terms available at https://www.qt.io/terms-conditions/ and as further defined in Section 8 hereunder.

"Taxes" shall have the meaning set forth in Section 10.5.

"Term" shall have the meaning set forth in Section 12.

"The Qt Company" shall mean:
(i) in the event Licensee is an individual residing in the United States or a legal entity incorporated in the United States or having its headquarters in the United States, The Qt Company Inc., a Delaware corporation with its office at 2350 Mission College Blvd., Suite 1020, Santa Clara, CA 95054, USA.; or
(ii) in the event the Licensee is an individual residing outside of the United States or a legal entity incorporated outside of the United States or having its registered office outside of the United States, The Qt Company Ltd., a Finnish company with its registered office at Bertel Jungin aukio D3A, 02600 Espoo, Finland.

"Third-Party Software" shall have the meaning set forth in Section 4.

"Updates" shall mean a release or version of the Licensed Software containing bug fixes, error corrections and other changes that are generally made available
to users of the Licensed Software that have contracted for Support. Updates are generally depicted as a change to the digits following the decimal in the Licensed Software version number. The Qt Company shall make Updates available to the Licensee under the Support. Updates shall be considered as part of the Licensed Software hereunder.

"Upgrades" shall mean a release or version of the Licensed Software containing enhancements and new features and are generally depicted as a change to the first digit of the Licensed Software version number. In the event Upgrades are provided to the Licensee under this Agreement, they shall be considered as part of the Licensed Software hereunder.

2. OWNERSHIP

2.1. Ownership of The Qt Company
The Licensed Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Licensed Software is licensed, not sold.

All of The Qt Company's Intellectual Property Rights are and shall remain the exclusive property of The Qt Company or its licensors respectively.

2.2. Ownership of Licensee
All the Licensee's Intellectual Property Rights are and shall remain the exclusive property of the Licensee or its licensors respectively.

All Intellectual Property Rights to the Modified Software, Applications and Devices shall remain with the Licensee and no rights thereto shall be granted by the Licensee to The Qt Company under this Agreement (except as set forth in Section 2.3 below).

2.3. Modified Software
Licensee may create bug-fixes, error corrections, patches or modifications to the Licensed Software ("Modified Software"). Such Modified Software may break the source or binary compatibility with the Licensed Software (including without limitation through changing the application programming interfaces ("API") or by adding, changing or deleting any variable, method, or class signature in the Licensed Software and/or any inter-process protocols, services or standards in the Licensed Software libraries). To the extent that Licensee's Modified Software so breaks source or binary compatibility with the Licensed Software, Licensee acknowledges that The Qt Company's ability to provide Support may be prevented or limited and Licensee's ability to make use of Updates may be restricted.

Licensee may, at its sole and absolute discretion, choose to submit Modified Software to The Qt Company ("Submitted Modified Software") in connection with Licensee's Support request, service request or otherwise. In the event Licensee does so, then, Licensee hereby grants The Qt Company a sublicensable, assignable, irrevocable, perpetual, worldwide, non-exclusive, royalty-free and
fully paid-up license, under all of Licensee's Intellectual Property Rights, to reproduce, adapt, translate, modify, and prepare derivative works of, publicly display, publicly perform, sublicense, make available and distribute such Submitted Modified Software as The Qt Company sees fit at its free and absolute discretion.

3. LICENSES GRANTED

3.1. Development with Licensed Software
Subject to the terms of this Agreement, The Qt Company grants to Licensee a worldwide, non-exclusive, non-transferable license, valid for the License Term, to use, modify and copy the Licensed Software by Designated Users on the Development Platforms for the sole purposes of designing, developing, demonstrating and testing Application(s) and/or Devices, and to provide thereto related support and other related services to end-user Customers. Each Application and/or Device can only include, incorporate or integrate code generated under this Agreement by such Designated Users who are duly licensed for the applicable Development Platform(s) and Deployment Platform(s) (i.e have a valid license for the appropriate Licensed Software product).

Licensee may install copies of the Licensed Software on five (5) computers per Designated User, provided that only the Designated Users who have a valid Development License may use the Licensed Software.

Licensee may at any time designate another Designated User to replace a then-current Designated User by notifying The Qt Company in writing, provided that any Designated User may be replaced only once during any six-month period.

Upon expiry of the initially agreed License Term, the respective License Terms shall be automatically extended to one or more Renewal Term(s), unless and until either Party notifies the other Party in writing that it does not wish to continue the License Term, such notification to be provided to the other Party no less than thirty (30) days before expiry of the respective License Term.

Unless otherwise agreed between the Parties, Renewal Term shall be of equal length with the initial License Term.

Any such Renewal Term shall be subject to License Fees agreed between the Parties or, if no advance agreement exists, subject to The Qt Company's standard pricing applicable at the commencement date of any such Renewal Term.

Any price or other term specified for a Renewal Term shall be valid only for the specified time.

The Qt Company may request the Licensee to place a purchase order corresponding to a quote by The Qt Company for the relevant Renewal Term.
In the event Licensee does not prevent auto-renewal pursuant the above, but a Renewal Term is nevertheless not duly ordered within 30 days from the date of the respective notification from The Qt Company and/or the respective License Fee paid by due date specified in The Qt Company's respective invoice, The Qt Company shall apply a reinstatement fee equal to ten percent (10%) of the total value of the License Fees of the Development Licenses for the expired term to be added to the License Fee of the respective Renewal Term.

In the event Licensee chooses not to renew a Development License for a Renewal Term by notifying The Qt Company thereof no less than thirty (30) days before expiry of the respective License Term, Licensee may still reinstate such expired Development Licenses for a Renewal Term subject to applicable renewal Term License Fees until thirty (30) days from the expiry of the initially agreed License Term or preceding Renewal Term. After such thirty (30) day period a Development License shall be subject to applicable License Fees for a new Development License and not any Renewal Term License Fees.

3.2. Distribution of Applications
Subject to the terms of this Agreement, The Qt Company grants to Licensee a worldwide, non-exclusive, non-transferable, revocable (for cause pursuant to this Agreement) right and license, valid for the Term, to

(i) distribute, by itself or through its Contractors, Redistributables as installed, incorporated or integrated into Applications for execution on the Deployment Platforms, and

(ii) grant sublicenses to Redistributables, as distributed hereunder, for Customers solely for Customer's internal use and to the extent necessary in order for the Customers to use the Applications for their respective intended purposes.

Right to distribute the Redistributables as part of an Application as provided herein is not royalty-bearing but is conditional upon the Licensee not having any unpaid License Fees for Development Licenses owed to The Qt Company at the time of distribution of any Redistributables to Customers.

3.3. Distribution of Devices
Subject to the terms of this Agreement, The Qt Company grants to Licensee a worldwide, non-exclusive, non-transferable, revocable (for cause pursuant to this Agreement) right and license, valid for the Term, to

(i) distribute, by itself or through one or more tiers of Contractors, Redistributables as installed, incorporated or integrated, or intended to be installed, incorporated or integrated into Devices for execution on the Deployment Platforms, and

(ii) grant sublicenses to Redistributables, as distributed hereunder, for Customers solely for Customer's internal use and to the extent necessary in order for the Customers to use the Devices for their respective intended purposes.

Right to distribute the Redistributables with Devices as provided herein is
3.4. Further Requirements

The licenses granted above in this Section 3 by The Qt Company to Licensee are conditional and subject to Licensee's compliance with the following terms:

(i) Licensee shall not remove or alter any copyright, trademark or other proprietary rights notice(s) contained in any portion of the Licensed Software;

(ii) Applications must add primary and substantial functionality to the Licensed Software so as not to compete with the Licensed Software;

(iii) Applications may not pass on functionality which in any way makes it possible for others to create software with the Licensed Software; provided however that Licensee may use the Licensed Software's scripting and QML ("Qt Quick") functionality solely in order to enable scripting, themes and styles that augment the functionality and appearance of the Application(s) without adding primary and substantial functionality to the Application(s);

(iv) Licensee shall not use Licensed Software in any manner or for any purpose that infringes, misappropriates or otherwise violates any Intellectual property or right of any third party, or that violates any applicable law;

(v) Licensee shall not use The Qt Company's or any of its suppliers' names, logos, or trademarks to market Applications, except that Licensee may use "Built with Qt" logo to indicate that Application(s) was developed using the Licensed Software;

(vi) Licensee shall not distribute, sublicense or disclose source code of Licensed Software to any third party (provided however that Licensee may appoint employee(s) of Contractors as Designated Users to use Licensed Software pursuant to this Agreement). Such right may be available for the Licensee subject to a separate software development kit ("SDK") license agreement to be concluded with The Qt Company;

(vii) Licensee shall not grant the Customers a right to (i) make copies of the Redistributables except when and to the extent required to use the Applications and/or Devices for their intended purpose, (ii) modify the Redistributables or create derivative works thereof, (iii) decompile, disassemble or otherwise reverse engineer Redistributables, or (iv) redistribute any copy or portion of the Redistributables to any third party, except as part of the onward sale of the Device on which the Redistributables are installed;

(viii) Licensee shall not and shall cause that its Affiliates or Contractors shall not use Licensed Software in any Prohibited Combination, unless Licensee has received an advance written permission from The Qt Company to do so. Absent such written permission, any and all distribution by the Licensee during the Term...
of a hardware device or product a) which incorporate or integrate any part of Licensed Software or Open Source Qt; or b) where the main user interface or substantial functionality is provided by software built with Licensed Software or Open Source Qt or otherwise depends on the Licensed Software or Open Source Qt, shall be considered to be Device distribution under this Agreement and shall be dependent on Licensee's compliance thereof (including but not limited to obligation to pay applicable License Fees for such distribution). Notwithstanding what is provided above in this sub-section (viii), Licensee is entitled to use and combine Licensed Software with any Permitted Software;

(ix) Licensee shall cause all of its Affiliates and Contractors entitled to make use of the licenses granted under this Agreement, to be contractually bound to comply with the relevant terms of this Agreement and not to use the Licensed Software beyond the terms hereof and for any purposes other than operating within the scope of their services for Licensee. Licensee shall be responsible for any and all actions and omissions of its Affiliates and Contractors relating to the Licensed Software and use thereof (including but not limited to payment of all applicable License Fees);

(x) Except when and to the extent explicitly provided in this Section 3, Licensee shall not transfer, publish, disclose, display or otherwise make available the Licensed Software; and

(xi) Licensee shall not attempt or enlist a third party to conduct or attempt to conduct any of the above.

Above terms shall not be applicable if and to the extent they conflict with any mandatory provisions of any applicable laws.

Any use of Licensed Software beyond the provisions of this Agreement is strictly prohibited and requires an additional license from The Qt Company.

4. THIRD-PARTY SOFTWARE

The Licensed Software may provide links to third party libraries or code (collectively "Third-Party Software") to implement various functions. Third-Party Software does not comprise part of the Licensed Software. In some cases, access to Third-Party Software may be included with the Licensed Software. Such Third-Party Software will be listed in the "/src/3rdparty" source tree delivered with the Licensed Software or documented in the Licensed Software, as such may be amended from time to time. Licensee acknowledges that use or distribution of Third-Party Software is in all respects subject to applicable license terms of applicable third-party right holders.

5. PRE-RELEASE CODE

The Licensed Software may contain pre-release code and functionality, or sample code marked or otherwise stated with appropriate designation such as "Technology Preview", "Alpha", "Beta", "Sample" etc. ("Pre-Release Code").
Such Pre-Release Code may be present complimentary for the Licensee, in order to provide experimental support or information for new platforms or preliminary versions of one or more new functionalities or for other similar reasons. The Pre-Release Code may not be at the level of performance and compatibility of a final, generally available, product offering. The Pre-Release Code may not operate correctly, may contain errors and may be substantially modified by The Qt Company prior to the first commercial product release, if any. The Qt Company is under no obligation to make Pre-Release Code commercially available, or provide any Support or Updates relating thereto. The Qt Company assumes no liability whatsoever regarding any Pre-Release Code, but any use thereof is exclusively at Licensee’s own risk and expense.

For clarity, unless Licensed Software specifies different license terms for the respective Pre-Release Code, the Licensee is entitled to use such pre-release code pursuant to Section 3, just like other Licensed Software, provided however that in the event Add-on Products are included and available as such Pre-Release Code, Licensee’s right to use such Add-on Products is nevertheless subject to and conditional upon conclusion of separate agreement with The Qt Company.

6. LIMITED WARRANTY AND WARRANTY DISCLAIMER

The Qt Company hereby represents and warrants that it has the power and authority to grant the rights and licenses granted to Licensee under this Agreement.

Except as set forth above, the Licensed Software is licensed to Licensee "as is" and Licensee's exclusive remedy and The Qt Company's entire liability for errors in the Licensed Software shall be limited, at The Qt Company's option, to correction of the error, replacement of the Licensed Software or return of the applicable fees paid for the defective Licensed Software for the time period during which the License is not able to utilize the Licensed Software under the terms of this Agreement.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE QT COMPANY ON BEHALF OF ITSELF AND ITS LICENSORS, SUPPLIERS AND AFFILIATES, DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT WITH REGARD TO THE LICENSED SOFTWARE. THE QT COMPANY DOES NOT WARRANT THAT THE LICENSED SOFTWARE WILL SATISFY LICENSEE'S REQUIREMENTS OR THAT IT WILL OPERATE WITHOUT DEFECT OR ERROR OR THAT THE OPERATION THEREOF WILL BE UNINTERRUPTED.

7. INDEMNIFICATION AND LIMITATION OF LIABILITY

7.1. Limitation of Liability
EXCEPT FOR (I) CASES OF GROSS NEGLIGENCE OR INTENTIONAL MISCONDUCT, AND (II)
BREACH OF CONFIDENTIALITY, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL EITHER PARTY BE LIABLE TO THE OTHER PARTY FOR ANY LOSS OF PROFIT, LOSS OF DATA, LOSS OF BUSINESS OR GOODWILL OR ANY OTHER INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE COST, DAMAGES OR EXPENSE OF ANY KIND, HOWSOEVER ARISING UNDER OR IN CONNECTION WITH THIS AGREEMENT.

EXCEPT FOR (I) CASES OF GROSS NEGLIGENCE OR INTENTIONAL MISCONDUCT, AND (II) BREACH OF CONFIDENTIALITY, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL EITHER PARTY’S TOTAL AGGREGATE LIABILITY UNDER THIS AGREEMENT EXCEED THE AGGREGATE LICENSE FEES PAID OR PAYABLE TO THE QT COMPANY FROM LICENSEE DURING THE PERIOD OF TWELVE (12) MONTHS IMMEDIATELY PRECEDING THE EVENT RESULTING IN SUCH LIABILITY.

THE PROVISIONS OF THIS SECTION 7 ALLOCATE THE RISKS UNDER THIS AGREEMENT BETWEEN THE QT COMPANY AND LICENSEE AND THE PARTIES HAVE RELIED UPON THE LIMITATIONS SET FORTH HEREIN IN DETERMINING WHETHER TO ENTER INTO THIS AGREEMENT.

NOTWITHSTANDING ANYTHING TO THE CONTRARY IN THIS AGREEMENT, LICENSEE SHALL ALWAYS BE LIABLE TO PAY THE APPLICABLE LICENSE FEES CORRESPONDING TO ITS ACTUAL USE OF LICENSED SOFTWARE.

8. SUPPORT, UPDATES AND ONLINE SERVICES

Upon due payment of the agreed License Fees the Licensee will be eligible to receive Support and Updates and to use the Online Services during the License Term, provided, however, that in the event the License Term is longer than 36 months, the initial payment includes Support for only the first 12 months, unless the Parties specifically otherwise agree.

Unless otherwise decided by The Company at its free and absolute discretion, Upgrades will not be included in the Support but may be available subject to additional fees.

From time to time The Qt Company may change the Support terms, provided that during the respective ongoing License Term the level of Support provided by The Qt Company may not be reduced without the consent of the Licensee.

Unless otherwise agreed, The Qt Company shall not be responsible for providing any service or support to Customers.

9. CONFIDENTIALITY

Each Party acknowledges that during the Term of this Agreement each Party may receive information about the other Party’s business, business methods, business plans, customers, business relations, technology, and other information, including the terms of this Agreement, that is confidential and of great value to the other Party, and the value of which would be
significantly reduced if disclosed to third parties ("Confidential Information"). Accordingly, when a Party (the "Receiving Party") receives Confidential Information from the other Party (the "Disclosing Party"), the Receiving Party shall only disclose such information to employees and Contractors on a need to know basis, and shall cause its employees and employees of its Affiliates to: (i) maintain any and all Confidential Information in confidence; (ii) not disclose the Confidential Information to a third party without the Disclosing Party's prior written approval; and (iii) not, directly or indirectly, use the Confidential Information for any purpose other than for exercising its rights and fulfilling its responsibilities pursuant to this Agreement. Each Party shall take reasonable measures to protect the Confidential Information of the other Party, which measures shall not be less than the measures taken by such Party to protect its own confidential and proprietary information.

Obligation of confidentiality shall not apply to information that (i) is or becomes generally known to the public through no act or omission of the Receiving Party; (ii) was in the Receiving Party's lawful possession prior to the disclosure hereunder and was not subject to limitations on disclosure or use; (iii) is developed independently by employees or Contractors of the Receiving Party or other persons working for the Receiving Party who have not had access to the Confidential Information of the Disclosing Party, as proven by the written records of the Receiving Party; (iv) is lawfully disclosed to the Receiving Party without restrictions, by a third party not under an obligation of confidentiality; or (v) the Receiving Party is legally compelled to disclose, in which case the Receiving Party shall notify the Disclosing Party of such compelled disclosure and assert the privileged and confidential nature of the information and cooperate fully with the Disclosing Party to limit the scope of disclosure and the dissemination of disclosed Confidential Information to the minimum extent necessary.

The obligations under this Section 9 shall continue to remain in force for a period of five (5) years after the last disclosure, and, with respect to trade secrets, for so long as such trade secrets are protected under applicable trade secret laws.

10. FEES, DELIVERY AND PAYMENT

10.1. License Fees
License Fees are described in The Qt Company's standard price list, quote or Purchase Order confirmation or in an appendix hereto, as the case may be.

The License Fees shall not be refunded or claimed as a credit in any event or for any reason whatsoever.

10.2. Ordering Licenses
Licensee may purchase Development Licenses and Distribution Licenses pursuant to agreed pricing terms or, if no specific pricing terms have been agreed
upon, at The Qt Company’s standard pricing terms applicable at the time of purchase. Unless specifically otherwise provided, any pricing terms referenced in this Agreement shall be valid for twelve (12) months from the date of this Agreement.

Licensee shall submit all purchase orders for Development Licenses and Distribution Licenses to The Qt Company by email or any other method acceptable to The Qt Company (each such order is referred to herein as a "Purchase Order") for confirmation, whereupon the Purchase Order shall become binding between the Parties.

10.3. Distribution License Packs
Unless otherwise agreed, Distribution Licenses shall be purchased by way of Distribution License Packs.

Upon due payment of the ordered Distribution License Pack(s), the Licensee will have an account of Distribution Licenses available for distributing the Redistributables in accordance with this Agreement.

Each time Licensee distributes a copy of Redistributables, then one Distribution License is used, and Licensee’s account of available Distribution Licenses is decreased accordingly.

Licensee may distribute copies of the Redistributables so long as Licensee has Distribution Licenses remaining on its account.

10.4. Payment Terms
License Fees and any other charges under this Agreement shall be paid by Licensee no later than thirty (30) days from the date of the applicable invoice from The Qt Company.

The Qt Company will submit an invoice to Licensee after the date of this Agreement and/or after The Qt Company receives a Purchase Order from Licensee.

A late payment charge of the lower of (a) one percent per month; or (b) the interest rate stipulated by applicable law, shall be charged on any unpaid balances that remain past due.

10.5. Taxes
All License Fees and other charges payable hereunder are gross amounts but exclusive of any value added tax, use tax, sales tax, withholding tax and other taxes, duties or tariffs ("Taxes") levied directly for the sale, delivery or use of Licensed Software hereunder pursuant to any applicable law. Such applicable Taxes shall be paid by Licensee to The Qt Company, or, where applicable, in lieu of payment of such Taxes to The Qt Company, Licensee shall provide an exemption certificate to The Qt Company and any applicable authority.
11. RECORD-KEEPING AND REPORTING OBLIGATIONS; AUDIT RIGHTS

11.1. Licensee's Record-keeping
Licensee shall at all times during the Term of this Agreement and for a period of seven (7) years thereafter maintain Licensee's Records in an accurate and up-to-date form. Licensee's Records shall be adequate to reasonably enable The Qt Company to determine Licensee's compliance with the provisions of this Agreement. The records shall conform to general good accounting practices.

Licensee shall, within thirty (30) days from receiving The Qt Company's request to that effect, deliver to The Qt Company a report based on Licensee's Records, such report to contain information, in sufficient detail, on (i) number and identity of users working with Licensed Software or Open Source Qt, (ii) copies of Redistributables distributed by Licensee during the most recent calendar quarter and/or any other term specified by The Qt Company, (iii) number of undistributed copies of Redistributables and corresponding number of unused Distribution Licenses remaining on Licensee's account, and (iv) any other information as The Qt Company may reasonably require from time to time.

11.2. The Qt Company's Audit Rights
The Qt Company or an independent auditor acting on behalf of The Qt Company's, may, upon at least five (5) business days' prior written notice and at its expense, audit Licensee with respect to the Licensee's use of the Licensed Software, but not more frequently than once during each 6-month period. Such audit may be conducted by mail, electronic means or through an in-person visit to Licensee's place of business. Any such in-person audit shall be conducted during regular business hours at Licensee's facilities and shall not unreasonably interfere with Licensee's business activities. The Qt Company or the independent auditor acting on behalf of The Qt Company shall be entitled to inspect Licensee's Records and conduct necessary interviews of Licensee's relevant employees and Contractors. All such Licensee's Records and use thereof shall be subject to an obligation of confidentiality under this Agreement.

If an audit reveals that Licensee is using the Licensed Software beyond scope of the licenses Licensee has paid for, Licensee agrees to pay The Qt Company any amounts owed for such unauthorized use within 30 days from receipt of the corresponding invoice from The Qt Company.

In addition, in the event the audit reveals a material violation of the terms of this Agreement (without limitation, either (i) underpayment of more than 10 % of License Fees or 10,000 euros (whichever is more) or (ii) distribution of products, which include or result from Prohibited Combination, shall be deemed a material violation for purposes of this section), then the Licensee shall pay The Qt Company's reasonable cost of conducting such audit.
12. TERM AND TERMINATION

12.1. Agreement Term
This Agreement shall enter into force upon due acceptance by both Parties and remain in force for as long as there is any Development License(s) purchased under this Agreement in force ("Term"), unless and until terminated pursuant to the terms of this Section 12.

12.2. Termination and suspension of rights
Either Party shall have the right to terminate this Agreement upon thirty (30) days prior written notice if the other Party commits a material breach of any obligation of this Agreement and fails to remedy such breach within such notice period.

Instead of termination, The Qt Company shall have the right to suspend or withhold grants of all rights to the Licensed Software hereunder, including but not limited to the Development Licenses, Distribution License, and Support, should Licensee fail to make payment in timely fashion or otherwise violates or is reasonably suspected to violate its obligations or terms of this Agreement, and where such violation or breach is not cured within five (5) business days following The Qt Company's written notice thereof.

12.3. Mutual Right to Terminate
Either Party shall have the right to terminate this Agreement immediately upon written notice in the event that the other Party becomes insolvent, files for any form of bankruptcy, makes any assignment for the benefit of creditors, has a receiver, administrative receiver or officer appointed over the whole or a substantial part of its assets, ceases to conduct business, or an act equivalent to any of the above occurs under the laws of the jurisdiction of the other Party.

12.4. Parties' Rights and Duties upon Termination
Upon expiry or termination of the Agreement, Licensee shall cease and shall cause all Designated Users (including those of its Affiliates' and Contractors') to cease using the Licensed Software and distribution of the Redistributables under this Agreement.

Notwithstanding the above, in the event the Agreement expires or is terminated:
(i) as a result of The Qt Company choosing not to renew the Development License(s) as set forth in Section 3.1, then all valid licenses possessed by the Licensee at such date shall be extended to be valid in perpetuity under the terms of this Agreement and Licensee is entitled to purchase additional licenses as set forth in Section 10.2; or
(ii) for reason other than by The Qt Company pursuant to item (i) above or pursuant to Section 12.2, then the Licensee is entitled, for a period of six (6) months after the effective date of termination,
to continue distribution of Devices under the Distribution Licenses paid but unused at such effective date of termination.

Upon any such termination the Licensee shall destroy or return to The Qt Company all copies of the Licensed Software and all related materials and will certify the same to The Qt Company upon its request, provided however that Licensee may retain and exploit such copies of the Licensed Software as it may reasonably require in providing continued support to Customers.

Expiry or termination of this Agreement for any reason whatsoever shall not relieve Licensee of its obligation to pay any License Fees accrued or payable to The Qt Company prior to the effective date of termination, and Licensee shall immediately pay to The Qt Company all such fees upon the effective date of termination. Termination of this Agreement shall not affect any rights of Customers to continue use of Applications and Devices (and therein incorporated Redistributables).

12.5. Extension in case of bankruptcy
In the event The Qt Company is declared bankrupt under a final, non-cancellable decision by relevant court of law, and this Agreement is not, at the date of expiry of the Development License(s) pursuant to Section 3.1, assigned to party, who has assumed The Qt Company’s position as a legitimate licensor of Licensed Software under this Agreement, then all valid licenses possessed by the Licensee at such date of expiry, and which the Licensee has not notified for expiry, shall be extended to be valid in perpetuity under the terms of this Agreement.

13. GOVERNING LAW AND LEGAL VENUE

In the event this Agreement is in the name of The Qt Company Inc., a Delaware Corporation, then:

(i) this Agreement shall be construed and interpreted in accordance with the laws of the State of California, USA, excluding its choice of law provisions;
(ii) the United Nations Convention on Contracts for the International Sale of Goods will not apply to this Agreement; and
(iii) any dispute, claim or controversy arising out of or relating to this Agreement or the breach, termination, enforcement, interpretation or validity thereof, including the determination of the scope or applicability of this Agreement to arbitrate, shall be determined by arbitration in San Francisco, USA, before one arbitrator. The arbitration shall be administered by JAMS pursuant to JAMS’ Streamlined Arbitration Rules and Procedures. Judgment on the Award may be entered in any court having jurisdiction. This Section shall not preclude parties from seeking provisional remedies in aid of arbitration from a court of appropriate jurisdiction.

In the event this Agreement is in the name of The Qt Company Ltd., a Finnish
Company, then:
(i) this Agreement shall be construed and interpreted in accordance with the laws of Finland, excluding its choice of law provisions;
(ii) the United Nations Convention on Contracts for the International Sale of Goods will not apply to this Agreement; and
(iii) any disputes, controversy or claim arising out of or relating to this Agreement, or the breach, termination or validity thereof shall be finally settled by arbitration in accordance with the Arbitration Rules of Finland Chamber of Commerce. The arbitration tribunal shall consist of one (1), or if either Party so requires, of three (3), arbitrators. The award shall be final and binding and enforceable in any court of competent jurisdiction. The arbitration shall be held in Helsinki, Finland and the process shall be conducted in the English language. This Section shall not preclude parties from seeking provisional remedies in aid of arbitration from a court of appropriate jurisdiction.

14. GENERAL PROVISIONS

14.1. No Assignment
Except in the case of a merger or sale of substantially all of its corporate assets, Licensee shall not be entitled to assign or transfer all or any of its rights, benefits and obligations under this Agreement without the prior written consent of The Qt Company, which shall not be unreasonably withheld or delayed. The Qt Company shall be entitled to freely assign or transfer any of its rights, benefits or obligations under this Agreement.

14.2. No Third-Party Representations
Licensee shall make no representations or warranties concerning the Licensed Software on behalf of The Qt Company. Any representation or warranty Licensee makes or purports to make on The Qt Company's behalf shall be void as to The Qt Company.

14.3. Surviving Sections
Any terms and conditions that by their nature or otherwise reasonably should survive termination of this Agreement shall so be deemed to survive. Such sections include especially the following: 1, 2, 6, 7, 9, 11, 12.4, 13 and 14.

14.4. Entire Agreement
This Agreement, the exhibits hereto, the License Certificate and any applicable Purchase Order accepted by The Qt Company constitute the complete agreement between the Parties and supersede all prior or contemporaneous discussions, representations, and proposals, written or oral, with respect to the subject matters discussed herein.

In the event of any conflict or inconsistency between this Agreement and any Purchase Order, the terms of this Agreement will prevail over the terms of the Purchase Order with respect to such conflict or inconsistency.
Parties specifically acknowledge and agree that this Agreement prevails over any click-to-accept or similar agreements the Designated Users may need to accept online upon download of the Licensed Software, as may be required by The Qt Company's applicable processes relating to Licensed Software.

14.5. Modifications
No modification of this Agreement shall be effective unless contained in a writing executed by an authorized representative of each Party. No term or condition contained in Licensee's Purchase Order ("Deviating Terms") shall apply unless The Qt Company has expressly agreed such Deviating Terms in writing. Unless and to the extent expressly agreed by The Qt Company, any such Deviating Terms shall be deemed void and with no legal effect. For clarity, delivery of the Licensed Software following the receipt of the Purchase Order including Deviating Terms shall not constitute acceptance of such Deviating Terms."

14.6. Force Majeure
Except for the payment obligations hereunder, neither Party shall be liable to the other for any delay or non-performance of its obligations hereunder in the event and to the extent that such delay or non-performance is due to an event of act of God, terrorist attack or other similar unforeseeable catastrophic event that prevents either Party for fulfilling its obligations under this Agreement and which such Party cannot avoid or circumvent ("Force Majeure Event"). If the Force Majeure Event results in a delay or non-performance of a Party for a period of three (3) months or longer, then either Party shall have the right to terminate this Agreement with immediate effect without any liability (except for the obligations of payment arising prior to the event of Force Majeure) towards the other Party.

14.7. Notices
Any notice given by one Party to the other shall be deemed properly given and deemed received if specifically acknowledged by the receiving Party in writing or when successfully delivered to the recipient by hand, fax, or special courier during normal business hours on a business day to the addresses specified for each Party on the signature page. Each communication and document made or delivered by one Party to the other Party pursuant to this Agreement shall be in the English language.

14.8. Export Control
Licensee acknowledges that the Redistributables, as incorporated in Applications or Devices, may be subject to export control restrictions under the applicable laws of respective countries. Licensee shall fully comply with all applicable export license restrictions and requirements as well as with all laws and regulations relating to the Redistributables and exercise of licenses hereunder and shall procure all necessary
governmental authorizations, including without limitation, all necessary licenses, approvals, permissions or consents, where necessary for the re-exportation of the Redistributables, Applications and/or Devices.

14.9. No Implied License
There are no implied licenses or other implied rights granted under this Agreement, and all rights, save for those expressly granted hereunder, shall remain with The Qt Company and its licensors. In addition, no licenses or immunities are granted to the combination of the Licensed Software with any other software or hardware not delivered by The Qt Company under this Agreement.

14.10. Attorney Fees
The prevailing Party in any action to enforce this Agreement shall be entitled to recover its attorney's fees and costs in connection with such action.

14.11. Privacy
Licensee acknowledges and agrees that for the purpose of this Agreement, The Qt Company may collect, use, transfer and disclose personal data pertaining to Designated Users as well as any other employees and directors of the Licensee and its Contractors relevant for carrying out the intent of this Agreement. Such personal data may be collected from the Licensee or directly from the relevant individuals. The Parties acknowledge that with regard to such personal data processed hereunder, The Qt Company shall be regarded as the Data Controller under the applicable Data Protection Legislation. The Qt Company shall process any such personal data in accordance with its privacy policies and practices, which will comply with all applicable requirements of the Data Protection Legislation.

14.12. Severability
If any provision of this Agreement shall be adjudged by any court of competent jurisdiction to be unenforceable or invalid, that provision shall be limited or eliminated to the minimum extent necessary so that this Agreement shall otherwise remain in full force and effect and enforceable.

APPENDICES

The Agreement includes Appendix 1, and possibly one or more of the appendices 3-5, depending on the product(s) purchased by the Licensee, what is stated in the quote or invoice, and/or what is stated in the Licensee's License Certificate.

APPENDIX 1

The modules and/or tools that are included in the respective product - Qt for Application Development (QtAD), Qt for Device Creation (QtDC), Qt for MCUs (QtMCU), Qt 3D Studio (Qt3DS) and Qt Design Studio (QtDS) - are
marked with 'X' in the below table.

Parts of the product that are permitted for distribution in object-code form only ("Redistributables") are marked with 'R' in the below table.

<table>
<thead>
<tr>
<th>Modules / Tools</th>
<th>QtAD</th>
<th>QtDC</th>
<th>QtMCU</th>
<th>Qt3DS</th>
<th>QtDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qt Core</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt GUI</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Multimedia</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Multimedia Widgets</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Network</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt QML</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Controls 2</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Dialogs</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Layouts</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Test</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt SQL</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Test</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Widgets</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active Qt</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt 3D</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Android Extras</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Bluetooth</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Canvas 3D</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Concurrent</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Source Used In DNAC Platform Antelope 12179</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt D-Bus</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Gamepad</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Graphical Effects</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Help</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Image Formats</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Location</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Mac Extras</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Network Authorization</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt NFC</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Platform Headers</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Positioning</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Print Support</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Purchasing</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt for Python</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Quick Controls</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Quick Extras</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Quick Widgets</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt SCXML</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Sensors</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Serial Bus</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Serial Port</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt Speech</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt SVG</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt UI Tools</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
<td>X, R</td>
</tr>
<tr>
<td>Qt WebChannel</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt WebEngine</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt WebSockets</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt WebView</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Windows Extras</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt X11 Extras</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt XML</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt XML Patterns</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Wayland Compositor</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Charts</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Data visualization</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Virtual Keyboard</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boot 2 Qt stack</td>
<td></td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt OTA</td>
<td></td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device Utilities</td>
<td></td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Debugging Bridge (QDB) Daemon</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Ultralite Controls</td>
<td></td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Ultralite</td>
<td></td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Creator</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Designer (Qt Widget Designer)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Designer (Qt Creator plugin)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Linguist</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Assistant</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>---------------+-----+-----+---+---+---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lupdate</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Lrelease</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Qmake</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uic</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rcc</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qlalr</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qdoc</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qmlscene</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qmlviewer</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target toolchains</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Debugging Bridge (QDB) Host</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qconfig-gui</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Emulator</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qmlinterfacegenerator</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qmltocpp</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qulfontcompiler</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt53DStudioRuntime2</td>
<td>X, R</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt 3D Studio</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Design Studio</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPENDIX 3: ADDITIONS TO LICENSED SOFTWARE

In addition to what is provided under the definition of the Licensed Software, Parties agree that Licensed Software shall also include the Add-On Products of The Qt Company, as mentioned in this Appendix, if included in the quote / invoice.

The Modules and/or Tools of the Licensed Software that are included with each
Add-On Product respectively are marked with 'X' in the below table.

Parts of the respective Add-On Product that are permitted for distribution in object-code form only (“Redistributables”) are marked with ‘R’ in the below table.

<table>
<thead>
<tr>
<th>Modules / Tools of Licensed Software</th>
<th>Add-On Product(s)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Qt MQTT</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt KNX</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt OPC UA</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt CoAP</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Safe Renderer</td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Application Manager</td>
<td></td>
<td>X, R</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt IVI</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference UI</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt GENIVI Extras</td>
<td>X, R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QML Live</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Creator</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deployment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Creator Plugin</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>for Qt Application Manager</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Automotive Suite</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deployment Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Design Studio</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt 3D Studio</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GammaRay</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Platform adaptations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>for specified</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deployment Platforms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt for Device</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All the above Redistributables are subject to applicable provisions and limitations including but not limited to what is defined in section 3 of the Agreement.

**APPENDIX 4: SMALL BUSINESS AND STARTUP**

The provisions of this Appendix 4 are applicable for companies with an annual revenue, including funding, equivalent to maximum of 250,000 USD (in applicable currency) during the latest full calendar year, as evidenced by duly audited records of the Licensee and approved by The Qt Company ("Start-up Company").

Start-up Companies are qualified for a discounted License Fee for maximum of four (4) Development Licenses ("Start-up Development License") unless otherwise agreed between the parties.

Start-up Development License entitles the respective Designated User for Support only for installation related issues.

Upon expiry of the respective License Term, the Start-up Development Licenses shall be automatically extended, pursuant to Section 3.1 of the Agreement, for a Renewal Term either as new Start-up Development Licenses (if the Licensee still qualifies as a Start-up Company), or as normal Development Licenses (if the Licensee no longer qualifies as a Start-up Company).

**APPENDIX 5: NON-COMMERCIAL USE**

The provisions of this Appendix 5 are applicable for non-commercial use of the Licensed Software by the Licensee.

For the purpose of this Appendix 5, the following additional definitions (replacing the relevant definition of the Agreement, where applicable) shall be applicable:

"Demo Units" shall mean (i) hardware development platform, which incorporates the Licensed Software along with Licensee's software and/or hardware, and (ii) prototype versions of Applications or Devices.
"Designated User(s)" shall mean the employees and students of the Licensee.

"Licensee Products" shall mean Applications and/or Devices.

"Permitted Purpose" shall mean (i) Licensee's internal evaluation and testing of Licensed Software, (ii) building Demo Units as well as (iii) educational use.

"Term" shall mean a period of twelve (12) months or any such other period as may be agreed between the Parties.

For the purpose of this Appendix 5, the following changes shall be agreed with respect to relevant Sections of the Agreement:

I. Recital (A) shall be replaced in its entirety to read as follows:
   "(A) Licensee wishes to use the Licensed Software for the Permitted Purpose."

II. Section 3.1 shall be replaced in its entirety to read as follows:
   "The Qt Company grants to Licensee a personal, non-exclusive, non-transferable, revocable, royalty-free license, valid for the Term, to use, modify and copy the Licensed Software solely for the Permitted Purpose.
   Licensee may install copies of the Licensed Software on five (5) computers per Designated User, provided that only the Designated Users who have a valid Development License may use the Licensed Software.
   Licensee may demonstrate the Demo Units, provided that such demonstrations must be conducted by Licensee, and the Demo Units must remain in Licensee's possession and under Licensee's control at all times.
   For clarity, this Agreement does not (i) entitle Licensee to use Licensed Software to create Applications or Devices (other than prototypes thereof) or (ii) carry any distribution rights to Licensee, but such rights are subject to and conditional upon conclusion of a separate license agreement with The Qt Company."

III. Sections 3.2, 3.3, 8 and 10 shall be deleted.

IV. Section 3.4 shall be replaced in its entirety to read as follows:
   "Licensee shall not:
   - remove or alter any copyright, trademark or other proprietary rights notice contained in any portion of the Licensed Software;
   - transfer, publish, sublicense, disclose, display or otherwise make the Licensed Software available to any third party (except that Licensee may demonstrate the Demo Units pursuant to Section 3.1);
   - in any way combine, incorporate or integrate Licensed Software with, or use Licensed Software for creation of, any software created with or incorporating Open Source Qt;"
Licensee shall cause all Designated Users who make use of the licenses granted under this Agreement, to be contractually bound to comply with the relevant terms of this Agreement and not to use the Licensed Software beyond the terms hereof. Licensee shall be responsible for any and all actions and omissions of its Designated Users relating to the Licensed Software and use thereof.

Any use of Licensed Software beyond the provisions of this Agreement is strictly prohibited and requires an additional license from The Qt Company.

V. Section 12 shall be replaced in its entirety to read as follows:
"This Agreement shall enter into force upon due acceptance by both Parties and remain in force for the Term, unless and until terminated pursuant to the terms of Section 12.

Upon termination of the Agreement, Licensee shall cease using the Licensed Software. All other copies of Licensed Software in the possession or control of Licensee must be erased or destroyed. An officer of Licensee must, upon request, promptly deliver to The Qt Company a written confirmation that this has occurred."

Except for the modifications specified above, this Appendix carries no change to the terms of the Agreement which shall remain in full force.

Copyright (c) 2008-2012, SARL Adaltas.
All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of SARL Adaltas nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of SARL Adaltas.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2014-2015 Calvin Metcalf & contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT

Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHERWISE ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
# This file contains a list of people who've made non-trivial
# contribution to the Google C++ Testing Framework project. People
# who commit code to the project are encouraged to add their names
# here. Please keep the list sorted by first names.

Ajay Joshi <jaj@google.com>
Balzs Dn <balazs.dan@gmail.com>
Bharat Mediratta <bharat@menalto.com>
Chandler Carruth <chandlerc@google.com>
Chris Prince <cprince@google.com>
Chris Taylor <taylorc@google.com>
Dan Egnor <egnor@google.com>
Eric Roman <eroman@chromium.org>
Hady Zalek <hady.zalek@gmail.com>
Jeffrey Yasskin <jyasskin@gmail.com>
Ji Sigursson <joi@gmail.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kenton Varda <kenton@google.com>
Manuel Klimek <klimek@google.com>
Markus Heule <markus.heule@gmail.com>
Mika Raento <mikie@iki.fi>
Mikls Fazekas <mifazekas@szemafor.com>
Pasi Valminen <pasi.valminen@gmail.com>
Patrick Hanna <phanna@gmail.com>
Patrick Riley <pfr@google.com>
Peter Kaminski <piotrk@gmail.com>
Preston Jackson <preston.a.jackson@gmail.com>
Rainer Klaffenboeck <rainer.klaffenboeck@dynatrace.com>
Russ Cox <rsco@google.com>
Russ Rufer <rufr@pentad.com>
Sean Mcafee <seefacm@gmail.com>
Sigurur sgeirsson <siggi@google.com>
Tracy Bialik <tracy@pentad.com>
Vadim Berman <vadimb@google.com>
Vlad Losev <vlad@google.com>
Zhanyong Wan <wan@google.com>
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013-2018 sha.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 1998 - 2009, Paul Johnston & Contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for

Open Source Used In DNAC Platform Antelope 12189
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is
"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.
(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.
You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.
11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
You may use, distribute and copy the Qt Toolkit under the terms of GNU Lesser General Public License version 3, which is displayed below. This license makes reference to the version 3 of the GNU General Public License, which you can find in the LICENSE.GPLv3 file.

-------------------------------------------------------------------------

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.
1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based
on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Copyright 2018, VideoLAN and dav1d authors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (C) 2016-2019 Erik Barke, Monounity.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Suite 500, Boston, MA 02110-1335 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

^L

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those
When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least
three years, to give the same user the materials specified in
Subsection 6a, above, for a charge no more than the cost of
performing this distribution.
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above,

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License
may add an explicit geographical distribution limitation excluding those
countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILUIRE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Suite 500, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.
That's all there is to it!

Copyright (c) 2011, The WebRTC project authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) Feross Aboukhadieh, and other contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license
(Fonts) and associated documentation files (the Font Software), to reproduce and distribute the Font Software,
including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font
Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following
conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more
of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the
Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed
to names not containing either the words Bitstream or the word Vera.

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is
distributed under the Bitstream Vera names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software
typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK,
OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR
ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT,
INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR
FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of GNOME, the GNOME Foundation, and Bitstream Inc., shall not be
used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written
authorization from the GNOME Foundation or Bitstream Inc., respectively. For further information, contact: fonts at
gnome dot org.

**************************************************************************
**
** Copyright (C) YYYY The Qt Company Ltd.
** Contact: https://www.qt.io/licensing/
**
** This file is part of the FOO module of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:LGPL3-COMMS

Open Source Used In DNAC Platform Antelope 12216
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms
** and conditions see http://www.qt.io/terms-conditions. For further
** information use the contact form at http://www.qt.io/contact-us.
**
** GNU Lesser General Public License Usage
** Alternatively, this file may be used under the terms of the GNU Lesser
** General Public License version 3 as published by the Free Software
** Foundation and appearing in the file LICENSE.LGPLv3 included in the
** packaging of this file. Please review the following information to
** ensure the GNU Lesser General Public License version 3 requirements
** will be met: https://www.gnu.org/licenses/lgpl.html.
**
** $QT_END_LICENSE$
**
****************************************************************************/

The MIT License (MIT)

Copyright (c) 2015 Elijah Insua

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2012, Linux USB Project
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

- Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-2017 Ole Michelsen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) Vsevolod Strukchinsky <floatdrop@gmail.com> (github.com/floatdrop)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2009-2014 Contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The FreeType Project LICENSE
----------------------------

2006-Jan-27

Copyright 1996-2002, 2006 by
David Turner, Robert Wilhelm, and Werner Lemberg

Introduction
============

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.
This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- We don’t promise that this software works. However, we will be interested in any kind of bug reports. (`as is' distribution)

- You can use this software for whatever you want, in parts or full form, without having to pay us. (`royalty-free' usage)

- You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (`credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```
Portions of this software are copyright <year> The FreeType Project (www.freetype.org). All rights reserved.
```

Please replace <year> with the value from the FreeType version you actually use.

Legal Terms
============

0. Definitions
-------------

Throughout this license, the terms 'package', 'FreeType Project', and 'FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the 'FreeType Project', be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where
'using' is a generic term including compiling the project's source code as well as linking it to form a 'program' or 'executable'. This program is referred to as 'a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty

---------------

THE FREETYPE PROJECT IS PROVIDED 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

---------------

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- Redistribution of source code must retain this license file ('FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.

- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your
documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: 'FreeType Project', 'FreeType Engine', 'FreeType library', or 'FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts

There are two mailing lists related to FreeType:

 o freetype@nongnu.org

Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

 o freetype-devel@nongnu.org

Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

https://www.freetype.org
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) Feross Aboukhadijeh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2015-2019, Brandon Jones, Colin MacKenzie IV.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright JS Foundation and other contributors, https://js.foundation
Copyright (C) 2012-2013 Yusuke Suzuki (twitter: @Constellation) and other contributors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Change Log

### v25.0.1 (2019/01/10 19:29 +00:00)
- [f1c835b](https://github.com/davglass/license-checker/commit/f1c835b991b65dc928eafe64c29b73df9e703169) Updated readme with new args (@davglass)
- [ada667f](https://github.com/davglass/license-checker/commit/ada667f28e8ef4e96b5b6059fe05541d05ca936c) changelog (@davglass)
- [7813f0d](https://github.com/davglass/license-checker/commit/7813f0db3b51f7cc4cd37d8e7b4670f9005fed06) contrib (@davglass)

### v25.0.0 (2019/01/10 15:09 +00:00)
- [cdb77d9](https://github.com/davglass/license-checker/commit/cdb77d9a2b36cd87503f32265a0ee388b25c24ae) 25.0.0 (@davglass)
- [b897345](https://github.com/davglass/license-checker/commit/b89734549bc05fb5e143083df6fd3959d87eb2b8) [Fixes #136](https://github.com/davglass/license-checker/pull/136) Document --start option and add the depth option for looking into root dependencies or infinitely (@davglass)
- [#185](https://github.com/davglass/license-checker/pull/185) Merge pull request #185 from theboolean/fixed/fix-exclude-packages-and-private (@theboolean)
- [#184](https://github.com/davglass/license-checker/pull/184) Merge pull request #184 from theboolean/docs/fix-typo (@theboolean)
- [987a60f](https://github.com/davglass/license-checker/commit/987a60fbb58052cf36d5a80e6252e81136bb1dd8) changes 24.1.0 (@davglass)

### v24.1.0 (2018/12/14 20:54 +00:00)
- [18ab8a1](https://github.com/davglass/license-checker/commit/18ab8a1eb478d1354429ca4318c1df3de3e4cb1) 24.1.0 (@davglass)
- [1dca0f6](https://github.com/davglass/license-checker/commit/1dca0f6161aebeddef2d398c130a2a4f86c2b4e88c8) Changelog (@davglass)
- [#188](https://github.com/davglass/license-checker/pull/188) Fix for #153: remove deprecated package 'spdx' (#188) (@rogierschouten)
- [d1d34b1](https://github.com/davglass/license-checker/commit/d1d34b1a8894987807419c093c3ee10b4f06db3d) Fix typo in package.json description (@theboolean)
- [f5c9008](https://github.com/davglass/license-checker/commit/f5c9008a6d20b503e21c0d8130e52be56a5bf8e8) Fix: Allow excluding packages and private packages together (@theboolean)
- [98c4dc6](https://github.com/davglass/license-checker/commit/98c4dc670fdebb6e58b62b38ebbc12302e5b1db5) changes 24.0.0 (@davglass)

### v24.0.1 (2018/10/11 16:40 +00:00)
- [d72eae8](https://github.com/davglass/license-checker/commit/d72eae8bcb80d8582e6d12c5e2664e9c19135709) 24.0.1 (@davglass)
- [9433e64](https://github.com/davglass/license-checker/commit/9433e640ea9f629aad5f00c224a9f61026b8116d) [Fixes #181](https://github.com/davglass/license-checker/pull/181) - Removed extra line break with --cvs (@davglass)
- [65baa28](https://github.com/davglass/license-checker/commit/65baa28f532f87eb75a32578c7ac5837806d12f) contrib and changes (@davglass)

### v24.0.0 (2018/10/09 21:12 +00:00)
- [84d64a2](https://github.com/davglass/license-checker/commit/84d64a28fa971c3bfb20b9e9de27ca2be6e8481b) 24.0.0 (@davglass)
- [#180](https://github.com/davglass/license-checker/pull/180) Feature: Exclude Private Packages (#180) (Jonny Reeves)
- [#179](https://github.com/davglass/license-checker/pull/179) Merge pull request #179 from jonny-improbable/fixed/fix-strict-checks (@jonny-improbable)
- [179077d](https://github.com/davglass/license-checker/commit/179077d079acbc0d183aae6691cfe5c2ec0f9307) Fixes 176, Packages are not ignored when using --onlyAllow and --excludePackages. (@jonny-improbable)
- [890ca17](https://github.com/davglass/license-checker/commit/890ca17d179bc43a6cb7cf96269d1521d9ac2)
### v23.0.0 (2018/10/09 14:37 +00:00)
- [f243287](https://github.com/davglass/license-checker/commit/f243287fbc6da33838a226ddf8e412535390d4) 23.0.0 (@davglass)
- [#178](https://github.com/davglass/license-checker/pull/178) Merge pull request #178 from jonny-improbable/fixed/fixing (@jonny-improbable)
- [#175](https://github.com/davglass/license-checker/pull/175) Merge pull request #175 from jonny-improbable/patch-1 (@jonny-improbable)
- [f2c6611](https://github.com/davglass/license-checker/commit/f2c6611d6c68e50b2b6f56260026837e7167f44a) Fixes #177, --packages and --excludePackages filtering (@jonny-improbable)
- [0a598df](https://github.com/davglass/license-checker/commit/0a598df55c3ec33803ef33be07cc47a9100ee961) Improve --onlyAllow violation Error
- [ee723af](https://github.com/davglass/license-checker/commit/ee723af79889a4d98ed067494a58f6aa41a44ac7) changes (@davglass)

### v22.0.0 (2018/10/08 19:00 +00:00)
- [bd398c3](https://github.com/davglass/license-checker/commit/bd398c3cbe0279699e9aad1dc4b592f398a75766) 22.0.0 (@davglass)
- [#172](https://github.com/davglass/license-checker/pull/172) Merge pull request #172 from fzaninotto/exclude-packages (@fzaninotto)
- [014fd96](https://github.com/davglass/license-checker/commit/014fd9613fe5460e0b7a0fe118a659a337e25e2c) added changes (@davglass)

### v21.0.0 (2018/10/06 13:24 +00:00)
- [d75fe43](https://github.com/davglass/license-checker/commit/d75fe43d61a9d5b6696d71a663e25a53fd3d44f7) 21.0.0 (@davglass)
- [05cb221](https://github.com/davglass/license-checker/commit/05cb221359f4f1e039f586cd9e8fb7cb9d6398) [Fixes #174] - Moved private and unknown detection above filtering (@davglass)
- [bd5700](https://github.com/davglass/license-checker/commit/bd570088236db90253bdac27763523b2e74f8b) Add an excludePackages option (@fzaninotto)
- [231bd3f](https://github.com/davglass/license-checker/commit/231bd3f2412507be1d702ad100b23c1b1fe93602) updated readme (@davglass)
- [6c61a7b](https://github.com/davglass/license-checker/commit/6c61a7bd024666702065d42706392c11717cc1d) changelog (@davglass)

### v20.2.0 (2018/07/30 13:49 +00:00)
- [28dc521](https://github.com/davglass/license-checker/commit/28dc5218dec3340778bd801593275ad1b222e53f) 20.2.0 (@davglass)
- [#169](https://github.com/davglass/license-checker/pull/169) Merge pull request #169 from codepunkt/restrictPackages (@codepunkt)
- [5e4b274](https://github.com/davglass/license-checker/commit/5e4b274e955485f2aba37e65fec913b8042f38d) refactor: restore previous code style (@codepunkt)
- [0f53922](https://github.com/davglass/license-checker/commit/0f5392208be00b0df48db7ea8848b53d0137e45) feat: allow restricting output to specified packages (@codepunkt)
- [5f63f87](https://github.com/davglass/license-checker/commit/5f63f8766a17e589b57db99bda4eb54d91b85b77)
changes and contrib (@davglass)

### v20.1.0 (2018/06/07 14:33 +00:00)
- [970882a](https://github.com/davglass/license-checker/commit/970882a99b345c7f32496431e5ca4c069ad7e320) 20.1.0 (@davglass)
- [#165](https://github.com/davglass/license-checker/pull/165) Merge pull request #165 from santiagocanti/fix-exclude-not-adding-custom-licenses (@santiagocanti)
- [89e48a9](https://github.com/davglass/license-checker/commit/89e48a9d3d0c9f2e48bfcee0dc3563ce4b652e22) Fix exclude not including custom licenses (@santiagocanti)

### v20.0.0 (2018/05/17 14:14 +00:00)
- [7e89310](https://github.com/davglass/license-checker/commit/0e8931021df777cb401d9a332fa2a8c9a2c49eb2) 20.0.0 (@davglass)
- [c36a5c9](https://github.com/davglass/license-checker/commit/c36a5c99bb4f68816e4e19cad5090ba3c7ba67a4) changes (@davglass)
- [ccb66f3](https://github.com/davglass/license-checker/commit/ccb66f3410a4589e612b7cf1c3ae7b1446bab460) contributors (@davglass)
- [#156](https://github.com/davglass/license-checker/pull/156) Merge pull request #156 from zodiac403/z4/copyright (@zodiac403)
- [56414b4](https://github.com/davglass/license-checker/commit/56414b4a67cb25bcc8eb8c050b0ac712b677dc2) Add unit test for Copyright statement. (@zodiac403)
- [3eafe24](https://github.com/davglass/license-checker/commit/3eafe249ad662af727a9b7472f7ad6c1b9a73ade) fixing test coverage with switch to nyc from istanbul (@davglass)

### v19.0.0 (2018/04/30 14:51 +00:00)
- [7e83a06](https://github.com/davglass/license-checker/commit/7e83a069d23b1043cd8257eb6466704ecca465) 19.0.0 (@davglass)
- [e5eb49b](https://github.com/davglass/license-checker/commit/e5eb49bcb5546dbc681736f139d63908cd39ce9) upgrading packages (@davglass)
- [442ae6](https://github.com/davglass/license-checker/commit/442ae69b5d44e47bf2e441c4e1f86d387e49c6) Changlog (@davglass)

### v18.1.0 (2018/04/30 14:16 +00:00)
- [82ec91b](https://github.com/davglass/license-checker/commit/82ec91bb02e6bf998164b62b4aa827198156ad4c) 18.1.0 (@davglass)
- [4b546d3](https://github.com/davglass/license-checker/commit/4b546d38ce91bcbe8d21e7bc53e44635b0447fc2) contrib (@davglass)
- [#155](https://github.com/davglass/license-checker/pull/155) Merge pull request #155 from ktmouk/supress-path-property (@ktmouk)
- [#160](https://github.com/davglass/license-checker/pull/160) Merge pull request #160 from helio-frota/pr (@helio-frota)
- [8b068f2](https://github.com/davglass/license-checker/commit/8b068f25fe910d087e06e29312bc250b49aba4c7) fix: Updating debug dependency to fix a security warning (@helio-frota)
- [040dd94](https://github.com/davglass/license-checker/commit/040dd9468261966c5c40e4a18aa9ba8f716c3a2) Copyright statement: Handle multi-line statements (@zodiac403)
- [0480750](https://github.com/davglass/license-checker/commit/04807506d61a1d704e8b51d3a289baa025b510) Fix layout issue. (@zodiac403)
- [9cb45f2](https://github.com/davglass/license-checker/commit/9cb45f2b9ce2c1142c0d1f556f01f7d754737a4c)
- [f83aaae](https://github.com/davglass/license-checker/commit/f83aaae85ad4c2ce9880802951dfacbc86dc2d7f) Merge branch 'z4/copyright' of https://github.com/zodiac403/license-checker into z4/copyright (@zodiac403)
- [7536af9](https://github.com/davglass/license-checker/commit/7536af935d91078657f0e169ed8238f099607f16) Trim output. (@zodiac403)
- [8716845](https://github.com/davglass/license-checker/commit/87168456cae35263882db2b6d60081f228b6b66) Suppress path property (@ktmouk)
- [be57ee8](https://github.com/davglass/license-checker/commit/be57ee8f76c8aada0d227913b7d0e12a0dc640) Add example for (c) statement. (@zodiac403)
- [fe17f7c](https://github.com/davglass/license-checker/commit/fe17f7c76dbdaee0cbf689a0cd260a746698f3281) Parse (c) statements from LICENSE file. (@zodiac403)
- [#149](https://github.com/davglass/license-checker/pull/149) Merge pull request #149 from honzajavorek/patch-1 (@honzajavorek)
- [955bbf6](https://github.com/davglass/license-checker/commit/955bbf6070ffcd9f0d01ef37bb84fbbac65e007b) remove the old status badge (@honzajavorek)

### v18.0.0 (2018/03/20 12:30 +00:00)
- [315de0b](https://github.com/davglass/license-checker/commit/315de0b1a881cf8737c3c0131c217c4820df13a8) 18.0.0 (@davglass)
- [76655d5](https://github.com/davglass/license-checker/commit/76655d59bad95c2e52724aaa3a0cf036f0e60bd6) changelog (@davglass)
- [c8aa823](https://github.com/davglass/license-checker/commit/c8aa823342b7a250d510582c8a03dd76af50bd9c) added contributors (@davglass)
- [#147](https://github.com/davglass/license-checker/pull/147) Merge pull request #147 from honzajavorek/honzajavorek/license-/* (@honzajavorek)
- [#148](https://github.com/davglass/license-checker/pull/148) Merge pull request #148 from honzajavorek/honzajavorek/use-stderr (@honzajavorek)
- [#146](https://github.com/davglass/license-checker/pull/146) Merge pull request #146 from honzajavorek/honzajavorek/fixed-e2e-test (@honzajavorek)
- [#145](https://github.com/davglass/license-checker/pull/145) Merge pull request #145 from honzajavorek/honzajavorek/drop-4-add-8 (@honzajavorek)
- [0105f7d](https://github.com/davglass/license-checker/commit/0105f7dcef01a67b8f7c34371d95a1d76b191d5ed) use stderr where applicable (@honzajavorek)
- [0ff8d91](https://github.com/davglass/license-checker/commit/0ff8d917c659446256074c0940ae43dd8493233) add support for LICENSE-MIT and similar (@honzajavorek)
- [f02589e](https://github.com/davglass/license-checker/commit/f02589e62c0ba8221081b8b1799d561c3ff20500) isolate and test the license file detection (@honzajavorek)
- [f0da0fe](https://github.com/davglass/license-checker/commit/f0da0fe2ab28f1658e4edfd4bb034f1b536b9ff8) add warning about commas and test it (@honzajavorek)
- [3db0b6f](https://github.com/davglass/license-checker/commit/3db0b6f05a68ccff4f4181503ca8edce4ef10cd7) fix e2e test (transition from comma delimiter to semicolon) (@honzajavorek)
- [cecb373](https://github.com/davglass/license-checker/commit/cecb373505ac4a2dca211e78427624d009585dc) let Travis CI to cache node_modules (@honzajavorek)
- [8cea3dc](https://github.com/davglass/license-checker/commit/8acea3dccf1d185d3c3ae3f37f44a82d7e123a1988) add Travis CI badge to the README (@honzajavorek)
- [8f59c3b](https://github.com/davglass/license-checker/commit/8f59c3b494f69b1d51535bcff80c0e006d0ef66b) drop support for node 4, add support for node 8 (@honzajavorek)
- [5bb9a17](https://github.com/davglass/license-checker/commit/5bb9a17f89d7e180b3a9d929ebc073135262b2d2f) changes (@davglass)
- [b1e0392](https://github.com/davglass/license-checker/commit/b1e0392afa4b511b7c532125cd3165a828d3d4e3) added contrib (@davglass)
- [#121](https://github.com/davglass/license-checker/pull/121) Add `--summary` mode to show a summary with counts of the used licenses (#121) (@tobilg)
- [#119](https://github.com/davglass/license-checker/pull/119) Fix minor spelling mistakes (#119) (@tobiasbueschel)
- [#118](https://github.com/davglass/license-checker/pull/118) Minor readme text improvements (#118) (@peteruithoven)
- [d2bc5ea](https://github.com/davglass/license-checker/commit/d2bc5ea91ab50becea91d5f0f050955fe9a6ac59) added license parsing info to README (@davglass)
- [3e9d320](https://github.com/davglass/license-checker/commit/3e9d320acb2e9f14297d6a4355680f34a9b0609d) changes (@davglass)

### v13.0.3 (2017/08/04 17:15 +00:00)
- [2caf76b](https://github.com/davglass/license-checker/commit/2caf76be7f2849e153ee28410f3b7a41df3283b5) 13.0.3 (@davglass)
- [fadddc8](https://github.com/davglass/license-checker/commit/fadddc895ef9c6f182e2283c5605a8c1d8fa8a25) [fixes #117] fixed minor URL parsing (@davglass)
- [e58a868](https://github.com/davglass/license-checker/commit/e58a868268aefd97d735b364bd87d03140d76482) changes (@davglass)

### v13.0.2 (2017/07/18 16:24 +00:00)
- [2c712b7](https://github.com/davglass/license-checker/commit/2c712b7c7e6efe916d60f3ef3125f7f019c46d59) 13.0.2 (@davglass)
- [ef812c1](https://github.com/davglass/license-checker/commit/ef812c1168901d08b97dbd450a64ca03bb7062ad) [fixes #114] - Custom URL/File parsing skipped if file found on disk (@davglass)

### v13.0.1 (2017/07/17 12:48 +00:00)
- [a1bf8f1](https://github.com/davglass/license-checker/commit/a1bf8f152b100d81b44757b3b7e141df6a971c44) 13.0.1 (@davglass)
- [e6d0c89](https://github.com/davglass/license-checker/commit/e6d0c893b8a86e12493f7a5c22fd74ef26267994) contrib & changes (@davglass)
- [#113](https://github.com/davglass/license-checker/pull/113) Use LICENSE file in preference to README (#113) (@ibeucaly)
- [6b7f79f](https://github.com/davglass/license-checker/commit/6b7f79f654e21903ea37e4391fb3601ece32e90) changes (@davglass)

### v13.0.0 (2017/07/14 12:49 +00:00)
- [dd9e880](https://github.com/davglass/license-checker/commit/dd9e880abdb234d683d8c1e57e7a5f889c9156de4) 13.0.0 (@davglass)
- [051b20b](https://github.com/davglass/license-checker/commit/051b20b0c4b35f223f724705a0025dacf809373f) cleaned up some tests (@davglass)
- [#110](https://github.com/davglass/license-checker/pull/110) FEATURE: Recognise some custom licenses by file or url (#110) (@dancrumb)
- [#111](https://github.com/davglass/license-checker/pull/111) Resolves #60 (#111) Adding support for GPL and LGPL (@dancrumb)
- [#109](https://github.com/davglass/license-checker/pull/109) FEATURE: Improve license exclusion (#109) (@dancrumb)

- [#108](https://github.com/davglass/license-checker/pull/108) FEATURE: Detect private modules and report them as UNLICENSED (#108) (@dancrumb)

### v12.1.0 (2017/07/12 16:48 +00:00)
- [dc944e5](https://github.com/davglass/license-checker/commit/dc944e54d55d77b0ef6b6321d9abe847662cccb8) 12.1.0 (@davglass)

- [d415c58](https://github.com/davglass/license-checker/commit/d415c580b04bc5eea9f2dc9f9a0a31b244c1bd3) updated contrib and changes (@davglass)

- [#107](https://github.com/davglass/license-checker/pull/107) Merge pull request #107 from dancrumb/feature/support-for-public-domain-licenses (@dancrumb)

- [0c55770](https://github.com/davglass/license-checker/commit/0c55770294167161c48b79cc8a15a6545154a87) FEATURE: Add support for Public Domain licenses (@dancrumb)

### v12.0.0 (2017/07/05 13:33 +00:00)
- [8b74fa1](https://github.com/davglass/license-checker/commit/8b74fa1fe3b5edb0d217aa3ce86f4662eb024621) 12.0.0 (@davglass)

- [4f3367c](https://github.com/davglass/license-checker/commit/4f3367c2bf2e9713a6c0c11341f7f4a70ed400) changes & contrib (@davglass)

- [#106](https://github.com/davglass/license-checker/pull/106) Merge pull request #106 from helio-frota/licensefile-readme (@helio-frota)

- [a229507](https://github.com/davglass/license-checker/commit/a2295073d0178dca1b88e85ea8f52c21d35e52fc) src: the README is added as licenseFile (@helio-frota)

- [b59d90d](https://github.com/davglass/license-checker/commit/b59d90d8705f6f728c817b8993695097d26857b) changes & contrib (@davglass)

### v11.0.0 (2017/05/30 13:33 +00:00)
- [258df10](https://github.com/davglass/license-checker/commit/258df102da484943088da8e0b6160b80c0e703cd) 11.0.0 (@davglass)

- [#102](https://github.com/davglass/license-checker/pull/102) Merge pull request #102 from xswordsx/multiple-licenses (@xswordsx)

- [c5c8b0f](https://github.com/davglass/license-checker/commit/c5c8b0f408f2a26259a7b1109f7cc135ed5b9e4) Check for SPDX-compatible licenses (@xswordsx)

### v10.1.0 (2017/05/25 16:16 +00:00)
- [b52a16d](https://github.com/davglass/license-checker/commit/b52a16d0ba414e7287f82325dd65619cd76afdd1) 10.1.0 (@davglass)

- [f911a99](https://github.com/davglass/license-checker/commit/f911a9919033254945ecbf69362cb17143668e) contrib & changes (@davglass)

- [#100](https://github.com/davglass/license-checker/pull/100) Suppress JSON properties with custom format (#100) (@tbbstny)

- [ed5b3ab](https://github.com/davglass/license-checker/commit/ed5b3ab508250be032958f8233dd3000ee98e85) changelog (@davglass)

### v10.0.0 (2017/05/17 13:51 +00:00)
- [b2c2283](https://github.com/davglass/license-checker/commit/b2c2283cf5a937e64a07ab200ec2379e481903b6) 10.0.0 (@davglass)
- [a0406ac](https://github.com/davglass/license-checker/commit/a0406ac2c7a0b1037ca31b47e36d316b9c088938) [fixes #96] - do not replace newline, etc in licenses for exporting to json/markdown (@davglass)
- [98a03f4a](https://github.com/davglass/license-checker/commit/98a03f4a700ef6d038cc2f9104ea5cf17c82a10ad) [fixes #97] - Added a newline to the output file (@davglass)
- [998](https://github.com/davglass/license-checker/pull/98) Merge pull request #98 from davglass/eslint (@davglass)
- [10fe441](https://github.com/davglass/license-checker/commit/10fe441a2e70e549e57f7da99d2287fc553555500) moved to eslint (@davglass)
- [f82b535](https://github.com/davglass/license-checker/commit/f82b535d3b7f152fa00bb4680e94af18d1c1421d) changelog (@davglass)

### v9.0.3 (2017/04/10 13:42 +00:00)
- [a0fd60f](https://github.com/davglass/license-checker/commit/a0fd60f972b7c84a32117b3d6c1f95bfa2282f27) 9.0.3 (@davglass)
- [#95](https://github.com/davglass/license-checker/pull/95) Merge pull request #95 from abtris/prefix-component (@abtris)
- [3b80272](https://github.com/davglass/license-checker/commit/3b80272a217687c2f0cd7a0c1e98775064d820ff) test: add test for csvComponentPrefix (@abtris)
- [d593554](https://github.com/davglass/license-checker/commit/d593554c1a6b4f7e004e4a23fca88cb7294f8351) feat: add support for csvComponentPrefix (@abtris)
- [5b2eaa1](https://github.com/davglass/license-checker/commit/5b2eaa12d8c54068a01a2cf1fd9bceda8c1f0fa2) added contrib (@davglass)

### v9.0.2 (2017/04/05 14:55 +00:00)
- [f24833b](https://github.com/davglass/license-checker/commit/f24833bce846f608ec3d8c327d335da56c42b895) 9.0.2 (@davglass)
- [#94](https://github.com/davglass/license-checker/pull/94) Merge pull request #94 from timbru31/fix-comma-exclude (@timbru31)
- [2d08a77](https://github.com/davglass/license-checker/commit/2d08a77030362c7ab03ac12358dd524c0be47eee) contrib & changelog (@davglass)

### v9.0.1 (2017/04/05 13:09 +00:00)
- [9f1a1f3](https://github.com/davglass/license-checker/commit/9f1a1f32194a9bb312858f624e303af1ad4f61ca) 9.0.1 (@davglass)
- [c99f612](https://github.com/davglass/license-checker/commit/c99f612fee2d9eb03db0ad41e7918fba6e8ed9) added test for licenseText (@davglass)
- [#93](https://github.com/davglass/license-checker/pull/93) Merge pull request #93 from abtris/licenseText-just-in-customFormat (@abtris)
- [8d7fb2f](https://github.com/davglass/license-checker/commit/8d7fb2f85fc585d8e75ab1459e0b0cef43993ba4) fix: allow escaped commas in exclude string (@timbru31)
- [da4d73c](https://github.com/davglass/license-checker/commit/da4d73c0fe22f50ddbe54364eef5b7bbcc7f1a4) test: fix test, remove check from non custom format (@abtris)
- [dcd3b59](https://github.com/davglass/license-checker/commit/dcd3b59b9457df354fa3095189235b6e030df1a) feat: add licenseText only for customFormat (@davglass)
- [4e02eca](https://github.com/davglass/license-checker/commit/4e02eca6e8bd332ff6d90e5039cacb50f0df1bf3) updated contribs and changelog (@davglass)

### v9.0.0 (2017/04/04 12:26 +00:00)
- [267e80e](https://github.com/davglass/license-checker/commit/267e80e8c89eef70c71a12bf123d8e6d58dd) 9.0.0 (@davglass)
- [057af47](https://github.com/davglass/license-checker/commit/057af47bc52ac526f646ba21496f338689801809) added licenseFile assertion to tests (@davglass)
- [92](https://github.com/davglass/license-checker/pull/92) Merge pull request #92 from abtris/includeFileContent (@abtris)
- [1cda118](https://github.com/davglass/license-checker/commit/1cda118e1aa18cc391ba2765dca89172943a0f) fix: replace problematics chars for easy import to Excel (@abtris)
- [dd6ecab](https://github.com/davglass/license-checker/commit/dd6ecab1a7b1354c4d1d001491f58dbee0f5e3e63) feat: add licenseText (@abtris)
- [6395498](https://github.com/davglass/license-checker/commit/639549890c54a12043244579ad0c0d5972309fb2) changes and contrib (@davglass)

### v8.0.4 (2017/02/15 16:21 +00:00)
- [28d379a](https://github.com/davglass/license-checker/commit/28d379a51da7a47af324bc66eb63736ea02295) 8.0.4 (@davglass)
- [89](https://github.com/davglass/license-checker/pull/89) Merge pull request #89 from Pawamoy/master (@Pawamoy)
- [90](https://github.com/davglass/license-checker/pull/90) Merge pull request #90 from drewfish/bsd-source-code (@drewfish)
- [b704bc1](https://github.com/davglass/license-checker/commit/b704bc18a43dec246302b8ff6933f8fe5f96ee3a6) guess BSD-Source-Code license (@drewfish)
- [8d31eab](https://github.com/davglass/license-checker/commit/8d31eabf6b71d326f3e63215d6530d7f4acc59) Add docs about customFormat option (@pawamoy)
- [2529d31](https://github.com/davglass/license-checker/commit/2529d3118dfae53f1e01056ec99f5c542cb8d63e) changes (@davglass)

### v8.0.3 (2016/11/10 17:04 +00:00)
- [66fd7b5](https://github.com/davglass/license-checker/commit/66fd7b5ad5577cc2c85668a7c6607365c86a2e85) 8.0.3 (@davglass)
- [87](https://github.com/davglass/license-checker/pull/87) Merge pull request #87 from drewfish/guess-cc0 (@drewfish)
- [668bbd3](https://github.com/davglass/license-checker/commit/668bbd38fd5148f99c5263f4e0506e04d5e3f8a0) guess CC0-1.0, upgraded jshint (@drewfish)

### v8.0.2 (2016/11/09 21:57 +00:00)
- [0bdded8](https://github.com/davglass/license-checker/commit/0bdded8cb990e1f5f5acde1c12cc708185a5b866) 8.0.2 (@davglass)
- [ef67bb4](https://github.com/davglass/license-checker/commit/ef67bb454373352343f4a9d4dc978a295bd1544) updated semver (@davglass)
- [14b5c85](https://github.com/davglass/license-checker/commit/14b5c8551c5b85c5ce8aa76fb5367857c891d013) Changelog (@davglass)

### v8.0.1 (2016/11/09 21:51 +00:00)
- [594871d](https://github.com/davglass/license-checker/commit/594871d6197a030121f940500495e6ca7d82f02c) 8.0.1 (@davglass)
- [c243663](https://github.com/davglass/license-checker/commit/c243663b0ab8a0126251292a57b2c47369a5lebf) added contrib (@davglass)
Open Source Used In DNAC Platform Antelope 12235

- [#86](https://github.com/davglass/license-checker/pull/86) Merge pull request #86 from drewfish/copying (@drewfish)
- [380aa26](https://github.com/davglass/license-checker/commit/380aa26799cb527739907484e69c6daed0d5f17a) also support COPYING file (@drewfish)
- [329c367](https://github.com/davglass/license-checker/commit/329c3676d0cd5c5947a53fd205ff23b767b7ff82) handling undefined@undefined error better (@davglass)
- [f193a07](https://github.com/davglass/license-checker/commit/f193a072632a0d660b6361c56c3bd2fe109474) changelog (@davglass)

### v8.0.0 (2016/09/09 13:23 +00:00)
- [44c2c48](https://github.com/davglass/license-checker/commit/44c2c48ae177915523fd89a74474fe0953408bf) 8.0.0 (@davglass)
- [b5f5c6a](https://github.com/davglass/license-checker/commit/b5f5c6a03a8c60c52fdd308f1721744a4baaa99) [fixes #78] - include notice file if found (@davglass)
- [dfcbd6f](https://github.com/davglass/license-checker/commit/dfcbd6f4966ad5e065f71e45cd08ba8fe62ab) changelog (@davglass)
- [99976d0](https://github.com/davglass/license-checker/commit/99976d07980e241384be10341d0305d08de9ca5) 7.1.1 (@davglass)
- [ef857b7](https://github.com/davglass/license-checker/commit/ef857b7e9bab706299a1cc565ec5d144893af424) [fixes #77] fixed readme file name parsing, made it more strict (@davglass)

### v7.1.1 (2016/09/07 19:16 +00:00)
- [28be6d4](https://github.com/davglass/license-checker/commit/28be6d4128a3625cf8ed37ba149e79ed3a41b4fa) changelog (@davglass)

### v7.1.0 (2016/09/07 15:12 +00:00)
- [f8b7f1](https://github.com/davglass/license-checker/commit/f8b7f1c367dfe86342ee5f5cc97171bf97281aa) 7.1.0 (@davglass)
- [60fcef1a](https://github.com/davglass/license-checker/commit/60fcef1a2a4a4ef5add98a08dd8c8a85870db3f4a6) Fixed order of operations for parsing (@davglass)
- [9d4db11](https://github.com/davglass/license-checker/commit/9d4db1127948b8110a1bfe0afbe2d3e63bc812299) Updated changelog (@davglass)

### v7.0.0 (2016/09/06 14:50 +00:00)
- [a3a08da](https://github.com/davglass/license-checker/commit/a3a08da70b680d72405f655e107f36b16b4f5ea) 7.0.0 (@davglass)
- [b993399](https://github.com/davglass/license-checker/commit/b9933994472162cc9c4292cb42f2900e17a5167f) added contrib (@davglass)
- [d783029](https://github.com/davglass/license-checker/commit/d7830297b053496e45b7145b3ca5809ac88b1d1) Fixes #76 - If readme data not found, look for it on disk (@davglass)
- [#75](https://github.com/davglass/license-checker/pull/75) Merge pull request #75 from revov/master (@revov)
- [6204a24](https://github.com/davglass/license-checker/commit/6204a244ee71ae0329bf4f2e1ca6cf37713741f32) Fix incorrect example in README (@revov)
- [90e24ac](https://github.com/davglass/license-checker/commit/90e24ac9ad20591d5ac166cc2974034fb997b30a) Added changelog (@davglass)
- [dd768c6](https://github.com/davglass/license-checker/commit/dd768c68c8b39b596881a2edc582ad4d816879) Added contributors (@davglass)
### v6.0.0 (2016/08/26 19:46 +00:00)
- [4bca140](https://github.com/davglass/license-checker/commit/4bca140db752b7b98ad1e261e5ee965d84995741) 6.0.0 (@davglass)
- [889ff17](https://github.com/davglass/license-checker/commit/889ff17ae0b1a2180f4c9100ca9ea528a264d81) updating tests (@davglass)
- [ed18801](https://github.com/davglass/license-checker/commit/ed188010351c4afbb44db0d32dfa1025de53bf10f) Merge branch 'nodeback' of git://github.com/bengl/license-checker into bengl-nodeback (@davglass)
- [34bb266](https://github.com/davglass/license-checker/commit/34bb26677cd0e25500fa545df4f1e8ae6f216b8cc7) dropping node 0.10 and adding node 6.0 to travis (@davglass)
- [1d30667](https://github.com/davglass/license-checker/commit/1d3066747af7e2cad31f2b7d67d5784c7922bb) updating tests (@davglass)
- [65d7c73](https://github.com/davglass/license-checker/commit/65d7c73f7967111003fcc71a24a4a5391bd48046) Cleaned up some code (@davglass)
- [7f53998](https://github.com/davglass/license-checker/commit/7f53998a06cbb5fd9ff6430b3588a3b3cf81339) Merge branch 'master' of git://github.com/amittkSharma/license-checker into amittkSharma-master (@davglass)
- [28e4462](https://github.com/davglass/license-checker/commit/28e4462543724685080c16232dc2bd57cabb1c30) Added tests and cleaned up linting (@davglass)
- [153857a](https://github.com/davglass/license-checker/commit/153857a251a97d14ac883f6479579844b2980b5aa) Publisher Information incorporated.
- [1265c83](https://github.com/davglass/license-checker/commit/1265c83d186ab1609dbda4124f23f8aa15846) Added ability to output individual license files (@aweber1)
- [49e11d0](https://github.com/davglass/license-checker/commit/49e11d049645d6567e6fd3c778a5f4eb13af888) fix tests (@bengl)
- [26ff6c6](https://github.com/davglass/license-checker/commit/26ff6c63d43c372c9a18f9c0c137608b3d5e4fa) use nodeback/errback (@bengl)

### v5.1.2 (2016/05/04 18:32 +00:00)
- [7101c77](https://github.com/davglass/license-checker/commit/7101c77667ad134fe01f02c298e7df2a4080e311) 5.1.2 (@davglass)
- [867](https://github.com/davglass/license-checker/pull/67) Merge pull request #67 from graingert/patch-2 (@grainert)
- [a6ee290](https://github.com/davglass/license-checker/commit/a6ee290eb7fced220defe8548f80ec74fde3a2) Upgrade read-installed (@grainert)
- [42e29c6](https://github.com/davglass/license-checker/commit/42e29c637940959121842672d74921bece5e8d93) migrating tests from vows to mocha (@davglass)

### v5.1.1 (2016/04/14 13:20 +00:00)
- [6ae0334](https://github.com/davglass/license-checker/commit/6ae03348ca8e5835ac732028a9db664a381cf4) 5.1.1 (@davglass)
- [0720a46](https://github.com/davglass/license-checker/commit/0720a46569f3e12eb398165526944f070e08aad6) Fixes #64 - Turn off color for --json/markdown/csv options (@davglass)

### v5.1.0 (2016/03/09 15:50 +00:00)
- [86772d2](https://github.com/davglass/license-checker/commit/86772d202b63d4a4c6729b9a774193ca6ef5cb3f) 5.1.0 (@davglass)
- [6f1](https://github.com/davglass/license-checker/pull/61) Merge pull request #61 from sgtduck/master (@sgtduck)
- [6547ac1](https://github.com/davglass/license-checker/commit/6547ac1ded58560c7c4abdc68fa69d17b811fe1e) Removed obsolete $key.name assignment from tests (@sgtduck)
- [00b18a6](https://github.com/davglass/license-checker/commit/00b18a6012387e1a4a8284e07bf2d08740e03ddd) Added a --relativeLicensePath option to display the license file relatively to the current path. (@sgtdck)
- [629da20](https://github.com/davglass/license-checker/commit/629da201e0d1170057fd9efe05efc82932f1bda) Added latest contributors (@davglass)

### v5.0.1 (2016/02/08 15:57 +00:00)
- [33731fca](https://github.com/davglass/license-checker/commit/33731fca482936a7a18ffe3fca2f3cd41e3a625) 5.0.1 (@davglass)
- [fe06eaef](https://github.com/davglass/license-checker/commit/fe06eaef8844d0500a008d576a221dd65131de) Fixed replace (@davglass)
- [997914f](https://github.com/davglass/license-checker/commit/997914f4d6f94030452d4d9bf549eb156ae8ef7c0) fix git+https URLs (@graingert)
- [#57](https://github.com/davglass/license-checker/pull/57) Merge pull request #57 from Damien-Larmine/doc-license-asterisk (@Damien-Larmine)
- [2adff2d](https://github.com/davglass/license-checker/commit/2adff2da98a5af79497a162e4998eb5892ab31c3) Explicit the asterisk in license report (@Damien-Larmine)
- [b2fc9cf](https://github.com/davglass/license-checker/commit/b2fc9cf2a8ea008c7593037d046f245a7ce7f25) 5.0.0.0 (@davglass)
- [7f71c53](https://github.com/davglass/license-checker/commit/7f71c53833cf1318ffe84215fcb129f19ce4fb5) updated contributors (@davglass)

### v5.0.0 (2016/01/22 15:31 +00:00)
- [#55](https://github.com/davglass/license-checker/pull/55) Merge pull request #55 from davglass/devflag (@davglass)
- [f1d8f2](https://github.com/davglass/license-checker/commit/f1d8f20d5d7248a4f61e2fc2979fdfe91a6ec6) supporting --production and --development flags (@davglass)
- [#53](https://github.com/davglass/license-checker/pull/53) Merge pull request #53 from swashcap/enhancement/#46-verbose-output (@swashcap)
- [#54](https://github.com/davglass/license-checker/pull/54) Merge pull request #54 from davglass/travis-updates (@davglass)
- [2ac80d9d](https://github.com/davglass/license-checker/commit/2ac80d9d2d59340698630144256096a9e23f22) fixing tests in other node versions (@davglass)
- [78820ac](https://github.com/davglass/license-checker/commit/78820ac33ab8aedd7409ce8a5f887705c8b2848e) travis config updates (@davglass)
- [f94263b](https://github.com/davglass/license-checker/commit/f94263b203046e2f33d64997b0261498010eb38d) Update documentation for debug use. (@swashcap)
- [429e444d](https://github.com/davglass/license-checker/commit/429e444d97527221e2f2bb03af9d67820015446f) Change `console.error` call to use debug module. (@swashcap)
- [7312daf](https://github.com/davglass/license-checker/commit/7312daf7af195a1b9d1bf290fae15963fbb2ab4) Add debug to project. (@swashcap)

### v4.2.0 (2015/10/20 13:44 +00:00)
- [3583d92](https://github.com/davglass/license-checker/commit/3583d925e96fe9f63b3bbf0110b11e8484aed0e7) 4.2.0 (@davglass)
- [981e6da](https://github.com/davglass/license-checker/commit/981e6da898084dea88d54e29c56da93c5313fb6) Fixes #49: Default to unknown if licenses null and added ISC support (@davglass)

### v4.1.2 (2015/09/29 13:46 +00:00)
Open Source Used In DNAC Platform Antelope 12238

- [ba946dd](https://github.com/davglass/license-checker/commit/ba946dd1065e762c36b27dcecd9bb7389ad18d4) 4.1.2 (@davglass)
- [#48](https://github.com/davglass/license-checker/pull/48) Merge pull request #48 from CantemoInternal/master (@CantemoInternal)
- [e314a6d](https://github.com/davglass/license-checker/commit/e314a6d1f505d19478c33871640a763fabff6abd) Fix json, csv and markdown output for license-checker (@mattiasa)

### v4.1.1 (2015/09/12 14:39 +00:00)
- [c918314](https://github.com/davglass/license-checker/commit/c918314bfc687b647d1d288bb5857dd1caac1393) 4.1.1 (@davglass)
- [#47](https://github.com/davglass/license-checker/pull/47) Merge pull request #47 from tmpvar/ensure-bin-runs (@tmpvar)
- [a842eca](https://github.com/davglass/license-checker/commit/a842eca4f008c07df3039dcaccd4641d7273625) add missing ) (@tmpvar)
- [1d513bb](https://github.com/davglass/license-checker/commit/1d513bb673d726f34286061446f965e46d96170e) ensure bin/license-checker works at a high-level (@tmpvar)

### v4.1.0 (2015/09/09 14:52 +00:00)
- [a4bb437](https://github.com/davglass/license-checker/commit/a4bb437fafa806c0832877da26d7a107863d0a1f) 4.1.0 (@davglass)
- [ddd317a](https://github.com/davglass/license-checker/commit/ddd317a5a09344f0021812b8dece2f39bbbf07a0) fixed all of the windows line endings (@davglass)
- [11e6487](https://github.com/davglass/license-checker/commit/11e648704484435e1e6f8e0f9b0f13e6b2737a6b) Fix issues caused by merge (@ptusch)
- [313a7ed](https://github.com/davglass/license-checker/commit/313a7edc3dbdf8d50e43ba8da89306d3c89326bc) Merge remote-tracking branch `remotes/upstream/master` into ft_custom_format (@ptusch)
- [1f5e326](https://github.com/davglass/license-checker/commit/1f5e326d2ca91f8a396b9e02ef3eac9fd82a82) added contributors to package.json (@davglass)
- [#45](https://github.com/davglass/license-checker/pull/45) Merge pull request #45 from hufsm/ft_error_handling (@hufsm)
- [c806034](https://github.com/davglass/license-checker/commit/c806034ab66eaec1297d0c23af536b290ba8c81) Remove unused var statement (@ptusch)
- [778b1fa](https://github.com/davglass/license-checker/commit/778b1faa9e1be0c81654516502522d2dbf17ef) More tests to fulfill the required % (@ptusch)
- [22257d9](https://github.com/davglass/license-checker/commit/22257d95490765db816578cc48c2b5b62699f1aa) Add tests for licenses (WTFPL) (@ptusch)
- [0fe7561](https://github.com/davglass/license-checker/commit/0fe75611a73970fe9e9e0aabb2d4dbee5250435b2) Remove unused filter option (never used) (@ptusch)
- [83c3a10](https://github.com/davglass/license-checker/commit/83c3a10493a5a0f5fbaf6c0e8b5c03f9cd4f65d6) Change license shortcut for WTFPL (@ptusch)
- [3e348db](https://github.com/davglass/license-checker/commit/3e348dbe5fee7bd356ebebf3cd34e69329dbc8c1) More tests for the custom formatter (@ptusch)
- [96e2c6f](https://github.com/davglass/license-checker/commit/96e2c6f24c509df2bf3f6b8b83a613fe8dd9ecc) Add tests for customFormatter (@ptusch)
- [c9948c4](https://github.com/davglass/license-checker/commit/c9948c4b7b2cccd053ba63f4f2a6a6d0a46493) Add test configs (@ptusch)
- [4b2dea0](https://github.com/davglass/license-checker/commit/4b2dea0c124f6a63abd2129625825c3b5a6570) Fix merge issues - semicolon slipped through (@ptusch)
Open Source Used In DNAC Platform Antelope

### v4.0.0 (2015/09/04 13:40 +00:00)
- [3fbba5b](https://github.com/davglass/license-checker/commit/3fbba5bd5a489ea8c17328eabf71c4d5e9beb17e) 4.0.0 (@davglass)
- [ce8d661](https://github.com/davglass/license-checker/commit/ce8d66108a49af7855c16231ce675c567a2ffe7f) tests: added 100% coverage for license file (@davglass)
- [#42](https://github.com/davglass/license-checker/pull/42) Merge pull request #42 from ptusch/ft_onlyunknown (@ptusch)
- [8929e90](https://github.com/davglass/license-checker/commit/8929e90413be0362f19e755048e9218c372abf68) Add tests for the error generation (@ptusch)
- [1a93a41](https://github.com/davglass/license-checker/commit/1a93a41569a53a1580c94c2e903a5bc127ad2692) Add error handling (@ptusch)
- [6b79145](https://github.com/davglass/license-checker/commit/6b791457463dca71b22b09e0ef16bbf982db682) Add additional check to negate the test (@ptusch)
- [559e72c](https://github.com/davglass/license-checker/commit/559e72ccab2420bc42f28b498702dad13e9f67a) Add test for the first function call. (@ptusch)
- [faf8ffd](https://github.com/davglass/license-checker/commit/faf8ffd69fd8ae4754584a9b47b57a7b6ea738856) Remove trivial UNKNOWN check (@ptusch)
- [6fa44ef](https://github.com/davglass/license-checker/commit/6fa44ef8c045ab2baa1291b45990e52f52a0d0e2) Merge branch 'master' into ft_custom_format (@ptusch)
- [eef23d9](https://github.com/davglass/license-checker/commit/eef23d991cfc04dc6980fb13f461408da822b1f) Merge branch 'master' into ft_onlyunknown (@ptusch)
- [c287959](https://github.com/davglass/license-checker/commit/c2879592858805d7ecd0304a39c5ef079e6a7c) Add option to only show unknown licenses (@ptusch)
- [#43](https://github.com/davglass/license-checker/pull/43) Merge pull request #43 from senotrusov/master (@senotrusov)
- [d7a663f](https://github.com/davglass/license-checker/commit/d7a663f9dee4d7a648a62c9dbd84d5667ee540) Add test for exclude option. (@senotrusov)
- [45a27de](https://github.com/davglass/license-checker/commit/45a27de02a3c1d57a86e3f8e14b6e335788c23b) Add option to exclude from the output modules which licenses are in the comma-separated list (@senotrusov)
- [#41](https://github.com/davglass/license-checker/pull/41) Merge pull request #41 from ptusch/ft_path_to_unknown_dependency (@ptusch)
- [1ff5f22](https://github.com/davglass/license-checker/commit/1ff5f225fd2ba97edfd3e3f6462981832740c19) Add option to see the path of a dependency if unknown (@ptusch)
- [ea4d629](https://github.com/davglass/license-checker/commit/ea4d6297de32d94201bb4786da7ba824d928c8f2) Add additional 'undefined' checks (@ptusch)
- [af043a2](https://github.com/davglass/license-checker/commit/af043a2c11af45d28bfe71914cd10672855b3297) Merge branch 'master' into ft_custom_format (@ptusch)
- [2d94a19](https://github.com/davglass/license-checker/commit/2d94a19ecee177836345a4227893a82d956141b2e) updating the docs for --unknown to make it clear (@davglass)
- [38a2bd2](https://github.com/davglass/license-checker/commit/38a2bd28a9597fd0b6d844258f1070054b38bbcf) Fix codestyle and whitespaces (@ptusch)
- [dd173ba](https://github.com/davglass/license-checker/commit/dd173badc65368dc6140476f5eae3768bf9618) Add --customPath in README.md (@ptusch)
- [911fc1c](https://github.com/davglass/license-checker/commit/911fc16b5cc27a32aaac60bb409c08be138bc707) Finalize the customFormat (markdown) (@ptusch)
- [8102091](https://github.com/davglass/license-checker/commit/8102091f7949fa8b737cf803e4aeed3624b81148) Add functionality to add a custom format (@ptusch)

### v3.1.0 (2015/06/29 14:25 +00:00)
- [59e1e58](https://github.com/davglass/license-checker/commit/59e1e58ca0bea1cd38e3da1712fe7ca624d860b0) 3.1.0 (@davglass)
- [0e33f7a](https://github.com/davglass/license-checker/commit/0e33f7a2cb9fa69d13d1eb2b569f70f885e09e08) reformatted code, fixed lint issues and added tests (@davglass)
- [83a7f09](https://github.com/davglass/license-checker/commit/83a7f0933d1cfc331279e53f1cc13e4e9ace91) Merge branch 'asMarkdown' of git://github.com/ptusch/license-checker into ptusch-asMarkdown (@davglass)
- [703ad47](https://github.com/davglass/license-checker/commit/703ad4714173559e05267e8fd24e74eb9d3ae0) fixing tests (@davglass)
- [2d9fffd](https://github.com/davglass/license-checker/commit/2d9fffd75192e9812692c9099481d23e265bb493) Added Markdown Option (@ptusch)
- [67dca36](https://github.com/davglass/license-checker/commit/67dca365b941a7265e4a98d2d533b2c761b169) Remove color tags (@ptusch)

### v3.0.3 (2015/06/03 13:35 +00:00)
- [7475c12](https://github.com/davglass/license-checker/commit/7475c129bf21298e4161d9692a962ae537401e8) 3.0.3 (@davglass)
- [c7dae0f](https://github.com/davglass/license-checker/commit/c7dae0fe5317a17b3aef61e8c118f6ac6bb43a56) spdx license (@davglass)

### v3.0.2 (2015/05/12 20:29 +00:00)
- [01f3b53](https://github.com/davglass/license-checker/commit/01f3b5384998be6243d38e0d95a4478e5fe3ca5) 3.0.2 (@davglass)
- [fc0c0e83](https://github.com/davglass/license-checker/commit/fc0c0e832ee291ce55e53ef6b37a7b9d89f191bc7) defensive code, fixes #34 (@davglass)

### v3.0.1 (2015/05/04 12:28 +00:00)
- [0479570](https://github.com/davglass/license-checker/commit/0479570af87ecce9b37fbb25afaee7faae6636) 3.0.1 (@davglass)
- [d85e355](https://github.com/davglass/license-checker/commit/d85e355b55998952facf202ebaf8d374efc69a7e) Checking for valid path first (@davglass)

### v3.0.0 (2015/04/24 14:38 +00:00)
- [e22b61d](https://github.com/davglass/license-checker/commit/e22b61d5d32fe7ffbf18627f55a97e1863a78096) 3.0.0 (@davglass)
- [fde9635](https://github.com/davglass/license-checker/commit/fde963583dd207eca4badfb3e0d33a038f17b5) always try to find and log the license file (@davglass)

### v2.0.2 (2015/01/26 18:18 +00:00)
- [490f31d](https://github.com/davglass/license-checker/commit/490f31d6820f068814cfe27adf3af1ac6d58f733) 2.0.2 (@davglass)
- [a3fefdb](https://github.com/davglass/license-checker/commit/a3fefdb8fafd7ec64714defb793d757a9e8ad065) fixes #30 (@davglass)

### v2.0.1 (2015/01/12 18:34 +00:00)
- [898fbd5](https://github.com/davglass/license-checker/commit/898fbd53fa289338c8a9a1735ee440c2a6c525) 2.0.1 (@davglass)
- [2da23db](https://github.com/davglass/license-checker/commit/2da23db2f240e7e73efdf93cc554e4d7783af225) cleaned up spacing and moved vars around (@davglass)
- [d47dbe4](https://github.com/davglass/license-checker/commit/d47dbe45c992dbbd15bf48eaf7f18ee0ca737) fixes #29 (@davglass)

### v2.0.0 (2014/12/04 16:34 +00:00)
- [f94edfe](https://github.com/davglass/license-checker/commit/f94edfec7a9914b4a36f455e8bfec6230d6f7a5e) 2.0.0 (@davglass)
- [f4c7b1a](https://github.com/davglass/license-checker/commit/f4c7b1a3d4e9c0e4753a9b26de751714fe7f051e) syntax cleanup (@davglass)
- [d189623](https://github.com/davglass/license-checker/commit/d1896234a5649621ee57ff9a901877d7693179d194) Document CLI options in Readme. (@timoxley)
- [11e204c](https://github.com/davglass/license-checker/commit/11e204c5c078aaee1f165fa010da229d2a9fb4b5) Default to --no-colour for json, csv & file output. (@timoxley)
- [7e80d50](https://github.com/davglass/license-checker/commit/7e80d50c733f8ee42c7b4a79dc3344b3dd330c0f) Add --out flag for writing an output file. (@timoxley)
- [47bffe9](https://github.com/davglass/license-checker/commit/47bffe9e0e249b2847c8ed4f3eaaadb0d6d771b9) Always print to stdout, don't write files. (@timoxley)
- [b725080](https://github.com/davglass/license-checker/commit/b725080d2dbdadbfe0994867577d4c5e3dc0ee2a) Don't print info messages to stdout. (@timoxley)

### v1.0.1 (2014/11/27 15:23 +00:00)
- [e60f792](https://github.com/davglass/license-checker/commit/e60f792159351aa7ccf1ff7a73e289aafe261a5) 1.0.1 (@davglass)
- [3f74178](https://github.com/davglass/license-checker/commit/3f7417822dc88049781d1be4ed84190d327df15c) Merge branch 'fixinfinitercursion' of github.com:yzapuchlak/license-checker into yzapuchlak-fixinfinitercursion (@davglass)
- [dc78eaf0](https://github.com/davglass/license-checker/commit/dc78eaf0dca54625340c89fec2dec6818dc4cc1c) sticking vows 0.8.0 for testing (@davglass)
- [bd549b3](https://github.com/davglass/license-checker/commit/bd549b3b68ee74fdd0e8dcal0d5f88a8bb51ca6) Fixed issue where circular dependencies in dependency tree would cause the flatten function to infinitely recurse. If the current key is found to be already in the data object (which means it was already processed), we stop processing that key and return the data object. (@yzapuchlak)

### v1.0.0 (2014/11/21 13:56 +00:00)
- [917193d](https://github.com/davglass/license-checker/commit/917193dab27147534976b71307a77f48198ca322)
1.0.0 (@davglass)
- [63e5f9c](https://github.com/davglass/license-checker/commit/63e5f9c77e348720b3867171bccc67541e846d088) added options.filter to checker.init (@davglass)

### v0.1.2 (2014/11/20 15:48 +00:00)
- [8e5f5f0](https://github.com/davglass/license-checker/commit/8e5f5f07f83252e4039f6c7d16cc266acc1a33a) 0.1.2 (@davglass)
- [3d7a74e](https://github.com/davglass/license-checker/commit/3d7a74e90122f908057b2008ab29d8bd49011291) fixed lint issue (@davglass)

### v0.1.1 (2014/11/20 15:47 +00:00)
- [750d507](https://github.com/davglass/license-checker/commit/750d5079d32b44ec984ff7ed672498d1b197235) 0.1.1 (@davglass)
- [94522e0](https://github.com/davglass/license-checker/commit/94522e0994343dfdf163c3f2c5cabc3e6aa196cf8) fixed merge issues (@davglass)

### v0.1.0 (2014/11/20 15:41 +00:00)
- [8127f5e](https://github.com/davglass/license-checker/commit/8127f5e79eecebd7fd86cd5f694080e4183dc11) 0.1.0 (@davglass)
- [76c71ac](https://github.com/davglass/license-checker/commit/76c71ac46baeca8c349fe5892dcb659e859b608) Updated to newer read-installed (@davglass)

### v0.0.10 (2014/08/12 12:48 +00:00)
- [807507d](https://github.com/davglass/license-checker/commit/807507df1da293bc036a0dffe725a9f11331ff2b) 0.0.10 (@davglass)
- [e39ad9b](https://github.com/davglass/license-checker/commit/e39ad9bce6b454772e241e2e7aba8f40d5bbb) Added checking that the license file is a normal file, because otherwise an appropriately named directory crashes the license-checker through EISDIR, illegal operation on a directory.
- [40ca4da](https://github.com/davglass/license-checker/commit/40ca4dadbc163ca774c87b1de249bd295952faa9) colorize (@couchand)

### v0.0.9 (2014/07/29 01:48 +00:00)
- [c14ef79](https://github.com/davglass/license-checker/commit/c14ef79f10950f0f985e7e0b3138147b98ab1ec) 0.0.9 (@davglass)
- [ba1b66b](https://github.com/davglass/license-checker/commit/ba1b666b172a2f9f8dbc65f40206820bc0916b) Fixes #12 - Using some nicer regexes (@davglass)
- [bce94d5](https://github.com/davglass/license-checker/commit/bce94d5a8ab2f146356851776b2543df65f750) case insensitive check (@davglass)
- [87c3406](https://github.com/davglass/license-checker/commit/87c3406ae2507eaf1d00d592f6db59a167e48) Allow for LICENCE (@gdw2)

### v0.0.8 (2014/07/07 20:10 +00:00)
- [ad6ee2a](https://github.com/davglass/license-checker/commit/ad6ee2e1813e698e7f995e98dcdabfd9616fa33) 0.0.8 (@davglass)
- [b8c5815](https://github.com/davglass/license-checker/commit/b8c58151135f79d1fbd1c5194d034c012073f) added tests and refactored the code a little (@davglass)
- [275f276](https://github.com/davglass/license-checker/commit/275f276d66e6d5635f9c5b57e9a978e286f2d174) Added --csv option to readme file
Open Source Used In DNAC Platform Antelope 12243

- [d91eb72](https://github.com/davglass/license-checker/commit/d91eb7289466853ae6e884488b953b4d3d98ebab5) Apparently abbrev is now at version 1.0.5 which caused the unit test to fail.
- [40b4a42](https://github.com/davglass/license-checker/commit/40b4a420e520c2008b9395d1a8fd38b56f846f5) Modified changes based on lshint output during Travis build
- [54e2390](https://github.com/davglass/license-checker/commit/54e2390cefd1dc664a15cd3a843472727889f48b) Added csv output option similar to the existing JSON output option. Requires an output filename.

### v0.0.7 (2014/02/26 04:52 +00:00)
- [1ce6d42](https://github.com/davglass/license-checker/commit/1ce6d42147b2d0237870818e8059003b331816cf) 0.0.7 (@davglass)
- [7d04195](https://github.com/davglass/license-checker/commit/7d041959540c68b3c5f427007c43cd6c73c6adfb) Added not check and a proper test (@davglass)
- [40d6e25](https://github.com/davglass/license-checker/commit/40d6e2580946c3cee1d408ebe28ff523852e8eb2) Undefined handling (@garrows)
- [90217d0](https://github.com/davglass/license-checker/commit/90217d070f6d2cf76e6b4e7b16160d5eef6ae840d) travis Stuff (@davglass)
- [2bd76c8](https://github.com/davglass/license-checker/commit/2bd76c894740cd83f62466e0c7341832c6432d8b) 0.0.6 (@davglass)
- [6f390ff](https://github.com/davglass/license-checker/commit/6f390f660b31a07939e6d325796bdcbefb5907) Added some better tests (@davglass)
- [2dbac3](https://github.com/davglass/license-checker/commit/2dbac37b5795ceb6db53452d0e41d1c4bc3153f) Looking for WTFPL as well as the string (@davglass)
- [5d91a07](https://github.com/davglass/license-checker/commit/5d91a073a8b76ee9a497c1e3b83198ce97e144fd) Flattened array of one (@davglass)
- [e95da82](https://github.com/davglass/license-checker/commit/e95da820e63e0d56c4567f1cfad07142d3110eca) Fixed extra var (@davglass)
- [fe8f2f7](https://github.com/davglass/license-checker/commit/fe8f2f7d39d520c0c4d87d34c9d4f55d59f4222d) Removing unnecessary log statement (@paulmand3l)
- [72a6f24](https://github.com/davglass/license-checker/commit/72a6f2403de43ecf5ec07bb504cb931ce536a3) Adding license checks in the LICENSE file (and variations) (@paulmand3l)
- [#7](https://github.com/davglass/license-checker/pull/7) Merge pull request #7 from jamesbloomer/master (@jamesbloomer)
- [a81ee0](https://github.com/davglass/license-checker/commit/a81ee0a27f580688e371cc15015e848f591bb92) Fixed missing “install” in npm -g (@jamesbloomer)
- [416b012](https://github.com/davglass/license-checker/commit/416b01219e07a8274e05cd883ec050376902adeb) Updated Version (@davglass)
- [6d9bd1e](https://github.com/davglass/license-checker/commit/6d9bd1e195202c871693d4b1e9bac8c055683e8b) Fixed lint issue and added a test placeholder. Will add more tests soon (@davglass)
- [655a316](https://github.com/davglass/license-checker/commit/655a316f6448daf3c449fbb88e8878a95a8b153e) Added Travis test file (@davglass)
- [99895df](https://github.com/davglass/license-checker/commit/99895df6d3ac0f9ae4e8c935311fb1b3f88a688e) Version Bump (@davglass)
- [#4](https://github.com/davglass/license-checker/pull/4) Merge pull request #4 from badunk/refactor (@badunk)
- [1123a1b](https://github.com/davglass/license-checker/commit/1123a1baefc1975f8c36cb59952254cd66c31017) Merge branch 'master' of https://github.com/davglass/license-checker (@badunk)
- [191612e](https://github.com/davglass/license-checker/commit/191612e5c21a01bd453368299a14cc5ff1c6fb13) Version Bump (@davglass)
- [#2](https://github.com/davglass/license-checker/pull/2) Merge pull request #2 from badunk/circular (@badunk)
- [9b5a117](https://github.com/davglass/license-checker/commit/9b5a1175892254e96d72f3d8dc793dce42bde37) fixed special case when an empty array is passed to license/licenses (@badunk)
- [025668b](https://github.com/davglass/license-checker/commit/025668b0a62b0d5b58662ce6aead91de909a6fef) fixed bug with multiple licenses objects not showing (@badunk)
- [e6bd5f2](https://github.com/davglass/license-checker/commit/e6bd5f2a8e5f9ebed29982014fa4d4586f63d06a) cleaned up the code (@badunk)
- [e6bd5f2](https://github.com/davglass/license-checker/commit/e6bd5f2a8e5f9ebed29982014fa4d4586f63d06a) added check for circular dependencies (@badunk)
- [825b40c](https://github.com/davglass/license-checker/commit/825b40cb61fee36dd60751daa9d44e28853587ac) Version Bump (@davglass)
- [91](https://github.com/davglass/license-checker/pull/1) Merge pull request #1 from mwilliamson/master (@mwilliamson)
- [2f24319](https://github.com/davglass/license-checker/commit/2f24319dd8dd212a1f78ad91a7fee21d50ab762) Don't ignore license field if it's a string (@mwilliamson)
- [7b8c34f](https://github.com/davglass/license-checker/commit/7b8c34f75b04a7541bf23d596bbbae5447492c53) Initial Commit (@davglass)

Authors: Loren Merritt <loremm@u.washington.edu>
Anton Mitrofanov <BugMaster@narod.ru>
Jason Garrett-Glaser <darkshikari@gmail.com>
Henrik Gramner <hengar-6@student.ltu.se>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

/*
Copyright (c) 2010, Linden Research, Inc.
Copyright (c) 2012, Joshua Bell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2017 CoderPuppy

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CC0 1.0 Universal

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and
subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;

ii. moral rights retained by the original author(s) and/or performer(s);

iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;

iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;

v. rights protecting the extraction, dissemination, use and reuse of data in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.
2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.

b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning the Work, express, implied, statutory or otherwise, including without limitation warranties of title, merchantability, fitness for a particular purpose, non infringement, or the absence of latent or other defects, accuracy, or the present or absence of errors, whether or not
discoverable, all to the greatest extent permissible under applicable law.

c. Affirmer disclaims responsibility for clearing rights of other persons
that may apply to the Work or any use thereof, including without limitation
any person's Copyright and Related Rights in the Work. Further, Affirmer
disclaims responsibility for obtaining any necessary consents, permissions
or other rights required for any use of the Work.

d. Affirmer understands and acknowledges that Creative Commons is not a
party to this document and has no duty or obligation with respect to this
CC0 or use of the Work.

For more information, please see
<http://creativecommons.org/publicdomain/zero/1.0/>
Copyright 2014 jQuery Foundation and other contributors
http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2012-2014 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schlueter
Copyright (c) 2014-2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

source: https://github.com/johan/world.geo.json

license:
Unless otherwise noted in individual files or directories, this is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <http://unlicense.org/>

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some
semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

   a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive
site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do
not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End
Copyright 2009 Nokia Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Google LLC and its affiliates ("Google") own all legal right, title and interest in and to the content decryption module software ("Software") and related documentation, including any intellectual property rights in the
Software. You may not use, modify, sell, or otherwise distribute the Software without a separate license agreement with Google. The Software is not open source software.

If you are interested in licensing the Software, please contact www.widevine.com

### v4.0.1 (2016-12-14)

#### WHOOPS

* [fb9b1ce](https://github.com/npm/nopt/commit/fb9b1ce57b3c69b4f7819015be87719204f77ef6)
  Merged so many patches at once that the code fencing
  ([@adius](https://github.com/adius)) added got broken. Sorry,
  ([@adius](https://github.com/adius))!
  ([@othiym23](https://github.com/othiym23))

### v4.0.0 (2016-12-13)

#### BREAKING CHANGES

* [651d447](https://github.com/npm/nopt/commit/651d4473946096d341a480bbe56793de3fc706aa)
  When parsing String-typed arguments, if the next value is `''`, don’t simply
  swallow it. ([@samjonester](https://github.com/samjonester))

#### PERFORMANCE TWEAKS

* [3370ce8](https://github.com/npm/nopt/commit/3370ce87a7618ba228883861db84ddbcdf252a9)
  Simplify initialization. ([@elidoran](https://github.com/elidoran))
* [356e58e](https://github.com/npm/nopt/commit/356e58e3b3b431a4b1af7fd7bdec44c2c0526a09)
  Store `Array.isArray(types[arg])` for reuse. ([@elidoran](https://github.com/elidoran))
* [0d95e90](https://github.com/npm/nopt/commit/0d95e90515844f266015b56d2c80b94e5d14a07e)
  Interpret single-item type arrays as a single type. ([@samjonester](https://github.com/samjonester))
* [07cc69d3](https://github.com/npm/nopt/commit/07cc69d3b5186450941bb505550bceb78a0e925)
  Simplify key-value extraction. ([@elidoran](https://github.com/elidoran))
* [39bf6e5c](https://github.com/npm/nopt/commit/39bf6e5c65ac47f60cd4a1fbeece5cd4c834c254)
  Only call `Date.parse(val)` once. ([@elidoran](https://github.com/elidoran))
* [934943d](https://github.com/npm/nopt/commit/934943dffeceb55123a2b15959f2e2a359319a5dbd)
  Use `osenv.home()` to find a user’s home directory instead of assuming it’s
  always `~SHOME`. ([@othiym23](https://github.com/othiym23))

#### TEST & CI IMPROVEMENTS

* [326ffff](https://github.com/npm/nopt/commit/326ffff7f78a00bcd316adecf69075f8a8093619)
  Fix `/tmp` test to work on Windows.
  ([@elidoran](https://github.com/elidoran))
* [c89d31a](https://github.com/npm/nopt/commit/c89d31a49d14f2238bc6672db08dafa697bbc57f1b)
  Only run Windows tests on Windows, only run Unix tests on a Unix.

Open Source Used in DNAC Platform Antelope 12253
#### DOC TWEAKS

* ["affd3d1"](https://github.com/npm/nopt/commit/affd3d1d0addffa93006397b2013b18447339366)
  Refresh Travis to run the tests against the currently-supported batch of npm versions. (@helio[https://github.com/helio-frota])

* ["55f9449"](https://github.com/npm/nopt/commit/55f94497d163ed4d16dd55fd6c4fb95cc440e66d)
  `tap@8.0.1` (@othiym23[https://github.com/othiym23])

### 1.2.11, January 15th, 2017

Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2010 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@bzip.org
bzip2/libbzip2 version 1.0.6 of 6 September 2010

ISC License

Copyright (c) 2018, Andrea Giammarchi, @WebReflection

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License

Copyright (C) 2019 Erik Barke, Monounity.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2010-2015 Benjamin Peterson

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
// Copyright (c) 2017 The Polymer Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
//   notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
//   copyright notice, this list of conditions and the following disclaimer
//   in the documentation and/or other materials provided with the
//   distribution.
// * Neither the name of Google Inc. nor the names of its
//   contributors may be used to endorse or promote products derived from
//   this software without specific prior written permission.
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (C) 1997 Gregory Pietsch

[These files] are hereby placed in the public domain without restrictions. Just
give the author credit, don't claim you wrote it or prevent anyone else from
using it.
LZMA SDK is placed in the public domain.
COPYRIGHT NOTICE, DISCLAIMER, and LICENSE

PNG Reference Library License version 2

* Copyright (c) 1995-2019 The PNG Reference Library Authors.
* Copyright (c) 2018-2019 Cosmin Truta.
* Copyright (c) 1996-1997 Andreas Dilger.
* Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

I. The origin of this software must not be misrepresented; you
must not claim that you wrote the original software. If you
use this software in a product, an acknowledgment in the product
documentation would be appreciated, but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Mans Rullgard
Cosmin Truta
Gilles Vollant
James Yu
Mandar Sahastrabuddhe
Google Inc.
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
libpng versions 0.89, June 1996, through 0.96, May 1997, are
Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88,
and are distributed according to the same disclaimer and license as
libpng-0.88, with the following individuals added to the list of
Contributing Authors:

  John Bowler
  Kevin Bracey
  Sam Bushell
  Magnus Holmgren
  Greg Roelofs
  Tom Tanner

Some files in the "scripts" directory have other copyright owners,
but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors"
is defined as the following set of individuals:

  Andreas Dilger
  Dave Martindale
  Guy Eric Schalnat
  Paul Schmidt
  Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing
Authors and Group 42, Inc. disclaim all warranties, expressed or
implied, including, without limitation, the warranties of
merchantability and of fitness for any purpose. The Contributing
Authors and Group 42, Inc. assume no liability for direct, indirect,
incidental, special, exemplary, or consequential damages, which may
result from the use of the PNG Reference Library, even if advised of
the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this
source code, or portions hereof, for any purpose, without fee, subject
to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not
   be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

APPLE PUBLIC SOURCE LICENSE Version 2.0 - August 6, 2003

Please read this License carefully before downloading this software. By downloading or using this software, you are agreeing to be bound by the terms of this License. If you do not or cannot agree to the terms of this License, please do not download or use the software.

Apple Note: In January 2007, Apple changed its corporate name from "Apple Computer, Inc." to "Apple Inc." This change has been reflected below and copyright years updated, but no other changes have been made to the APSL 2.0.

1. General; Definitions. This License applies to any program or other work which Apple Inc. ("Apple") makes publicly available and which contains a notice placed by Apple identifying such program or work as "Original Code" and stating that it is subject to the terms of this Apple Public Source License version 2.0 ("License"). As used in this License:

1.1 "Applicable Patent Rights" mean: (a) in the case where Apple is the grantor of rights, (i) claims of patents that are now or hereafter acquired, owned by or assigned to Apple and (ii) that cover subject matter contained in the Original Code, but only to the extent necessary to use, reproduce and/or distribute the Original Code without infringement; and (b) in the case where You are the grantor of rights, (i) claims of patents that are now or hereafter acquired, owned by or assigned to You and (ii) that cover subject matter in Your Modifications, taken alone or in combination with Original Code.

1.2 "Contributor" means any person or entity that creates or contributes to the creation of Modifications.

1.3 "Covered Code" means the Original Code, Modifications, the combination of Original Code and any Modifications, and/or any respective portions thereof.

1.4 "Externally Deploy" means: (a) to sublicense, distribute or otherwise make Covered Code available, directly or indirectly, to anyone other than You; and/or (b) to use Covered Code, alone or as part of a Larger Work, in any way to provide a service, including but not limited to delivery of content, through electronic communication with a client other than You.

1.5 "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.
1.6 "Modifications" mean any addition to, deletion from, and/or change to, the substance and/or structure of the Original Code, any previous Modifications, the combination of Original Code and any previous Modifications, and/or any respective portions thereof. When code is released as a series of files, a Modification is: (a) any addition to or deletion from the contents of a file containing Covered Code; and/or (b) any new file or other representation of computer program statements that contains any part of Covered Code.

1.7 "Original Code" means (a) the Source Code of a program or other work as originally made available by Apple under this License, including the Source Code of any updates or upgrades to such programs or works made available by Apple under this License, and that has been expressly identified by Apple as such in the header file(s) of such work; and (b) the object code compiled from such Source Code and originally made available by Apple under this License.

1.8 "Source Code" means the human readable form of a program or other work that is suitable for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an executable (object code).

1.9 "You" or "Your" means an individual or a legal entity exercising rights under this License. For legal entities, "You" or "Your" includes any entity which controls, is controlled by, or is under common control with, You, where "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

2. Permitted Uses; Conditions & Restrictions. Subject to the terms and conditions of this License, Apple hereby grants You, effective on the date You accept this License and download the Original Code, a world-wide, royalty-free, non-exclusive license, to the extent of Apple's Applicable Patent Rights and copyrights covering the Original Code, to do the following:

2.1 Unmodified Code. You may use, reproduce, display, perform, internally distribute within Your organization, and Externally Deploy verbatim, unmodified copies of the Original Code, for commercial or non-commercial purposes, provided that in each instance:

(a) You must retain and reproduce in all copies of Original Code the copyright and other proprietary notices and disclaimers of Apple as they appear in the Original Code, and keep intact all notices in the Original Code that refer to this License; and

(b) You must include a copy of this License with every copy of Source Code of Covered Code and documentation You distribute or Externally Deploy, and You may not offer or impose any terms on such Source Code that alter or restrict this
License or the recipients' rights hereunder, except as permitted under Section 6.

2.2 Modified Code. You may modify Covered Code and use, reproduce, display, perform, internally distribute within Your organization, and Externally Deploy Your Modifications and Covered Code, for commercial or non-commercial purposes, provided that in each instance You also meet all of these conditions:

(a) You must satisfy all the conditions of Section 2.1 with respect to the Source Code of the Covered Code;

(b) You must duplicate, to the extent it does not already exist, the notice in Exhibit A in each file of the Source Code of all Your Modifications, and cause the modified files to carry prominent notices stating that You changed the files and the date of any change; and

(c) If You Externally Deploy Your Modifications, You must make Source Code of all Your Externally Deployed Modifications either available to those to whom You have Externally Deployed Your Modifications, or publicly available. Source Code of Your Externally Deployed Modifications must be released under the terms set forth in this License, including the license grants set forth in Section 3 below, for as long as you Externally Deploy the Covered Code or twelve (12) months from the date of initial External Deployment, whichever is longer. You should preferably distribute the Source Code of Your Externally Deployed Modifications electronically (e.g. download from a web site).

2.3 Distribution of Executable Versions. In addition, if You Externally Deploy Covered Code (Original Code and/or Modifications) in object code, executable form only, You must include a prominent notice, in the code itself as well as in related documentation, stating that Source Code of the Covered Code is available under the terms of this License with information on how and where to obtain such Source Code.

2.4 Third Party Rights. You expressly acknowledge and agree that although Apple and each Contributor grants the licenses to their respective portions of the Covered Code set forth herein, no assurances are provided by Apple or any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Apple and each Contributor disclaim any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow You to distribute the Covered Code, it is Your responsibility to acquire that license before distributing the Covered Code.

3. Your Grants. In consideration of, and as a condition to, the licenses granted to You under this License, You hereby grant to any person or entity
receiving or distributing Covered Code under this License a non-exclusive, royalty-free, perpetual, irrevocable license, under Your Applicable Patent Rights and other intellectual property rights (other than patent) owned or controlled by You, to use, reproduce, display, perform, modify, sublicense, distribute and Externally Deploy Your Modifications of the same scope and extent as Apple's licenses under Sections 2.1 and 2.2 above.

4. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In each such instance, You must make sure the requirements of this License are fulfilled for the Covered Code or any portion thereof.

5. Limitations on Patent License. Except as expressly stated in Section 2, no other patent rights, express or implied, are granted by Apple herein. Modifications and/or Larger Works may require additional patent licenses from Apple which Apple may grant in its sole discretion.

6. Additional Terms. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations and/or other rights consistent with the scope of the license granted herein ("Additional Terms") to one or more recipients of Covered Code. However, You may do so only on Your own behalf and as Your sole responsibility, and not on behalf of Apple or any Contributor. You must obtain the recipient's agreement that any such Additional Terms are offered by You alone, and You hereby agree to indemnify, defend and hold Apple and every Contributor harmless for any liability incurred by or claims asserted against Apple or such Contributor by reason of any such Additional Terms.

7. Versions of the License. Apple may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Once Original Code has been published under a particular version of this License, You may continue to use it under the terms of that version. You may also choose to use such Original Code under the terms of any subsequent version of this License published by Apple. No one other than Apple has the right to modify the terms applicable to Covered Code created under this License.

8. NO WARRANTY OR SUPPORT. The Covered Code may contain in whole or in part pre-release, untested, or not fully tested works. The Covered Code may contain errors that could cause failures or loss of data, and may be incomplete or contain inaccuracies. You expressly acknowledge and agree that use of the Covered Code, or any portion thereof, is at Your sole and entire risk. THE COVERED CODE IS PROVIDED "AS IS" AND WITHOUT WARRANTY, UPDATES OR SUPPORT OF ANY KIND AND APPLE AND APPLE'S LICENSOR(S) (COLLECTIVELY REFERRED TO AS "APPLE" FOR THE PURPOSES OF SECTIONS 8 AND 9) AND ALL CONTRIBUTORS EXPRESSLY DISCLAIM ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY, OF SATISFACTORY
QUALITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY, OF QUIET ENJOYMENT, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. APPLE AND EACH CONTRIBUTOR DOES NOT WARRANT AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE COVERED CODE, THAT THE FUNCTIONS CONTAINED IN THE COVERED CODE WILL MEET YOUR REQUIREMENTS, THAT THE OPERATION OF THE COVERED CODE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE COVERED CODE WILL BE CORRECTED. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY APPLE, AN APPLE AUTHORIZED REPRESENTATIVE OR ANY CONTRIBUTOR SHALL CREATE A WARRANTY. You acknowledge that the Covered Code is not intended for use in the operation of nuclear facilities, aircraft navigation, communication systems, or air traffic control machines in which case the failure of the Covered Code could lead to death, personal injury, or severe physical or environmental damage.

9. LIMITATION OF LIABILITY. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT SHALL APPLE OR ANY CONTRIBUTOR BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE OR YOUR USE OR INABILITY TO USE THE COVERED CODE, OR ANY PORTION THEREOF, WHETHER UNDER A THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCTS LIABILITY OR OTHERWISE, EVEN IF APPLE OR SUCH CONTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY REMEDY. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OF LIABILITY OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION MAY NOT APPLY TO YOU. In no event shall Apple's total liability to You for all damages (other than as may be required by applicable law) under this License exceed the amount of fifty dollars ($50.00).

10. Trademarks. This License does not grant any rights to use the trademarks or trade names "Apple", "Mac", "Mac OS", "QuickTime", "QuickTime Streaming Server" or any other trademarks, service marks, logos or trade names belonging to Apple (collectively "Apple Marks") or to any trademark, service mark, logo or trade name belonging to any Contributor. You agree not to use any Apple Marks in or as part of the name of products derived from the Original Code or to endorse or promote products derived from the Original Code other than as expressly permitted by and in strict compliance at all times with Apple's third party trademark usage guidelines which are posted at http://www.apple.com/legal/guidelinesfor3rdparties.html.

11. Ownership. Subject to the licenses granted under this License, each Contributor retains all rights, title and interest in and to any Modifications made by such Contributor. Apple retains all rights, title and interest in and to the Original Code and any Modifications made by or on behalf of Apple ("Apple Modifications"), and such Apple Modifications will not be automatically subject to this License. Apple may, at its sole discretion, choose to license such Apple Modifications under this License, or on different terms from those contained in this License or may choose not to license them at all.

12. Termination.
12.1 Termination. This License and the rights granted hereunder will terminate:

(a) automatically without notice from Apple if You fail to comply with any term(s) of this License and fail to cure such breach within 30 days of becoming aware of such breach; (b) immediately in the event of the circumstances described in Section 13.5(b); or (c) automatically without notice from Apple if You, at any time during the term of this License, commence an action for patent infringement against Apple; provided that Apple did not first commence an action for patent infringement against You in that instance.

12.2 Effect of Termination. Upon termination, You agree to immediately stop any further use, reproduction, modification, sublicensing and distribution of the Covered Code. All sublicenses to the Covered Code which have been properly granted prior to termination shall survive any termination of this License. Provisions which, by their nature, should remain in effect beyond the termination of this License shall survive, including but not limited to Sections 3, 5, 8, 9, 10, 11, 12.2 and 13. No party will be liable to any other for compensation, indemnity or damages of any sort solely as a result of terminating this License in accordance with its terms, and termination of this License will be without prejudice to any other right or remedy of any party.


13.1 Government End Users. The Covered Code is a "commercial item" as defined in FAR 2.101. Government software and technical data rights in the Covered Code include only those rights customarily provided to the public as defined in this License. This customary commercial license in technical data and software is provided in accordance with FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for Department of Defense purchases, DFAR 252.227-7015 (Technical Data -- Commercial Items) and 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation). Accordingly, all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

13.2 Relationship of Parties. This License will not be construed as creating an agency, partnership, joint venture or any other form of legal association between or among You, Apple or any Contributor, and You will not represent to the contrary, whether expressly, by implication, appearance or otherwise.

13.3 Independent Development. Nothing in this License will impair Apple's right to acquire, license, develop, have others develop for it, market and/or distribute technology or products that perform the same or similar functions as, or otherwise compete with, Modifications, Larger Works, technology or products that You may develop, produce, market or distribute.

13.4 Waiver; Construction. Failure by Apple or any Contributor to enforce any provision of this License will not be deemed a waiver of future enforcement of that or any other provision. Any law or regulation which provides that the
language of a contract shall be construed against the drafter will not apply to this License.

13.5 Severability. (a) If for any reason a court of competent jurisdiction finds any provision of this License, or portion thereof, to be unenforceable, that provision of the License will be enforced to the maximum extent permissible so as to effect the economic benefits and intent of the parties, and the remainder of this License will continue in full force and effect. (b) Notwithstanding the foregoing, if applicable law prohibits or restricts You from fully and/or specifically complying with Sections 2 and/or 3 or prevents the enforceability of either of those Sections, this License will immediately terminate and You must immediately discontinue any use of the Covered Code and destroy all copies of it that are in your possession or control.

13.6 Dispute Resolution. Any litigation or other dispute resolution between You and Apple relating to this License shall take place in the Northern District of California, and You and Apple hereby consent to the personal jurisdiction of, and venue in, the state and federal courts within that District with respect to this License. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded.

13.7 Entire Agreement; Governing Law. This License constitutes the entire agreement between the parties with respect to the subject matter hereof. This License shall be governed by the laws of the United States and the State of California, except that body of California law concerning conflicts of law.

Where You are located in the province of Quebec, Canada, the following clause applies: The parties hereby confirm that they have requested that this License and all related documents be drafted in English. Les parties ont exig que le present contrat et tous les documents connexes soient rdigs en anglais.

EXHIBIT A.

"Portions Copyright (c) 1999-2007 Apple Inc. All Rights Reserved.

This file contains Original Code and/or Modifications of Original Code as defined in and that are subject to the Apple Public Source License Version 2.0 (the 'License'). You may not use this file except in compliance with the License. Please obtain a copy of the License at http://www.opensource.apple.com/apsl/ and read it before using this file.

The Original Code and all software distributed under the License are distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES, INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT. Please see the License for the specific language governing rights and limitations under the License."
Copyright 2015 The Android Open Source Project, Inc.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2014-2017 Oculus VR, LLC. All rights reserved.
The MIT License (MIT)

Copyright (c) 2013 Greg Thornton

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// Copyright (c) 2006, Google Inc.
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
Copyright (C) 2009 Tomer Filiba, 2010-2011 Corbin Simpson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015-present, Brian Woodward.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012 Niklas von Hertzen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
Except as contained in this notice, the names of the authors or
their institutions shall not be used in advertising or otherwise to
promote the sale, use or other dealings in this Software without
prior written authorization from the authors.

The SIL OPEN FONT LICENSE applies to the following files:
Gelasio-Bold.ttf
Gelasio-BoldItalic.ttf
Gelasio-Italic.ttf
Gelasio-Regular.ttf
Lohit-Devanagari.ttf
Lohit-Gurmukhi.ttf
Lohit-Tamil.ttf

Copyright 2011-13 Lohit Fonts Project contributors
<http://fedorahosted.org/lohit>

This Font Software is licensed under the SIL Open Font License, Version 1.1.
This license is copied below, and is also available with a FAQ at:
http://scripts.sil.org/OFL

SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007
__________________________________________________________

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide
development of collaborative font projects, to support the font creation
efforts of academic and linguistic communities, and to provide a free and
open framework in which fonts may be shared and improved in partnership
with others.

The OFL allows the licensed fonts to be used, studied, modified and
redistributed freely as long as they are not sold by themselves. The
fonts, including any derivative works, can be bundled, embedded,
redistributed and/or sold with any software provided that any reserved
names are not used by derivative works. The fonts and derivatives,
however, cannot be released under any other type of license. The
requirement for fonts to remain under this license does not apply
to any document created using the fonts or their derivatives.

DEFINITIONS
"Font Software" refers to the set of files released by the Copyright
Holder(s) under this license and clearly marked as such. This may
include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS
Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.
TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

The GPL v2 license applies to the following files
Garuda.ttf
MuktiNarrow.ttf

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.
4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy  <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

The Vera Bitstream License applies to the following files:
DejaVuSans-Bold.ttf
DejaVuSans.ttf

Fonts are (c) Bitstream (see below). DejaVu changes are in public domain.
Glyphs imported from Arev fonts are (c) Tavmjong Bah (see below)

Bitstream Vera Fonts Copyright
-----------------------------
Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is
a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of the fonts accompanying this license ("Fonts") and associated
documentation files (the "Font Software"), to reproduce and distribute the
Font Software, including without limitation the rights to use, copy, merge,
publish, distribute, and/or sell copies of the Font Software, and to permit
persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

Arev Fonts Copyright

----------------------

Copyright (c) 2006 by Tavmjong Bah. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the modifications to the Bitstream Vera Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to
the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Tavmjong Bah" or the word "Arev".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Tavmjong Bah Arev" names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL TAVMJONG BAH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the name of Tavmjong Bah shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from Tavmjong Bah. For further information, contact: tavmjong @ free . fr.

TeX Gyre DJV Math

-------------
Fonts are (c) Bitstream (see below). DejaVu changes are in public domain.

Math extensions done by B. Jackowski, P. Strzelczyk and P. Pianowski (on behalf of TeX users groups) are in public domain.

Letters imported from Euler Fraktur from AMSfonts are (c) American Mathematical Society (see below).

Bitstream Vera Fonts Copyright
Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.
Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license (Fonts) and associated documentation files (the Font Software), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words Bitstream or the word Vera.

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the Bitstream Vera names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of GNOME, the GNOME Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the GNOME Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.
AMSFonts (v. 2.2) copyright

The PostScript Type 1 implementation of the AMS Fonts produced by and previously distributed by Blue Sky Research and Y&Y, Inc. are now freely available for general use. This has been accomplished through the cooperation of a consortium of scientific publishers with Blue Sky Research and Y&Y. Members of this consortium include:

Elsevier Science IBM Corporation Society for Industrial and Applied Mathematics (SIAM) Springer-Verlag American Mathematical Society (AMS)

In order to assure the authenticity of these fonts, copyright will be held by the American Mathematical Society. This is not meant to restrict in any way the legitimate use of the fonts, such as (but not limited to) electronic distribution of documents containing these fonts, inclusion of these fonts into other public domain or commercial font collections or computer applications, use of the outline data to create derivative fonts and/or faces, etc. However, the AMS does require that the AMS copyright notice be removed from any derivative versions of the fonts which have been altered in any way. In addition, to ensure the fidelity of TeX documents using Computer Modern fonts, Professor Donald Knuth, creator of the Computer Modern faces, has requested that any alterations which yield different font metrics be given a different name.

$Id$
--------------------------------------------------------------------------------

The SIL OPEN FONT LICENSE for Noto applies to the following files:
NotoSansKhmer-Regular.ttf

This Font Software is licensed under the SIL Open Font License, Version 1.1.

This license is copied below, and is also available with a FAQ at:
http://scripts.sil.org/OFL

-----------------------------------------------------------
SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007
-----------------------------------------------------------

PREAMBLE
The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.
The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting -- in part or in whole -- any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.

2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.

3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the
primary font name as presented to the users.

4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.

5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION
This license becomes null and void if any of the above conditions are not met.

DISCLAIMER
THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

The Apache License applies to the following files
Arimo-Bold.ttf
Arimo-BoldItalic.ttf
Arimo-Italic.ttf
Arimo-Regular.ttf
Cousine-Bold.ttf
Cousine-BoldItalic.ttf
Cousine-Italic.ttf
Cousine-Regular.ttf
Tinos-Bold.ttf
Tinos-BoldItalic.ttf
Tinos-Italic.ttf
Tinos-Regular.ttf

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The public domain/Creative Commons Zero license applies to the following files
Ahem.ttf

The Ahem font in this directory belongs to the public domain. In jurisdictions that do not recognize public domain ownership of these files, the following Creative Commons Zero declaration applies:

<http://labs.creativecommons.org/licenses/zero-waive/1.0/us/legalcode>

which is quoted below:

The person who has associated a work with this document (the "Work") affirms that he or she (the "Affirmer") is the/an author or owner of the Work. The Work may be any work of authorship, including a database.

The Affirmer hereby fully, permanently and irrevocably waives and relinquishes all of her or his copyright and related or neighboring legal rights in the Work available under any federal or state law, treaty or contract, including but not limited to moral rights, publicity and privacy rights, rights protecting against unfair competition and any rights protecting the extraction, dissemination and reuse of data, whether such rights are present or future, vested or contingent (the "Waiver"). The Affirmer makes the Waiver for the benefit of the public at large and to the detriment of the Affirmer's heirs or successors.

The Affirmer understands and intends that the Waiver has the effect of eliminating and entirely removing from the Affirmer's control all the copyright and related or neighboring legal rights previously held by the Affirmer in the Work, to that extent making the Work freely available to the public for any and all uses and purposes without restriction of any kind, including commercial use and uses in media and formats or by methods that have not yet been invented or conceived. Should the Waiver for any reason be judged legally ineffective in any jurisdiction, the Affirmer hereby grants a free, full, permanent, irrevocable, nonexclusive and worldwide license for all her or his copyright and related or neighboring legal rights in the Work.
The public domain license applies to the following files
GardinerModBug.ttf
GardinerModCat.ttf

In lieu of a licence Fonts in this site are offered free for any use; they may be installed, embedded, opened, edited, modified, regenerated, posted, packaged and redistributed. George Douros

QT LICENSE AGREEMENT
Agreement version 4.2.1

This Qt License Agreement ("Agreement") is a legal agreement for the licensing of Licensed Software (as defined below) between The Qt Company (as defined below) and the Licensee who has accepted the terms of this Agreement by downloading or using the Licensed Software and/or as defined herein:

Capitalized terms used herein are defined in Section 1.

WHEREAS:
(A) Licensee wishes to use the Licensed Software for the purpose of developing and distributing Applications and/or Devices (each as defined below); and
(B) The Qt Company is willing to grant the Licensee a right to use Licensed Software for such a purpose pursuant to term and conditions of this Agreement.

NOW, THEREFORE, THE PARTIES HEREBY AGREE AS FOLLOWS:

1. DEFINITIONS
"Affiliate" of a Party shall mean an entity (i) which is directly or indirectly controlling such Party; (ii) which is under the same direct or indirect ownership or control as such Party; or (iii) which is directly or indirectly owned or controlled by such Party. For these purposes, an entity shall be treated as being controlled by another if that other entity has fifty percent (50 %) or more of the votes in such entity, is able to direct its affairs and/or to control the composition of its board of directors or equivalent body.

"Add-on Products" shall mean The Qt Company's specific add-on software products (for example Qt Safe Renderer, Qt for Automation, Qt Application Manager), which are not licensed as part of The Qt Company's standard offering, but shall be included into the scope of Licensed Software only if so specifically agreed between the Parties.

"Applications" shall mean Licensee's software products created using the Licensed Software, which may include the Redistributables, or part thereof.

"Contractor(s)" shall mean third party consultants, distributors and
contractors performing services to the Licensee under applicable contractual arrangement.

"Customer(s)" shall mean Licensee's end users to whom Licensee, directly or indirectly, distributes copies of the Redistributables.

"Data Protection Legislation" shall mean the General Data Protection Regulation (EU 2016/679) (GDPR) and any national implementing laws, regulations and secondary legislation, as may be amended or updated from time to time, as well as any other data protection laws or regulations applicable in relevant territory.

"Deployment Platforms" shall mean operating systems and/or hardware specified in the License Certificate, on which the Redistributables can be distributed pursuant to the terms and conditions of this Agreement.

"Designated User(s)" shall mean the employee(s) of Licensee or Licensee's Affiliates acting within the scope of their employment or Licensee's Contractors acting within the scope of their services for Licensee and on behalf of Licensee. Designated Users shall be named in the License Certificate.

"Development License" shall mean the license needed by the Licensee for each Designated User to use the Licensed Software under the license grant described in Section 3.1 of this Agreement. Development Licenses are available separately for Qt for Application Development and Qt for Device Creation products, each product having its designated scope and purpose of use. Distribution Licenses are always connected to Qt for Device Creation product only.

"Development Platforms" shall mean those operating systems specified in the License Certificate, in which the Licensed Software can be used under the Development License, but not distributed in any form or used for any other purpose.

"Devices" shall mean hardware devices or products that 1) are manufactured and/or distributed by the Licensee or its Affiliates or Contractors, and (2)(i) incorporate or integrate the Redistributables or parts thereof; or (ii) where the main user interface or substantial functionality of such unit, when used by a Customer, is provided by Application(s) or otherwise depends on the Licensed Software, regardless of whether the Redistributables are distributed together with the hardware or not. Devices covered with this Agreement shall be specified in Appendix 2 or in a quote.

"Distribution License(s)" shall mean the license required for any kind of sale, trade, exchange, loan, lease, rental or other distribution by or on behalf of Licensee to a third party of Redistributables in connection with Devices pursuant to license grant described in Section 3.3 of this Agreement.
"Distribution License Packs" shall mean set of prepaid Distribution Licenses for distribution of Redistributables, as defined in The Qt Company's standard price list, quote, Purchase Order confirmation or in an appendix hereto, as the case may be.

"Intellectual Property Rights" shall mean patents (including utility models), design patents, and designs (whether or not capable of registration), chip topography rights and other like protection, copyrights, trademarks, service marks, trade names, logos or other words or symbols and any other form of statutory protection of any kind and applications for any of the foregoing as well as any trade secrets.

"License Certificate" shall mean a certificate generated by The Qt Company for each Designated User respectively upon them downloading the Licensed Software, which will be available under respective Designated User's Qt Account at account.qt.io. License Certificates will specify the Designated User, the Development Platforms, Deployment Platforms and the License Term. Such terms are considered part of the licenses granted hereunder and shall be updated from time to time to reflect any agreed changes to the foregoing terms relating to Designated User's rights to theLicensed Software.

"License Fee" shall mean the fee charged to the Licensee for rights granted under the terms of this Agreement.

"License Term" shall mean the agreed validity period of the Development License of the respective Designated User, during which time the Designated User is entitled to use the Licensed Software, as set forth in the respective License Certificate.

"Licensed Software" shall mean either
(i) Qt for Application Development or
(ii) Qt for Device Creation, and/or
(iii) Qt 3D Studio, and/or
(iv) Qt Design Studio, and/or
(v) Qt for MCUs, and/or
(vi) selected Add-on Products, if any,

depending on which product(s) the Licensee has purchased under this Agreement, as well as corresponding online or electronic documentation, associated media and printed materials, including the source code (where applicable), example programs and the documentation, licensed to the Licensee under this Agreement. Licensed Software does not include Third Party Software (as defined in Section 4) or Open Source Qt. The Qt Company may, in the course of its development activities, at its free and absolute discretion and without any obligation to send or publish any notifications to the Licensee or in general, make changes, additions or deletions in the components and functionalities of the Licensed Software, provided that no such changes, additions or deletions will affect the already released version of the Licensed Software, but only upcoming
"Licensee" shall mean the individual or legal entity that is party to this Agreement, as identified on the signature page hereof.

"Licensee's Records" shall mean books and records that are likely to contain information bearing on Licensee's compliance with this Agreement, Licensee's use of Open Source Qt and/or the payments due to The Qt Company under this Agreement, including, but not limited to user information, assembly logs, sales records and distribution records.

"Modified Software" shall have the meaning as set forth in Section 2.3.

"Online Services" shall mean any services or access to systems made available by The Qt Company to the Licensee over the Internet relating to the Licensed Software or for the purpose of use by the Licensee of the Licensed Software or Support. Use of any such Online Services is discretionary for the Licensee and some of them may be subject to additional fees.

"Open Source Qt" shall mean the non-commercial Qt computer software products, licensed under the terms of the GNU Lesser General Public License, version 2.1 or later ("LGPL") or the GNU General Public License, version 2.0 or later ("GPL"). For clarity, Open Source Qt shall not be provided nor governed under this Agreement.

"Party" or "Parties" shall mean Licensee and/or The Qt Company.

"Permitted Combination" shall have the meaning as set forth in Section 3.4(viii).

"Pre-Release Code" shall have the meaning as set forth in Section 4.

"Prohibited Combination" shall mean any means to (i) use, combine, incorporate, link or integrate Licensed Software with any software created with or incorporating Open Source Qt, (ii) use Licensed Software for creation of any software created with or incorporating Open Source Qt, or (iii) incorporate or integrate Applications into a hardware device or product other than a Device.

"Qt 3D Studio" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.

"Qt Design Studio" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.

"Qt for Application Development" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.
"Qt for Device Creation" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.

"Qt for MCUs" shall mean The Qt Company's productized offering, which consist of all versions of modules and tools as set forth in Appendix 1.

"Redistributables" shall mean the portions of the Licensed Software set forth in Appendix 1 that may be distributed pursuant to the terms of this Agreement in object code form only, including any relevant documentation. Where relevant, any reference to Licensed Software in this Agreement shall include and refer also to Redistributables.

"Renewal Term" shall mean an extension of previous License Term as agreed between the Parties.

"Submitted Modified Software" shall have the meaning as set forth in Section 2.3.

"Support" shall mean standard developer support that is provided by The Qt Company to assist Designated Users in using the Licensed Software in accordance with The Qt Company's standard support terms available at https://www.qt.io/terms-conditions/ and as further defined in Section 8 hereunder.

"Taxes" shall have the meaning set forth in Section 10.5.

"Term" shall have the meaning set forth in Section 12.

"The Qt Company" shall mean:
(i) in the event Licensee is an individual residing in the United States or a legal entity incorporated in the United States or having its headquarters in the United States, The Qt Company Inc., a Delaware corporation with its office at 2350 Mission College Blvd., Suite 1020, Santa Clara, CA 95054, USA.; or
(ii) in the event the Licensee is an individual residing outside of the United States or a legal entity incorporated outside of the United States or having its registered office outside of the United States, The Qt Company Ltd., a Finnish company with its registered office at Bertel Jungin aukio D3A, 02600 Espoo, Finland.

"Third-Party Software" shall have the meaning set forth in Section 4.

"Updates" shall mean a release or version of the Licensed Software containing bug fixes, error corrections and other changes that are generally made available to users of the Licensed Software that have contracted for Support. Updates are generally depicted as a change to the digits following the decimal in the Licensed Software version number. The Qt Company shall make Updates available to the Licensee under the Support. Updates shall be considered as
part of the Licensed Software hereunder.

"Upgrades" shall mean a release or version of the Licensed Software containing enhancements and new features and are generally depicted as a change to the first digit of the Licensed Software version number. In the event Upgrades are provided to the Licensee under this Agreement, they shall be considered as part of the Licensed Software hereunder.

2. OWNERSHIP
2.1. Ownership of The Qt Company
The Licensed Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Licensed Software is licensed, not sold.

All of The Qt Company's Intellectual Property Rights are and shall remain the exclusive property of The Qt Company or its licensors respectively.

2.2. Ownership of Licensee
All the Licensee's Intellectual Property Rights are and shall remain the exclusive property of the Licensee or its licensors respectively.

All Intellectual Property Rights to the Modified Software, Applications and Devices shall remain with the Licensee and no rights thereto shall be granted by the Licensee to The Qt Company under this Agreement (except as set forth in Section 2.3 below).

2.3. Modified Software
Licensee may create bug-fixes, error corrections, patches or modifications to the Licensed Software ("Modified Software"). Such Modified Software may break the source or binary compatibility with the Licensed Software (including without limitation through changing the application programming interfaces ("API") or by adding, changing or deleting any variable, method, or class signature in the Licensed Software and/or any inter-process protocols, services or standards in the Licensed Software libraries). To the extent that Licensee's Modified Software so breaks source or binary compatibility with the Licensed Software, Licensee acknowledges that The Qt Company's ability to provide Support may be prevented or limited and Licensee's ability to make use of Updates may be restricted.

Licensee may, at its sole and absolute discretion, choose to submit Modified Software to The Qt Company ("Submitted Modified Software") in connection with Licensee's Support request, service request or otherwise. In the event Licensee does so, then, Licensee hereby grants The Qt Company a sublicensable, assignable, irrevocable, perpetual, worldwide, non-exclusive, royalty-free and fully paid-up license, under all of Licensee's Intellectual Property Rights, to reproduce, adapt, translate, modify, and prepare derivative works of, publicly display, publicly perform, sublicense, make available and distribute
such Submitted Modified Software as The Qt Company sees fit at its free and absolute discretion.

3. LICENSES GRANTED
3.1. Development with Licensed Software
Subject to the terms of this Agreement, The Qt Company grants to Licensee a worldwide, non-exclusive, non-transferable license, valid for the License Term, to use, modify and copy the Licensed Software by Designated Users on the Development Platforms for the sole purposes of designing, developing, demonstrating and testing Application(s) and/or Devices, and to provide thereto related support and other related services to end-user Customers.

Licensee may install copies of the Licensed Software on five (5) computers per Designated User, provided that only the Designated Users who have a valid Development License may use the Licensed Software.

Licensee may at any time designate another Designated User to replace a then-current Designated User by notifying The Qt Company in writing, provided that any Designated User may be replaced only once during any six-month period.

Upon expiry of the initially agreed License Term, the respective License Terms shall be automatically extended to one or more Renewal Term(s), unless and until either Party notifies the other Party in writing that it does not wish to continue the License Term, such notification to be provided to the other Party no less than ninety (90) days before expiry of the respective License Term. Unless otherwise agreed between the Parties, Renewal Term shall be of equal length with the initial License Term.

Any such Renewal Term shall be subject to License Fees agreed between the Parties or, if no advance agreement exists, subject to The Qt Company’s standard pricing applicable at the commencement date of any such Renewal Term.

Any price or other term specified for a Renewal Term shall be valid only for the specified time.

The Qt Company may request the Licensee to place a purchase order corresponding to a quote by The Qt Company for the relevant Renewal Term.

In the event Licensee does not prevent auto-renewal pursuant the above, but a Renewal Term is nevertheless not duly ordered within 30 days from the date of the respective quote from The Qt Company and/or the respective License Fee paid by due date specified in The Qt Company’s respective invoice, The Qt Company shall apply a reinstatement fee equal to ten percent (10 %) of the total value of the License Fees of the Development Licenses for the expired term to be added to the License Fee of the respective Renewal Term.

In the event Licensee chooses not to renew a Development License for a Renewal Term by notifying The Qt Company thereof no less than ninety (90) days before
expiry of the respective License Term, Licensee may still reinstate such expired Development Licenses for a Renewal Term subject to applicable renewal Term License Fees until thirty (30) days from the expiry of the initially agreed License Term or preceding Renewal Term. After such thirty (30) day period a Development License shall be subject to applicable License Fees for a new Development License and not any Renewal Term License Fees.

3.2. Distribution of Applications
Subject to the terms of this Agreement, The Qt Company grants to Licensee a worldwide, non-exclusive, non-transferable, revocable (for cause pursuant to this Agreement) right and license, valid for the Term, to
   (i) distribute, by itself or through its Contractors, Redistributables as installed, incorporated or integrated into Applications for execution on the Deployment Platforms, and
   (ii) grant sublicenses to Redistributables, as distributed hereunder, for Customers solely for Customer's internal use and to the extent necessary in order for the Customers to use the Applications for their respective intended purposes.

Right to distribute the Redistributables as part of an Application as provided herein is not royalty-bearing but is conditional upon the Licensee not having any unpaid License Fees for Development Licenses owed to The Qt Company at the time of distribution of any Redistributables to Customers.

3.3. Distribution of Devices
Subject to the terms of this Agreement, The Qt Company grants to Licensee a worldwide, non-exclusive, non-transferable, revocable (for cause pursuant to this Agreement) right and license, valid for the Term, to
   (i) distribute, by itself or through one or more tiers of Contractors, Redistributables as installed, incorporated or integrated, or intended to be installed, incorporated or integrated into Devices for execution on the Deployment Platforms, and
   (ii) grant sublicenses to Redistributables, as distributed hereunder, for Customers solely for Customer's internal use and to the extent necessary in order for the Customers to use the Devices for their respective intended purposes.

Right to distribute the Redistributables with Devices as provided herein is conditional upon the Licensee (i) not having any unpaid License Fees for Development Licenses owed to The Qt Company, and (ii) having purchased and paid corresponding Distribution Licenses at the time of distribution of any Redistributables to Customers.

3.4. Further Requirements
The licenses granted above in this Section 3 by The Qt Company to Licensee are conditional and subject to Licensee's compliance with the following terms:
   (i) Licensee shall not remove or alter any copyright, trademark or other proprietary rights notice(s) contained in any portion of the
Licensed Software;

(ii) Applications must add primary and substantial functionality to the Licensed Software so as not to compete with the Licensed Software;

(iii) Applications may not pass on functionality which in any way makes it possible for others to create software with the Licensed Software; provided however that Licensee may use the Licensed Software's scripting and QML ("Qt Quick") functionality solely in order to enable scripting, themes and styles that augment the functionality and appearance of the Application(s) without adding primary and substantial functionality to the Application(s);

(iv) Licensee shall not use Licensed Software in any manner or for any purpose that infringes, misappropriates or otherwise violates any Intellectual property or right of any third party, or that violates any applicable law;

(v) Licensee shall not use The Qt Company's or any of its suppliers' names, logos, or trademarks to market Applications, except that Licensee may use "Built with Qt" logo to indicate that Application(s) was developed using the Licensed Software;

(vi) Licensee shall not distribute, sublicense or disclose source code of Licensed Software to any third party (provided however that Licensee may appoint employee(s) of Contractors as Designated Users to use Licensed Software pursuant to this Agreement). Such right may be available for the Licensee subject to a separate software development kit ("SDK") license agreement to be concluded with The Qt Company;

(vii) Licensee shall not grant the Customers a right to (i) make copies of the Redistributables except when and to the extent required to use the Applications and/or Devices for their intended purpose, (ii) modify the Redistributables or create derivative works thereof, (iii) decompile, disassemble or otherwise reverse engineer Redistributables, or (iv) redistribute any copy or portion of the Redistributables to any third party, except as part of the onward sale of the Device on which the Redistributables are installed;

(viii) Licensee shall not and shall cause that its Affiliates or Contractors shall not use Licensed Software in any Prohibited Combination, unless Licensee has received an advance written permission from The Qt Company to do so. Absent such written permission, any and all distribution by the Licensee during the Term of a hardware device or product a) which incorporate or integrate any part of Licensed Software or Open Source Qt; or b) where the main user interface or substantial functionality is provided by software built with Licensed Software or Open Source Qt or otherwise depends on the Licensed Software or Open Source Qt, shall be considered to be Device distribution under this Agreement and shall be dependent on Licensee's compliance thereof (including but not limited to obligation to pay applicable License Fees for such distribution). Notwithstanding what is provided
above in this sub-section (viii), Licensee is entitled to use and combine Qt 3D Studio and/or Qt Design Studio with Open Source Qt ("Permitted Combination") for its internal evaluation purposes, provided that Licensee shall in no way transfer, publish, disclose, display or otherwise make available any software or work resulting from such Permitted Combination;

(ix) Licensee shall cause all of its Affiliates and Contractors entitled to make use of the licenses granted under this Agreement, to be contractually bound to comply with the relevant terms of this Agreement and not to use the Licensed Software beyond the terms hereof and for any purposes other than operating within the scope of their services for Licensee. Licensee shall be responsible for any and all actions and omissions of its Affiliates and Contractors relating to the Licensed Software and use thereof (including but not limited to payment of all applicable License Fees);

(x) Except when and to the extent explicitly provided in this Section 3, Licensee shall not transfer, publish, disclose, display or otherwise make available the Licensed Software; and

(xi) Licensee shall not attempt or enlist a third party to conduct or attempt to conduct any of the above.

Above terms shall not be applicable if and to the extent they conflict with any mandatory provisions of any applicable laws.
Any use of Licensed Software beyond the provisions of this Agreement is strictly prohibited and requires an additional license from The Qt Company.

4. THIRD-PARTY SOFTWARE
The Licensed Software may provide links to third party libraries or code (collectively "Third-Party Software") to implement various functions. Third-Party Software does not comprise part of the Licensed Software. In some cases, access to Third-Party Software may be included with the Licensed Software. Such Third-Party Software will be listed in the "./src/3rdparty" source tree delivered with the Licensed Software or documented in the Licensed Software, as such may be amended from time to time. Licensee acknowledges that use or distribution of Third-Party Software is in all respects subject to applicable license terms of applicable third-party right holders.

5. PRE-RELEASE CODE
The Licensed Software may contain pre-release code and functionality, or sample code marked or otherwise stated with appropriate designation such as "Technology Preview", "Alpha", "Beta", "Sample" etc. ("Pre-Release Code").

Such Pre-Release Code may be present complimentary for the Licensee, in order to provide experimental support or information for new platforms or preliminary versions of one or more new functionalities or for other similar reasons. The Pre-Release Code may not be at the level of performance and compatibility of a final, generally available, product offering. The Pre-Release Code may not
operate correctly, may contain errors and may be substantially modified by The Qt Company prior to the first commercial product release, if any. The Qt Company is under no obligation to make Pre-Release Code commercially available, or provide any Support or Updates relating thereto. The Qt Company assumes no liability whatsoever regarding any Pre-Release Code, but any use thereof is exclusively at Licensee's own risk and expense.

For clarity, unless Licensed Software specifies different license terms for the respective Pre-Release Code, the Licensee is entitled to use such pre-release code pursuant to Section 3, just like other Licensed Software, provided however that in the event Add-on Products are included and available as such Pre-Release Code, Licensee's right to use such Add-on Products is nevertheless subject to and conditional upon conclusion of separate agreement with The Qt Company.

6. LIMITED WARRANTY AND WARRANTY DISCLAIMER
The Qt Company hereby represents and warrants that it has the power and authority to grant the rights and licenses granted to Licensee under this Agreement.

Except as set forth above, the Licensed Software is licensed to Licensee "as is" and Licensee's exclusive remedy and The Qt Company's entire liability for errors in the Licensed Software shall be limited, at The Qt Company's option, to correction of the error, replacement of the Licensed Software or return of the applicable fees paid for the defective Licensed Software for the time period during which the License is not able to utilize the Licensed Software under the terms of this Agreement.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE QT COMPANY ON BEHALF OF ITSELF AND ITS LICENSORS, SUPPLIERS AND AFFILIATES, DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT WITH REGARD TO THE LICENSED SOFTWARE. THE QT COMPANY DOES NOT WARRANT THAT THE LICENSED SOFTWARE WILL SATISFY LICENSEE'S REQUIREMENTS OR THAT IT WILL OPERATE WITHOUT DEFECT OR ERROR OR THAT THE OPERATION THEREOF WILL BE UNINTERRUPTED.

7. INDEMNIFICATION AND LIMITATION OF LIABILITY
7.1. Limitation of Liability
EXCEPT FOR (I) CASES OF GROSS NEGLIGENCE OR INTENTIONAL MISCONDUCT, AND (II) BREACH OF CONFIDENTIALITY, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL EITHER PARTY BE LIABLE TO THE OTHER PARTY FOR ANY LOSS OF PROFIT, LOSS OF DATA, LOSS OF BUSINESS OR GOODWILL OR ANY OTHER INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE COST, DAMAGES OR EXPENSE OF ANY KIND, HOWSOEVER ARISING UNDER OR IN CONNECTION WITH THIS AGREEMENT.

EXCEPT FOR (I) CASES OF GROSS NEGLIGENCE OR INTENTIONAL MISCONDUCT, AND (II) BREACH OF CONFIDENTIALITY, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, IN NO
EVENT SHALL EITHER PARTY'S TOTAL AGGREGATE LIABILITY UNDER THIS AGREEMENT EXCEED THE AGGREGATE LICENSE FEES PAID OR PAYABLE TO THE QT COMPANY FROM LICENSEE DURING THE PERIOD OF TWELVE (12) MONTHS IMMEDIATELY PRECEDING THE EVENT RESULTING IN SUCH LIABILITY.

THE PROVISIONS OF THIS SECTION 7 ALLOCATE THE RISKS UNDER THIS AGREEMENT BETWEEN THE QT COMPANY AND LICENSEE AND THE PARTIES HAVE RELIED UPON THE LIMITATIONS SET FORTH HEREIN IN DETERMINING WHETHER TO ENTER INTO THIS AGREEMENT.

NOTWITHSTANDING ANYTHING TO THE CONTRARY IN THIS AGREEMENT, LICENSEE SHALL ALWAYS BE LIABLE TO PAY THE APPLICABLE LICENSE FEES CORRESPONDING TO ITS ACTUAL USE OF LICENSED SOFTWARE.

8. SUPPORT, UPDATES AND ONLINE SERVICES

Upon due payment of the agreed License Fees the Licensee will be eligible to receive Support and Updates and to use the Online Services during the License Term, provided, however, that in the event the License Term is longer than 36 months, the initial payment includes Support for only the first 12 months, unless the Parties specifically otherwise agree.

Unless otherwise decided by The Company at its free and absolute discretion, Upgrades will not be included in the Support but may be available subject to additional fees.

From time to time The Qt Company may change the Support terms, provided that during the respective ongoing License Term the level of Support provided by The Qt Company may not be reduced without the consent of the Licensee.

Unless otherwise agreed, The Qt Company shall not be responsible for providing any service or support to Customers.

9. CONFIDENTIALITY

Each Party acknowledges that during the Term of this Agreement each Party may receive information about the other Party's business, business methods, business plans, customers, business relations, technology, and other information, including the terms of this Agreement, that is confidential and of great value to the other Party, and the value of which would be significantly reduced if disclosed to third parties ("Confidential Information"). Accordingly, when a Party (the "Receiving Party") receives Confidential Information from the other Party (the "Disclosing Party"), the Receiving Party shall only disclose such information to employees and Contractors on a need to know basis, and shall cause its employees and employees of its Affiliates to: (i) maintain any and all Confidential Information in confidence; (ii) not disclose the Confidential Information to a third party without the Disclosing Party's prior written approval; and (iii) not, directly or indirectly, use the Confidential Information for any purpose other than for exercising its rights and fulfilling its responsibilities pursuant to this Agreement. Each Party
shall take reasonable measures to protect the Confidential Information of the other Party, which measures shall not be less than the measures taken by such Party to protect its own confidential and proprietary information.

Obligation of confidentiality shall not apply to information that (i) is or becomes generally known to the public through no act or omission of the Receiving Party; (ii) was in the Receiving Party’s lawful possession prior to the disclosure hereunder and was not subject to limitations on disclosure or use; (iii) is developed independently by employees or Contractors of the Receiving Party or other persons working for the Receiving Party who have not had access to the Confidential Information of the Disclosing Party, as proven by the written records of the Receiving Party; (iv) is lawfully disclosed to the Receiving Party without restrictions, by a third party not under an obligation of confidentiality; or (v) the Receiving Party is legally compelled to disclose, in which case the Receiving Party shall notify the Disclosing Party of such compelled disclosure and assert the privileged and confidential nature of the information and cooperate fully with the Disclosing Party to limit the scope of disclosure and the dissemination of disclosed Confidential Information to the minimum extent necessary.

The obligations under this Section 9 shall continue to remain in force for a period of five (5) years after the last disclosure, and, with respect to trade secrets, for so long as such trade secrets are protected under applicable trade secret laws.

10. FEES, DELIVERY AND PAYMENT

10.1. License Fees
License Fees are described in The Qt Company’s standard price list, quote or Purchase Order confirmation or in an appendix hereto, as the case may be.

The License Fees shall not be refunded or claimed as a credit in any event or for any reason whatsoever.

10.2. Ordering Licenses
Licensee may purchase Development Licenses and Distribution Licenses pursuant to agreed pricing terms or, if no specific pricing terms have been agreed upon, at The Qt Company’s standard pricing terms applicable at the time of purchase. Unless specifically otherwise provided, any pricing terms referenced in this Agreement shall be valid for twelve (12) months from the date of this Agreement.

Licensee shall submit all purchase orders for Development Licenses and Distribution Licenses to The Qt Company by email or any other method acceptable to The Qt Company (each such order is referred to herein as a “Purchase Order”) for confirmation, whereupon the Purchase Order shall become binding between the Parties.

10.3. Distribution License Packs
Unless otherwise agreed, Distribution Licenses shall be purchased by way of
Upon due payment of the ordered Distribution License Pack(s), the Licensee will have an account of Distribution Licenses available for distributing the Redistributables in accordance with this Agreement.

Each time Licensee distributes a copy of Redistributables, then one Distribution License is used, and Licensee's account of available Distribution Licenses is decreased accordingly. Licensee may distribute copies of the Redistributables so long as Licensee has Distribution Licenses remaining on its account.

10.4. Payment Terms
License Fees and any other charges under this Agreement shall be paid by Licensee no later than thirty (30) days from the date of the applicable invoice from The Qt Company.

The Qt Company will submit an invoice to Licensee after the date of this Agreement and/or after The Qt Company receives a Purchase Order from Licensee.

A late payment charge of the lower of (a) one percent per month; or (b) the interest rate stipulated by applicable law, shall be charged on any unpaid balances that remain past due.

10.5. Taxes
All License Fees and other charges payable hereunder are gross amounts but exclusive of any value added tax, use tax, sales tax, withholding tax and other taxes, duties or tariffs ("Taxes") levied directly for the sale, delivery or use of Licensed Software hereunder pursuant to any applicable law. Such applicable Taxes shall be paid by Licensee to The Qt Company, or, where applicable, in lieu of payment of such Taxes to The Qt Company, Licensee shall provide an exemption certificate to The Qt Company and any applicable authority.

11. RECORD-KEEPING AND REPORTING OBLIGATIONS; AUDIT RIGHTS
11.1. Licensee's Record-keeping
Licensee shall at all times during the Term of this Agreement and for a period of seven (7) years thereafter maintain Licensee's Records in an accurate and up-to-date form. Licensee's Records shall be adequate to reasonably enable The Qt Company to determine Licensee's compliance with the provisions of this Agreement. The records shall conform to general good accounting practices.

Licensee shall, within thirty (30) days from receiving The Qt Company's request to that effect, deliver to The Qt Company a report based on Licensee's Records, such report to contain information, in sufficient detail, on (i) number and identity of users working with Licensed Software or Open Source Qt, (ii) copies of Redistributables distributed by Licensee during the most recent calendar quarter and/or any other term specified by The Qt Company, (iii) number of undistributed copies of Redistributables and corresponding number of unused
Distribution Licenses remaining on Licensee's account, and (iv) any other information as The Qt Company may reasonably require from time to time.

11.2. The Qt Company's Audit Rights
The Qt Company or an independent auditor acting on behalf of The Qt Company's, may, upon at least five (5) business days' prior written notice and at its expense, audit Licensee with respect to the Licensee's use of the Licensed Software, but not more frequently than once during each 6-month period. Such audit may be conducted by mail, electronic means or through an in-person visit to Licensee's place of business. Any such in-person audit shall be conducted during regular business hours at Licensee's facilities and shall not unreasonably interfere with Licensee's business activities. The Qt Company or the independent auditor acting on behalf of The Qt Company shall be entitled to inspect Licensee's Records and conduct necessary interviews of Licensee's relevant employees and Contractors. All such Licensee's Records and use thereof shall be subject to an obligation of confidentiality under this Agreement.

If an audit reveals that Licensee is using the Licensed Software beyond scope of the licenses Licensee has paid for, Licensee agrees to pay The Qt Company any amounts owed for such unauthorized use within 30 days from receipt of the corresponding invoice from The Qt Company. In addition, in the event the audit reveals a material violation of the terms of this Agreement (without limitation, either (i) underpayment of more than 10% of License Fees or 10,000 euros (whichever is more) or (ii) distribution of products, which include or result from Prohibited Combination, shall be deemed a material violation for purposes of this section), then the Licensee shall pay The Qt Company's reasonable cost of conducting such audit.

12. TERM AND TERMINATION
12.1. Agreement Term
This Agreement shall enter into force upon due acceptance by both Parties and remain in force for as long as there is any Development License(s) purchased under this Agreement in force (“Term”), unless and until terminated pursuant to the terms of this Section 12.

12.2. Termination and suspension of rights
Either Party shall have the right to terminate this Agreement upon thirty (30) days prior written notice if the other Party commits a material breach of any obligation of this Agreement and fails to remedy such breach within such notice period.

Instead of termination, The Qt Company shall have the right to suspend or withhold grants of all rights to the Licensed Software hereunder, including but not limited to the Development Licenses, Distribution License, and Support, should Licensee fail to make payment in timely fashion or otherwise violates or is reasonably suspected to violate its obligations or terms of this Agreement, and where such violation or breach is not cured within five (5) business days following The Qt Company's written notice thereof.
12.3. Mutual Right to Terminate
Either Party shall have the right to terminate this Agreement immediately upon written notice in the event that the other Party becomes insolvent, files for any form of bankruptcy, makes any assignment for the benefit of creditors, has a receiver, administrative receiver or officer appointed over the whole or a substantial part of its assets, ceases to conduct business, or an act equivalent to any of the above occurs under the laws of the jurisdiction of the other Party.

12.4. Parties Rights and Duties upon Termination
Upon expiry or termination of the Agreement, Licensee shall cease and shall cause all Designated Users (including those of its Affiliates' and Contractors') to cease using the Licensed Software and distribution of the Redistributables under this Agreement.

Notwithstanding the above, in the event the Agreement expires or is terminated:
(i) as a result of The Qt Company choosing not to renew the Development License(s) as set forth in Section 3.1, then all valid licenses possessed by the Licensee at such date shall be extended to be valid in perpetuity under the terms of this Agreement and Licensee is entitled to purchase additional licenses as set forth in Section 10.2; or
(ii) for reason other than by The Qt Company pursuant to item (i) above or pursuant to Section 12.2, then the Licensee is entitled, for a period of six (6) months after the effective date of termination, to continue distribution of Devices under the Distribution Licenses paid but unused at such effective date of termination.

Upon any such termination the Licensee shall destroy or return to The Qt Company all copies of the Licensed Software and all related materials and will certify the same to The Qt Company upon its request, provided however that Licensee may retain and exploit such copies of the Licensed Software as it may reasonably require in providing continued support to Customers.

Expiry or termination of this Agreement for any reason whatsoever shall not relieve Licensee of its obligation to pay any License Fees accrued or payable to The Qt Company prior to the effective date of termination, and Licensee shall immediately pay to The Qt Company all such fees upon the effective date of termination. Termination of this Agreement shall not affect any rights of Customers to continue use of Applications and Devices (and therein incorporated Redistributables).

12.5. Extension in case of bankruptcy
In the event The Qt Company is declared bankrupt under a final, non-cancellable decision by relevant court of law, and this Agreement is not, at the date of expiry of the Development License(s) pursuant to Section 3.1, assigned to party, who has assumed The Qt Company's position as a legitimate licensor of
Licensed Software under this Agreement, then all valid licenses possessed by
the Licensee at such date of expiry, and which the Licensee has not notified
for expiry, shall be extended to be valid in perpetuity under the terms of this
Agreement.

13. GOVERNING LAW AND LEGAL VENUE
In the event this Agreement is in the name of The Qt Company Inc., a Delaware
Corporation, then:
(i) this Agreement shall be construed and interpreted in accordance
with the laws of the State of California, USA, excluding its choice
of law provisions;
(ii) the United Nations Convention on Contracts for the International
Sale of Goods will not apply to this Agreement; and
(iii) any dispute, claim or controversy arising out of or relating to
this Agreement or the breach, termination, enforcement,
interpretation or validity thereof, including the determination of
the scope or applicability of this Agreement to arbitrate, shall
be determined by arbitration in San Francisco, USA, before one
arbitrator. The arbitration shall be administered by JAMS pursuant
to JAMS' Streamlined Arbitration Rules and Procedures. Judgment on
the Award may be entered in any court having jurisdiction. This
Section shall not preclude parties from seeking provisional
remedies in aid of arbitration from a court of appropriate
jurisdiction.

In the event this Agreement is in the name of The Qt Company Ltd., a Finnish
Company, then:
(i) this Agreement shall be construed and interpreted in accordance
with the laws of Finland, excluding its choice of law provisions;
(ii) the United Nations Convention on Contracts for the International
Sale of Goods will not apply to this Agreement; and
(iii) any disputes, controversy or claim arising out of or relating to
this Agreement, or the breach, termination or validity thereof
shall be finally settled by arbitration in accordance with the
Arbitration Rules of Finland Chamber of Commerce. The arbitration
tribunal shall consist of one (1), or if either Party so requires,
of three (3), arbitrators. The award shall be final and binding and
enforceable in any court of competent jurisdiction. The arbitration
shall be held in Helsinki, Finland and the process shall be
conducted in the English language. This Section shall not preclude
parties from seeking provisional remedies in aid of arbitration
from a court of appropriate jurisdiction.

14. GENERAL PROVISIONS
14.1. No Assignment
Except in the case of a merger or sale of substantially all of its corporate
assets, Licensee shall not be entitled to assign or transfer all or any of its
rights, benefits and obligations under this Agreement without the prior written
consent of The Qt Company, which shall not be unreasonably withheld or delayed. The Qt Company shall be entitled to freely assign or transfer any of its rights, benefits or obligations under this Agreement.

14.2. No Third-Party Representations
Licensee shall make no representations or warranties concerning the Licensed Software on behalf of The Qt Company. Any representation or warranty Licensee makes or purports to make on The Qt Company’s behalf shall be void as to The Qt Company.

14.3. Surviving Sections
Any terms and conditions that by their nature or otherwise reasonably should survive termination of this Agreement shall so be deemed to survive. Such sections include especially the following: 1, 2, 6, 7, 9, 11, 12, 13 and 14.

14.4. Entire Agreement
This Agreement, the exhibits hereto, the License Certificate and any applicable Purchase Order accepted by The Qt Company constitute the complete agreement between the Parties and supersedes all prior or contemporaneous discussions, representations, and proposals, written or oral, with respect to the subject matters discussed herein.

In the event of any conflict or inconsistency between this Agreement and any Purchase Order, the terms of this Agreement will prevail over the terms of the Purchase Order with respect to such conflict or inconsistency.

Parties specifically acknowledge and agree that this Agreement prevails over any click-to-accept or similar agreements the Designated Users may need to accept online upon download of the Licensed Software, as may be required by The Qt Company’s applicable processes relating to Licensed Software.

14.5. Modifications
No modification of this Agreement shall be effective unless contained in a writing executed by an authorized representative of each Party. No term or condition contained in Licensee’s Purchase Order ("Deviating Terms") shall apply unless The Qt Company has expressly agreed such Deviating Terms in writing. Unless and to the extent expressly agreed by The Qt Company, any such Deviating Terms shall be deemed void and with no legal effect. For clarity, delivery of the Licensed Software following the receipt of the Purchase Order including Deviating Terms shall not constitute acceptance of such Deviating Terms."

14.6. Force Majeure
Except for the payment obligations hereunder, neither Party shall be liable to the other for any delay or non-performance of its obligations hereunder in the event and to the extent that such delay or non-performance is due to an event of act of God, terrorist attack or other similar unforeseeable catastrophic event that prevents either Party for fulfilling its obligations under this
Agreement and which such Party cannot avoid or circumvent ("Force Majeure Event"). If the Force Majeure Event results in a delay or non-performance of a Party for a period of three (3) months or longer, then either Party shall have the right to terminate this Agreement with immediate effect without any liability (except for the obligations of payment arising prior to the event of Force Majeure) towards the other Party.

14.7. Notices
Any notice given by one Party to the other shall be deemed properly given and deemed received if specifically acknowledged by the receiving Party in writing or when successfully delivered to the recipient by hand, fax, or special courier during normal business hours on a business day to the addresses specified for each Party on the signature page. Each communication and document made or delivered by one Party to the other Party pursuant to this Agreement shall be in the English language.

14.8. Export Control
Licensee acknowledges that the Redistributables, as incorporated in Applications or Devices, may be subject to export control restrictions under the applicable laws of respective countries. Licensee shall fully comply with all applicable export license restrictions and requirements as well as with all laws and regulations relating to the Redistributables and exercise of licenses hereunder and shall procure all necessary governmental authorizations, including without limitation, all necessary licenses, approvals, permissions or consents, where necessary for the re-exportation of the Redistributables, Applications and/or Devices.

14.9. No Implied License
There are no implied licenses or other implied rights granted under this Agreement, and all rights, save for those expressly granted hereunder, shall remain with The Qt Company and its licensors. In addition, no licenses or immunities are granted to the combination of the Licensed Software with any other software or hardware not delivered by The Qt Company under this Agreement.

14.10. Attorney Fees
The prevailing Party in any action to enforce this Agreement shall be entitled to recover its attorney's fees and costs in connection with such action.

14.11. Privacy
Licensee acknowledges and agrees that for the purpose of this Agreement, The Qt Company may collect, use, transfer and disclose personal data pertaining to Designated Users as well as any other employees and directors of the Licensee and its Contractors relevant for carrying out the intent of this Agreement. Such personal data may be collected from the Licensee or directly from the relevant individuals. The Parties acknowledge that with regard to such personal data processed hereunder, The Qt Company shall be regarded as the Data Controller under the applicable Data Protection Legislation. The Qt Company shall process any such personal data in accordance with its privacy policies.
and practices, which will comply with all applicable requirements of the Data Protection Legislation.

14.12. Severability
If any provision of this Agreement shall be adjudged by any court of competent jurisdiction to be unenforceable or invalid, that provision shall be limited or eliminated to the minimum extent necessary so that this Agreement shall otherwise remain in full force and effect and enforceable.

APPENDICES
The Agreement includes Appendix 1, and possibly one or more of the appendices 3-5, depending on the product(s) purchased by the Licensee, what is stated in the quote or invoice, and/or what is stated in the Licensee's License Certificate.

APPENDIX 1
The modules and/or tools that are included in the respective product - Qt for Application Development (QtAD), Qt for Device Creation (QtDC), Qt for MCUs (QtMCU), Qt 3D Studio (Qt3DS) and Qt Design Studio (QtDS) - are marked with 'X' in the below table.

Parts of the product that are permitted for distribution in object-code form only ("Redistributables") are marked with 'R' in the below table.

<table>
<thead>
<tr>
<th>Modules/Tools</th>
<th>QtAD</th>
<th>QtDC</th>
<th>QtMCU</th>
<th>Qt3DS</th>
<th>QtDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qt Core</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt GUI</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Multimedia</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Multimedia Widgets</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Network</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt QML</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Controls 2</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Dialogs</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Layouts</td>
<td>X,R</td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Test</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-----</td>
<td>-----</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Qt SQL</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Test</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Widgets</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active Qt</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt 3D</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Android Extras</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Bluetooth</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Canvas 3D</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Concurrent</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt D-Bus</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Gamepad</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Graphical Effects</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Help</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Image Formats</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Location</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Mac Extras</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Network Authorization</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt NFC</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Platform Headers</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Positioning</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Print Support</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Purchasing</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt for Python</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Library</td>
<td>X.R</td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----</td>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Qt Quick Controls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Extras</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Widgets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt SCXML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Sensors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Serial Bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Serial Port</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Speech</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt SVG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt UI Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt WebChannel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt WebEngine</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt WebSockets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt WebView</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Windows Extras</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt X11 Extras</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt XML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt XML Patterns</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Wayland Compositor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Charts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Data Visualization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Virtual Keyboard</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boot 2 Qt stack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt OTA</td>
<td></td>
<td>X.R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device Utilities</td>
<td></td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Qt Debugging Bridge (QDB) Daemon</td>
<td></td>
<td>X,R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Ultralite Controls</td>
<td></td>
<td></td>
<td>X,R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Ultralite</td>
<td></td>
<td></td>
<td>X,R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Creator</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Designer (Qt Widget Designer)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Quick Designer (Qt Creator plugin)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Linguist</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Assistant</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lupdate</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lrelease</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmake</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>uic</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rcc</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qlalr</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qdoc</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmlscene</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmlviewer</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target toolchains</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Debugging Bridge (QDB) Host Tools</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qtconfig-gui</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qt Emulator</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmlinterfacegenerator</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmltocpp</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX 3: ADDITIONS TO LICENSED SOFTWARE

In addition to what is provided under the definition of the Licensed Software, Parties agree that Licensed Software shall also include the Add-On Products of The Qt Company, as mentioned in this Appendix, if included in the quote / invoice.

The Modules and/or Tools of the Licensed Software that are included with each Add-On Product respectively are marked with 'X' in the below table. Parts of the respective Add-On Product that are permitted for distribution in object-code form only ("Redistributables") are marked with 'R' in the below table.

<table>
<thead>
<tr>
<th></th>
<th>Add-On Product(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modules / Tools of</td>
<td></td>
</tr>
<tr>
<td>Licensed Software</td>
<td></td>
</tr>
<tr>
<td>Qt for Automation Suite</td>
<td></td>
</tr>
<tr>
<td>Qt Automotive Application</td>
<td></td>
</tr>
<tr>
<td>Qt Safe Renderer</td>
<td></td>
</tr>
<tr>
<td>Qt Application Manager</td>
<td></td>
</tr>
<tr>
<td>Qt Platform Package</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Add-On Product(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qt MQTT</td>
</tr>
<tr>
<td>Qt KNX</td>
</tr>
<tr>
<td>Qt OPC UA</td>
</tr>
<tr>
<td>Qt CoAP</td>
</tr>
<tr>
<td>Qt Safe Renderer</td>
</tr>
<tr>
<td>Qt Application Manager</td>
</tr>
<tr>
<td>Qt IVI</td>
</tr>
<tr>
<td>Reference UI</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Qt GENIVI</td>
</tr>
<tr>
<td>Extras</td>
</tr>
<tr>
<td>Qt GENIVI</td>
</tr>
<tr>
<td>Extras</td>
</tr>
<tr>
<td>QML Live</td>
</tr>
<tr>
<td>Qt Creator</td>
</tr>
<tr>
<td>Deployment</td>
</tr>
<tr>
<td>Qt Creator</td>
</tr>
<tr>
<td>Plugin for</td>
</tr>
<tr>
<td>Qt</td>
</tr>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Manager</td>
</tr>
<tr>
<td>Qt</td>
</tr>
<tr>
<td>Automotive</td>
</tr>
<tr>
<td>Suite</td>
</tr>
<tr>
<td>Deployment</td>
</tr>
<tr>
<td>Server</td>
</tr>
<tr>
<td>Qt Design</td>
</tr>
<tr>
<td>Studio</td>
</tr>
<tr>
<td>Qt 3D Studio</td>
</tr>
<tr>
<td>GammaRay</td>
</tr>
<tr>
<td>Platform</td>
</tr>
<tr>
<td>adaptations</td>
</tr>
<tr>
<td>for</td>
</tr>
<tr>
<td>specified</td>
</tr>
<tr>
<td>Deployment</td>
</tr>
<tr>
<td>Platforms</td>
</tr>
<tr>
<td>Qt for</td>
</tr>
<tr>
<td>Device</td>
</tr>
<tr>
<td>Creation</td>
</tr>
</tbody>
</table>

All the above Redistributables are subject to applicable provisions and limitations including but not limited to what is defined in section 3 of the Agreement.

APPENDIX 4: SMALL BUSINESS AND STARTUP

The provisions of this Appendix 4 are applicable for Start-up Companies and for
the Evaluation Term. For the purpose of this Appendix 4, the following additional definitions shall be applicable:

"Trial Term" shall mean a period of twelve (12) months.

"Start-up Company" means a company with a maximum annual revenue, including funding, equivalent to 100,000 USD (in applicable currency) during a respective calendar year, as evidenced by duly audited records of the Licensee and approved by The Qt Company.

During the Trial Term, Section 3 shall apply with following modifications ("Trial Term Modifications"): (i) Licenses granted under Sections 3.1 and 3.2 shall be free of any charge. For clarity, License for distribution of Devices pursuant to Section 3.3 is subject to applicable License Fee for necessary Distribution Licenses;

(ii) Development License under Section 3.1 is limited to a maximum of three (3) Designated Users; and

(iii) Support is available subject to availability, as judged by The Qt Company at its free and absolute discretion, provided that support will be limited to a maximum of ten (10) tickets during the Trial Term.

Upon expiry of the Trial Term:

(a) This Appendix 4 is terminated, Trial Term Modifications cease to remain in force, Licensee's Development Licenses shall be automatically converted into licenses subject to a License Fee (in the amount specified in the quote or in Appendix 2 and payable with a 30-day payment term) and Licensee's rights and obligations under this Agreement shall continue to remain in force under the standard provisions of the Agreement, unless the Licensee notifies The Qt Company in writing no less than ninety (90) days before such expiry date that Licensee does not agree to such continuance, in which event the Agreement, and all rights of the Licensee thereunder, shall expire; provided however that

(b) in the event the Licensee still qualifies as a Start-up Company, the Licensee has an option ("Option"), instead of what is stated in item a) above, to renew the Trial Term. Renewal is limited to one time, and the total duration of Trial Term is thus 24 months after the effective date. Licensee shall notify The Qt Company in writing, no less than ninety (90) days before the expiry date, if Licensee wish to exercise the Option.

APPENDIX 5: NON-COMMERCIAL USE
The provisions of this Appendix 5 are applicable for non-commercial use of the Licensed Software by the Licensee.
For the purpose of this Appendix 5, the following additional definitions (replacing the relevant definition of the Agreement, where applicable) shall be applicable:

"Demo Units" shall mean (i) hardware development platform, which incorporates the Licensed Software along with Licensee's software and/or hardware, and (ii) prototype versions of Applications or Devices.

"Designated User(s)" shall mean the employees and students of the Licensee.

"Licensee Products" shall mean Applications and/or Devices.

"Permitted Purpose" shall mean (i) Licensee's internal evaluation and testing of Licensed Software, (ii) building Demo Units as well as (iii) educational use.

"Term" shall mean a period of twelve (12) months or any such other period as may be agreed between the Parties.

For the purpose of this Appendix 5, the following changes shall be agreed with respect to relevant Sections of the Agreement:

I. Recital (A) shall be replaced in its entirety to read as follows:

"(A) Licensee wishes to use the Licensed Software for the Permitted Purpose."

II. Section 3.1 shall be replaced in its entirety to read as follows:

"The Qt Company grants to Licensee a personal, non-exclusive, non-transferable, revocable, royalty-free license, valid for the Term, to use, modify and copy the Licensed Software solely for the Permitted Purpose.

Licensee may install copies of the Licensed Software on an unlimited number of computers provided that only Designated Users may use the Licensed Software.

Licensee may demonstrate the Demo Units, provided that such demonstrations must be conducted by Licensee, and the Demo Units must remain in Licensee's possession and under Licensee's control at all times.

For clarity, this Agreement does not (i) entitle Licensee to use Licensed Software to create Applications or Devices (other than prototypes thereof) or (ii) carry any distribution rights to Licensee, but such rights are subject to and conditional upon conclusion of a separate license agreement with The Qt Company."
III. Sections 3.2, 3.3, 8 and 10 shall be deleted.

IV. Section 3.4 shall be replaced in its entirety to read as follows:

"Licensee shall not:
- remove or alter any copyright, trademark or other proprietary rights notice contained in any portion of the Licensed Software;

- transfer, publish, sublicense, disclose, display or otherwise make theLicensed Software available to any third party (except that Licensee may demonstrate the Demo Units pursuant to Section 3.1);

- in any way combine, incorporate or integrate Licensed Software with, or use Licensed Software for creation of, any software created with or incorporating Open Source Qt;

Licensee shall cause all Designated Users who make use of the licenses granted under this Agreement, to be contractually bound to comply with the relevant terms of this Agreement and not to use the Licensed Software beyond the terms hereof. Licensee shall be responsible for any and all actions and omissions of its Designated Users relating to the Licensed Software and use thereof.

Any use of Licensed Software beyond the provisions of this Agreement is strictly prohibited and requires an additional license from The Qt Company."

V. Section 12 shall be replaced in its entirety to read as follows:

"This Agreement shall enter into force upon due acceptance by both Parties and remain in force for the Term, unless and until terminated pursuant to the terms of Section 12.

Upon termination of the Agreement, Licensee shall cease using the Licensed Software. All other copies of Licensed Software in the possession or control of Licensee must be erased or destroyed. An officer of Licensee must, upon request, promptly deliver to The Qt Company a written confirmation that this has occurred."

Except for the modifications specified above, this Appendix carries no change to the terms of the Agreement which shall remain in full force.

// Copyright (c) 2010 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (c) 2015-2018 Google, Inc., Netflix, Inc., Microsoft Corp. and contributors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2005 Patrick Lam
Copyright 2009 Roozbeh Pournader
Copyright 2008,2009 Red Hat, Inc.
Copyright 2008 Danilo egan
Copyright 2012 Google, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the author(s) not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The authors make no
representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE AUTHOR(S) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2015, Randall Stewart and Michael Tuexen
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of usrsctp nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2015 Roy Riojas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

The MIT License (MIT)

Copyright (c) 2011 Einar Otto Stangvik <einaros@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.
Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.
GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification").

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.
3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.
If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above,

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.
That's all there is to it!

Copyright 2010 James Halliday (mail@substack.net)

This project is free software released under the MIT/X11 license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2014 Forbes Lindesay

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2011-2016 Heather Arthur <fayearthur@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
void 0;
The JsonCpp library's source code, including accompanying documentation,
tests and demonstration applications, are licensed under the following
conditions...

The author (Baptiste Lepilleur) explicitly disclaims copyright in all
jurisdictions which recognize such a disclaimer. In such jurisdictions,
this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of
2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur, and is
released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this
software may choose to accept it either as 1) Public Domain, 2) under the
conditions of the MIT License (see below), or 3) under the terms of dual
Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is
described in clear, concise terms at:

    http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

========================================================================
Copyright (c) 2007-2010 Baptiste Lepilleur

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
========================================================================
The MIT license is compatible with both the GPL and commercial software, affording one all of the rights of Public Domain with the minor nuisance of being required to keep the above copyright notice and license text in the source code. Note also that by accepting the Public Domain "license" you can re-license your copy using whatever license you like.

Copyright (C) 2009 Apple Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY APPLE INC. "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APPLE INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

UglifyJS is released under the BSD license:

Copyright 2012-2019 (c) Mihai Bazon <mihai.bazon@gmail.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AS IS AND ANY
EXPRESSION OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The FreeType Project LICENSE

welcomes the use of FreeType.

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- We don't promise that this software works. However, we will be interested in any kind of bug reports. (as is' distribution)

- You can use this software for whatever you want, in parts or full form, without having to pay us. (royalty-free' usage)

- You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the

Copyright 1996-2002, 2006 by
David Turner, Robert Wilhelm, and Werner Lemberg
FreeType code. (‘credits’)

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```
Portions of this software are copyright <year> The FreeType Project (www.freetype.org). All rights reserved.
```

Please replace <year> with the value from the FreeType version you actually use.

Legal Terms
============

0. Definitions
-------------

Throughout this license, the terms `package', `FreeType Project', and `FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project', be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as
specified below.

1. No Warranty

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- Redistribution of source code must retain this license file (`FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.

- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.
We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: `FreeType Project', `FreeType Engine', `FreeType library', or `FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts

There are two mailing lists related to FreeType:

- freetype@nongnu.org
  Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution.
  If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- freetype-devel@nongnu.org
  Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

http://www.freetype.org

--- end of FTL.TXT ---

MIT License

Copyright (c) 2012 James Halliday

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
Permission to use, copy, modify, distribute, and sell this
software and its documentation for any purpose is hereby granted
without fee, provided that the above copyright notice appear in
all copies and that both that copyright notice and this permission
notice appear in supporting documentation, and that the name of
the copyright holders not be used in advertising or publicity
pertaining to distribution of the software without specific,
written prior permission. The copyright holders make no
representations about the suitability of this software for any
purpose. It is provided "as is" without express or implied
warranty.

THE COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO THIS
SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS, IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN
AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF
THIS SOFTWARE.

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.
This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant
Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty
Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.
If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified
Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section.

You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications",

Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retile any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.
5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not
apply to the other works in the aggregate which are not themselves
derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these
copies of the Document, then if the Document is less than one half of
the entire aggregate, the Document's Cover Texts may be placed on
covers that bracket the Document within the aggregate, or the
electronic equivalent of covers if the Document is in electronic form.
Otherwise they must appear on printed covers that bracket the whole
aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may
distribute translations of the Document under the terms of section 4.
Replacing Invariant Sections with translations requires special
permission from their copyright holders, but you may include
translations of some or all Invariant Sections in addition to the
original versions of these Invariant Sections. You may include a
translation of this License, and all the license notices in the
Document, and any Warranty Disclaimers, provided that you also include
the original English version of this License and the original versions
of those notices and disclaimers. In case of a disagreement between
the translation and the original version of this License or a notice
or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements",
"Dedications", or "History", the requirement (section 4) to Preserve
its Title (section 1) will typically require changing the actual
title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except
as expressly provided for under this License. Any other attempt to
copy, modify, sublicense or distribute the Document is void, and will
automatically terminate your rights under this License. However,
parties who have received copies, or rights, from you under this
License will not have their licenses terminated so long as such
parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions
of the GNU Free Documentation License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns. See

Each version of the License is given a distinguishing version number.
If the Document specifies that a particular numbered version of this
License "or any later version" applies to it, you have the option of
following the terms and conditions either of that specified version or
of any later version that has been published (not as a draft) by the
Free Software Foundation. If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and
license notices just after the title page:

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the "with...Texts." line with this:

    with the Invariant Sections being LIST THEIR TITLES, with the
    Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.
Copyright (c) 2010 Jonathan Hartley
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holders, nor those of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

From __init__.py:

```
# COPYRIGHT AND LICENSE
#
# Copyright 2001 Gareth Rees. All rights reserved.
# Copyright 2004-2013 Ned Batchelder. All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#
# 1. Redistributions of source code must retain the above copyright
#    notice, this list of conditions and the following disclaimer.
##
# 2. Redistributions in binary form must reproduce the above copyright
#    notice, this list of conditions and the following disclaimer in the
#    documentation and/or other materials provided with the
#    distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
# A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
# HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
# BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
# OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
# ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
```
# TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
# USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
# DAMAGE.

Copyright 2018-2019, VideoLAN and dav1d authors
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
   list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2014 Component contributors <dev@component.io>

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
This package contains code originally written by Isaac Z. Schlueter.
Used with permission.

Copyright (c) Meryn Stol ("Author")
All rights reserved.

The BSD License

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2014-2018 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Here, glslang proper means core GLSL parsing, HLSL parsing, and SPIR-V code
generation. Glslang proper requires use of two licenses, one that covers
non-preprocessing and an additional one that covers preprocessing.

Bison was removed long ago. You can build glslang from the source grammar,
using tools of your choice, without using bison or any bison files.

Other parts, outside of glslang proper, include:

- gl_types.h, only needed for OpenGL-like reflection, and can be left out of
  a parse and codegen project. See it for its license.

- update_glslang_sources.py, which is not part of the project proper and does
  not need to be used.

- the SPIR-V "remapper", which is optional, but has the same license as
glslang proper

- Google tests and SPIR-V tools, and anything in the external subdirectory
  are external and optional; see them for their respective licenses.

The core of glslang-proper, minus the preprocessor is licenced as follows:

//
// Copyright (C) 2015-2018 Google, Inc.
// Copyright (C) <various other dates and companies>
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//
// Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
// with the distribution.
Neither the name of 3Dlabs Inc. Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The preprocessor has the core license stated above, plus an additional licence:

Copyright (c) 2002, NVIDIA Corporation.

NVIDIA Corporation ("NVIDIA") supplies this software to you in consideration of your agreement to the following terms, and your use, installation, modification or redistribution of this NVIDIA software constitutes acceptance of these terms. If you do not agree with these terms, please do not use, install, modify or redistribute this NVIDIA software.

In consideration of your agreement to abide by the following terms, and subject to these terms, NVIDIA grants you a personal, non-exclusive license, under NVIDIA's copyrights in this original NVIDIA software (the "NVIDIA Software"), to use, reproduce, modify and redistribute the NVIDIA Software, with or without modifications, in source and/or binary forms; provided that if you redistribute the NVIDIA Software, you must retain the copyright notice of NVIDIA, this notice and the following text and disclaimers in all such redistributions of the NVIDIA Software. Neither the name, trademarks, service marks nor logos of NVIDIA Corporation may be used to endorse or promote products derived from the NVIDIA Software without specific prior written permission from NVIDIA. Except as expressly stated in this notice, no other rights or licenses express or implied, are granted by NVIDIA herein, including but not limited to any patent rights that may be infringed by your derivative works or by other works in which the NVIDIA Software may be
incorporated. No hardware is licensed hereunder.

THE NVIDIA SOFTWARE IS BEING PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR ITS USE AND OPERATION EITHER ALONE OR IN COMBINATION WITH OTHER PRODUCTS.

IN NO EVENT SHALL NVIDIA BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, EXEMPLARY, CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, LOST PROFITS; PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) OR ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION OF THE NVIDIA SOFTWARE, HOWEVER CAUSED AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\************************************************************************/
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (C) 2014 Rase-

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2017-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
/*
 *
 * Copyright (c) 2001-2017 Cisco Systems, Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * Redistributions in binary form must reproduce the above
 * copyright notice, this list of conditions and the following
 * disclaimer in the documentation and/or other materials provided
 * with the distribution.
 *
 * Neither the name of the Cisco Systems, Inc. nor the names of its
 * contributors may be used to endorse or promote products derived
 * from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 */

Poly2Tri Copyright (c) 2009-2010, Poly2Tri Contributors
http://code.google.com/p/poly2tri/

All rights reserved.
Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

* Neither the name of Poly2Tri nor the names of its contributors may be
  used to endorse or promote products derived from this software without specific
  prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2002-2006 Marcus Geelnard
Copyright (c) 2006-2016 Camilla Lwy <elmindreda@glfw.org>

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would
   be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not
   be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

Copyright (c) 2016, Mapbox

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

FLICTED маршрутный LEICENSE:LGPL-NOGPL2$  

Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms
** and conditions see https://www.qt.io/terms-conditions. For further
** information use the contact form at https://www.qt.io/contact-us.

** GNU Lesser General Public License Usage
** Alternatively, this file may be used under the terms of the GNU Lesser
** General Public License version 3 as published by the Free Software
** Foundation and appearing in the file LICENSE.LGPL3 included in the
** packaging of this file. Please review the following information to
** ensure the GNU Lesser General Public License version 3 requirements
** will be met: https://www.gnu.org/licenses/lgpl-3.0.html.

** GNU General Public License Usage
** Alternatively, this file may be used under the terms of the GNU
** General Public License version 3 or (at your option) any later version
** approved by the KDE Free Qt Foundation. The licenses are as published by
** the Free Software Foundation and appearing in the file LICENSE.GPL3
** included in the packaging of this file. Please review the following
** information to ensure the GNU General Public License requirements will
** be met: https://www.gnu.org/licenses/gpl-3.0.html.

** $QT_END_LICENSE$
**
*/
* http://www.kurims.kyoto-u.ac.jp/~ooura/fft.html
* Copyright Takuya OOURA, 1996-2001
*
* You may use, copy, modify and distribute this code for any purpose (include
* commercial use) and without fee. Please refer to this package when you modify
* this code.
*/
Copyright (C) 2016 Intel Corporation.
Copyright (C) 2015 Klarlvdalens Datakonsult AB, a KDAB Group company, info@kdab.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2014 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Parts of the following directories are available under Apache v2.0

src/de
Copyright (c) 2009-2011 Christian Kohlschutter

third_party/gwt_exporter
Copyright 2007 Timepedia.org

third_party/gwt-2.5.1
Copyright 2008 Google

java/org/chromium/distiller/dev
Copyright 2008 Google

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the
License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
The MIT License (MIT)

Copyright (c) 2015 JD Ballard

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

Open Source Used in DNAC Platform Antelope 12367
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2012
MIPS Technologies, Inc., California.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the MIPS Technologies, Inc., nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE MIPS TECHNOLOGIES, INC. ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MIPS TECHNOLOGIES, INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Authors:
Branimir Vasic (bvasic@mips.com)
Darko Laus (darko@mips.com)
Djordje Pesut (djordje@mips.com)
Goran Cordasic (goran@mips.com)
Nedeljko Babic (nedeljko.babic.imgtec.com)
Mirjana Vulin (mvulin@mips.com)
Stanislav Ocovaj (socovaj@mips.com)
MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// Copyright (c) 2013 The Chromium Authors. All rights reserved.
// Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
// * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
// * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2014-2017 Calvin Metcalf, Fedor Indutny & contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# spdx-license-ids


A list of [SPDX license](https://spdx.org/licenses/) identifiers

## Installation

[Download JSON directly](https://raw.githubusercontent.com/shinnn/spdx-license-ids/master/index.json), or [use](https://docs.npmjs.com/cli/install) [npm](https://docs.npmjs.com/about-npm):

```
```

npm install spdx-license-ids
```

### [Node.js](https://nodejs.org/) API

#### require('spdx-license-ids')

Type: `string[]`

All license IDs except for the currently deprecated ones.
```javascript
const ids = require('spdx-license-ids');
//=> ['0BSD', 'AAL', 'ADSL', 'AFL-1.1', 'AFL-1.2', 'AFL-2.0', 'AFL-2.1', 'AFL-3.0', 'AGPL-1.0-only', ...]

ids.includes('BSD-3-Clause'); //=> true
ids.includes('CC-BY-1.0'); //=> true

ids.includes('GPL-3.0'); //=> false
```

### require('spdx-license-ids/deprecated')

Type: `string[]`

Deprecated license IDs.

```javascript
const deprecatedIds = require('spdx-license-ids/deprecated');
//=> ['AGPL-1.0', 'AGPL-3.0', 'GFDL-1.1', 'GFDL-1.2', 'GFDL-1.3', 'GPL-1.0', 'GPL-2.0', ...]

deprecatedIds.includes('BSD-3-Clause'); //=> false
deprecatedIds.includes('CC-BY-1.0'); //=> false

deprecatedIds.includes('GPL-3.0'); //=> true
```

## License

[Creative Commons Zero v1.0 Universal](https://creativecommons.org/publicdomain/zero/1.0/deed)

Copyright (c) 2014 Nathan LaFreniere and other contributors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors
Copyright (c) 2013 Aleksey Tulinov <aleksey.tulinov@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the
Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license
document.


You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license
document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this
License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user’s computer
system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
Version.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the Library.
Notice some of these images are derived from Google applications resources. They were provided under the following license:
You may use the materials in this directory without restriction to develop your apps and to use in your apps.

# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
# https://developers.google.com/open-source/clia/individual
# https://developers.google.com/open-source/clia/corporate
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate CLA was used.
#
# Names should be added to this file as:
#   Name <email address>
#
# Please keep the list sorted.

Eric Veach <ericv@google.com>
Jesse Rosenstock <jmr@google.com>
Julien Basch <julienbasch@google.com>
Node.js is licensed for use as follows:

---

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
This license applies to parts of Node.js originating from the
https://github.com/joyent/node repository:

Copyright Joyent, Inc. and other Node contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Node.js license applies to all parts of Node.js that are not externally maintained libraries.

The externally maintained libraries used by Node.js are:

- c-ares, located at deps/cares, is licensed as follows:

Copyright 1998 by the Massachusetts Institute of Technology. Copyright (C) 2007-2013 by Daniel Stenberg

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in
advertising or publicity pertaining to distribution of the software without specific, written prior permission.
M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is"
without express or implied warranty.

- HTTP Parser, located at deps/http_parser, is licensed as follows:

http_parser.c is based on src/http/ngx_http_parse.c from NGINX copyright Igor Sysoev.

Additional changes are licensed under the same terms as NGINX and copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- ICU, located at deps/icu-small, is licensed as follows:

COPYRIGHT AND PERMISSION NOTICE (ICU 58 and later)

Copyright 1991-2016 Unicode, Inc. All rights reserved.
Distributed under the Terms of Use in http://www.unicode.org/copyright.html

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files
or Software are furnished to do so, provided that either
(a) this copyright and permission notice appear with all copies
of the Data Files or Software, or
(b) this copyright and permission notice appear in associated
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale,
use or other dealings in these Data Files or Software without prior
written authorization of the copyright holder.

---------------------
Third-Party Software Licenses

This section contains third-party software notices and/or additional
terms for licensed third-party software components included within ICU
libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies of
the Software and that both the above copyright notice(s) and this
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXCEPT OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale, use
or other dealings in this Software without prior written authorization
of the copyright holder.

All trademarks and registered trademarks mentioned herein are the
property of their respective owners.


# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
#   - https://sourceforge.net/project/?group_id=1519
#   - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#   - Its license terms and conditions are shown below.
#
# --------COPYING.libtabe -------- BEGIN-------------------
#
# /*
# Copyright (c) 1999 TaBE Project.
# Copyright (c) 1999 Pai-Hsiang Hsiao.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
#
# Redistributions of source code must retain the above copyright
# notice, this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions and the following disclaimer in
# the documentation and/or other materials provided with the
# distribution.
# Neither the name of the TaBE Project nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#
# ---------------COPYING.ipadic-----BEGIN-------------------------------
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise. The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
# project or thereafter. Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# ---------------COPYING.ipadic------END-------------------------------

3. Lao Word Break Dictionary Data (laodict.txt)

# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
# (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# -----------------------------------------------
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
#  

4. Burmese Word Break Dictionary Data (burmesedict.txt)

# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
#   github.com/kanyawtech/myanmar-karen-word-lists
#
#  
# Copyright (c) 2013, LeRoy Benjamin Sharon
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
# disclaimer. Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided
# with the distribution.
#
# Neither the name Myanmar Karen Word Lists, nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

# 7. Database Ownership

# The TZ database itself is not an IETF Contribution or an IETF document. Rather it is a pre-existing and regularly updated work that is in the public domain, and is intended to remain in the public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do not apply to the TZ Database or contributions that individuals make to it. Should any claims be made and substantiated against the TZ Database, the organization that is providing the IANA Considerations defined in this RFC, under the memorandum of understanding with the IETF, currently ICANN, may act in accordance with all competent court orders. No ownership claims will be made by ICANN or the IETF Trust on the database or the code. Any person making a contribution to the database or code waives all rights to future claims in that contribution or in the TZ Database.

- libuv, located at deps/uv, is licensed as follows:

libuv is part of the Node project: http://nodejs.org/
libuv may be distributed alone under Node's license:

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

This license applies to all parts of libuv that are not externally maintained libraries.

The externally maintained libraries used by libuv are:

- tree.h (from FreeBSD), copyright Niels Provos. Two clause BSD license.

- inet_pton and inet_ntop implementations, contained in src/inet.c, are copyright the Internet Systems Consortium, Inc., and licensed under the ISC license.

- stdint-msvc2008.h (from msinttypes), copyright Alexander Chemeris. Three clause BSD license.

- pthread-fixes.h, pthread-fixes.c, copyright Google Inc. and Sony Mobile Communications AB. Three clause BSD license.


---

- OpenSSL, located at deps.openssl, is licensed as follows:

---

Copyright (c) 1998-2016 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
   "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:
   "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

====================================================================

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

- Punycode.js, located at lib/punycode.js, is licensed as follows:
Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- V8, located at deps/v8, is licensed as follows:

```
This license applies to all parts of V8 that are not externally maintained libraries. The externally maintained libraries used by V8 are:

- PCRE test suite, located in test/mjsunit/third_party/regexp-pcre/regexp-pcre.js. This is based on the test suite from PCRE-7.3, which is copyrighted by the University of Cambridge and Google, Inc. The copyright notice and license are embedded in regexp-pcre.js.

- Layout tests, located in test/mjsunit/third_party/object-keys. These are based on layout tests from webkit.org which are copyrighted by Apple Computer, Inc. and released under a 3-clause BSD license.

- Strongtalk assembler, the basis of the files assembler-arm-inl.h, assembler-arm.cc, assembler-arm.h, assembler-ia32-inl.h, assembler-ia32.cc, assembler-ia32.h, assembler-x64-inl.h, assembler-x64.cc, assembler-x64.h, assembler-mips-inl.h, assembler-mips.cc, assembler-mips.h, assembler.cc and assembler.h. This code is copyrighted by Sun Microsystems Inc. and released under a 3-clause BSD license.

- Valgrind client API header, located at third_party/valgrind/valgrind.h
```
This is release under the BSD license.

These libraries have their own licenses; we recommend you read them, as their terms may differ from the terms below.

Further license information can be found in LICENSE files located in sub-directories.

Copyright 2014, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- zlib, located at deps/zlib, is licensed as follows:

zlib.h -- interface of the 'zlib' general purpose compression library version 1.2.8, April 28th, 2013

Copyright (C) 1995-2013 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly         Mark Adler
jloup@gzip.org            madler@alumni.caltech.edu

- npm, located at deps/npm, is licensed as follows:

    The npm application
    Copyright (c) npm, Inc. and Contributors
    Licensed on the terms of The Artistic License 2.0

Node package dependencies of the npm application
Copyright (c) their respective copyright owners
Licensed on their respective license terms

The npm public registry at https://registry.npmjs.org
and the npm website at https://www.npmjs.com
Operated by npm, Inc.
Use governed by terms published on https://www.npmjs.com

"Node.js"
Neither npm nor npm, Inc. are affiliated with Joyent, Inc.

The Node.js application
Project of Node Foundation, https://nodejs.org

The npm Logo
Copyright (c) Mathias Pettersson and Brian Hammond

"Gubblebum Blocky" typeface
Copyright (c) Tjarda Koster, https://jelloween.deviantart.com
Used with permission

--------

The Artistic License 2.0
Copyright (c) 2000-2006, The Perl Foundation.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

This license establishes the terms under which a given free software Package may be copied, modified, distributed, and/or redistributed. The intent is that the Copyright Holder maintains some artistic control over the development of that Package while still keeping the Package available as open source and free software.

You are always permitted to make arrangements wholly outside of this license directly with the Copyright Holder of a given Package. If the terms of this license do not permit the full use that you propose to make of the Package, you should contact the Copyright Holder and seek a different licensing arrangement.

Definitions

"Copyright Holder" means the individual(s) or organization(s) named in the copyright notice for the entire Package.

"Contributor" means any party that has contributed code or other material to the Package, in accordance with the Copyright Holder's procedures.

"You" and "your" means any person who would like to copy, distribute, or modify the Package.

"Package" means the collection of files distributed by the Copyright Holder, and derivatives of that collection and/or of those files. A given Package may consist of either the Standard Version, or a Modified Version.

"Distribute" means providing a copy of the Package or making it accessible to anyone else, or in the case of a company or organization, to others outside of your company or organization.

"Distributor Fee" means any fee that you charge for Distributing this Package or providing support for this Package to another party. It does not mean licensing fees.

"Standard Version" refers to the Package if it has not been modified, or has been modified only in ways explicitly requested by the Copyright Holder.
"Modified Version" means the Package, if it has been changed, and such changes were not explicitly requested by the Copyright Holder.

"Original License" means this Artistic License as Distributed with the Standard Version of the Package, in its current version or as it may be modified by The Perl Foundation in the future.

"Source" form means the source code, documentation source, and configuration files for the Package.

"Compiled" form means the compiled bytecode, object code, binary, or any other form resulting from mechanical transformation or translation of the Source form.

Permission for Use and Modification Without Distribution

(1) You are permitted to use the Standard Version and create and use Modified Versions for any purpose without restriction, provided that you do not Distribute the Modified Version.

Permissions for Redistribution of the Standard Version

(2) You may Distribute verbatim copies of the Source form of the Standard Version of this Package in any medium without restriction, either gratis or for a Distributor Fee, provided that you duplicate all of the original copyright notices and associated disclaimers. At your discretion, such verbatim copies may or may not include a Compiled form of the Package.

(3) You may apply any bug fixes, portability changes, and other modifications made available from the Copyright Holder. The resulting Package will still be considered the Standard Version, and as such will be subject to the Original License.

Distribution of Modified Versions of the Package as Source

(4) You may Distribute your Modified Version as Source (either gratis or for a Distributor Fee, and with or without a Compiled form of the Modified Version) provided that you clearly document how it differs from the Standard Version, including, but not limited to, documenting any non-standard features, executables, or modules, and provided that you do at least ONE of the following:

(a) make the Modified Version available to the Copyright Holder of the Standard Version, under the Original License, so that the Copyright Holder may include your modifications in the Standard Version.
(b) ensure that installation of your Modified Version does not prevent the user installing or running the Standard Version. In addition, the Modified Version must bear a name that is different from the name of the Standard Version.

(c) allow anyone who receives a copy of the Modified Version to make the Source form of the Modified Version available to others under

(i) the Original License or

(ii) a license that permits the licensee to freely copy, modify and redistribute the Modified Version using the same licensing terms that apply to the copy that the licensee received, and requires that the Source form of the Modified Version, and of any works derived from it, be made freely available in that license fees are prohibited but Distributor Fees are allowed.

Distribution of Compiled Forms of the Standard Version or Modified Versions without the Source

(5) You may Distribute Compiled forms of the Standard Version without the Source, provided that you include complete instructions on how to get the Source of the Standard Version. Such instructions must be valid at the time of your distribution. If these instructions, at any time while you are carrying out such distribution, become invalid, you must provide new instructions on demand or cease further distribution. If you provide valid instructions or cease distribution within thirty days after you become aware that the instructions are invalid, then you do not forfeit any of your rights under this license.

(6) You may Distribute a Modified Version in Compiled form without the Source, provided that you comply with Section 4 with respect to the Source of the Modified Version.

Aggregating or Linking the Package

(7) You may aggregate the Package (either the Standard Version or Modified Version) with other packages and Distribute the resulting aggregation provided that you do not charge a licensing fee for the Package. Distributor Fees are permitted, and licensing fees for other components in the aggregation are permitted. The terms of this license apply to the use and Distribution of the Standard or Modified Versions as included in the aggregation.

(8) You are permitted to link Modified and Standard Versions with
other works, to embed the Package in a larger work of your own, or to build stand-alone binary or bytecode versions of applications that include the Package, and Distribute the result without restriction, provided the result does not expose a direct interface to the Package.

Items That are Not Considered Part of a Modified Version

(9) Works (including, but not limited to, modules and scripts) that merely extend or make use of the Package, do not, by themselves, cause the Package to be a Modified Version. In addition, such works are not considered parts of the Package itself, and are not subject to the terms of this license.

General Provisions

(10) Any use, modification, and distribution of the Standard or Modified Versions is governed by this Artistic License. By using, modifying or distributing the Package, you accept this license. Do not use, modify, or distribute the Package, if you do not accept this license.

(11) If your Modified Version has been derived from a Modified Version made by someone other than you, you are nevertheless required to ensure that your Modified Version complies with the requirements of this license.

(12) This license does not grant you the right to use any trademark, service mark, tradename, or logo of the Copyright Holder.

(13) This license includes the non-exclusive, worldwide, free-of-charge patent license to make, have made, use, offer to sell, sell, import and otherwise transfer the Package with respect to any patent claims licensable by the Copyright Holder that are necessarily infringed by the Package. If you institute patent litigation (including a cross-claim or counterclaim) against any party alleging that the Package constitutes direct or contributory patent infringement, then this Artistic License to you shall terminate on the date that such litigation is filed.

(14) Disclaimer of Warranty:
THE PACKAGE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS "AS IS' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES. THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT ARE DISCLAIMED TO THE EXTENT PERMITTED BY YOUR LOCAL LAW. UNLESS REQUIRED BY LAW, NO COPYRIGHT HOLDER OR CONTRIBUTOR WILL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING IN ANY WAY OUT OF THE USE OF THE PACKAGE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- GYP, located at tools/gyp, is licensed as follows:

Copyright (c) 2009 Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- marked, located at tools/doc/node_modules/marked, is licensed as follows:

Copyright (c) 2011-2014, Christopher Jeffrey (https://github.com/chjj/)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- cpplint.py, located at tools/cpplint.py, is licensed as follows:

  Copyright (c) 2009 Google Inc. All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- ESLint, located at tools/eslint, is licensed as follows:

  Copyright jQuery Foundation and other contributors, https://jquery.org/

  Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

- gtest, located at deps/gtest, is licensed as follows:

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- node-weak, located at test/gc/node_modules/weak, is licensed as follows:

```
Copyright (c) 2011, Ben Noordhuis <info@bnoordhuis.nl>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

- v8_inspector, located at deps/v8_inspector/third_party/v8_inspector, is licensed as follows:

```
// Copyright 2015 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```
- Jinja2, located at deps/v8_inspector/third_party/jinja2, is licensed as follows:

```
Copyright (c) 2009 by the Jinja Team, see AUTHORS for more details.

Some rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following
  disclaimer in the documentation and/or other materials provided
  with the distribution.

* The names of the contributors may not be used to endorse or
  promote products derived from this software without specific
  prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

- Markupsafe, located at deps/v8_inspector/third_party/markupsafe, is licensed as follows:

```
Copyright (c) 2010 by Armin Ronacher and contributors. See AUTHORS
for more details.

Some rights reserved.

Redistribution and use in source and binary forms of the software as well
as documentation, with or without modification, are permitted provided
that the following conditions are met:

* Redistributions of source code must retain the above copyright
```
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following
  disclaimer in the documentation and/or other materials provided
  with the distribution.

* The names of the contributors may not be used to endorse or
  promote products derived from this software without specific
  prior written permission.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE AND DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

Apache License, Version 2.0

Copyright (c) 2011 Dominic Tarr

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Copyright (c) 1988-1997 Sam Leffler
Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and
its documentation for any purpose is hereby granted without fee, provided
that (i) the above copyright notices and this permission notice appear in
all copies of the software and related documentation, and (ii) the names of
Sam Leffler and Silicon Graphics may not be used in any advertising or
publicity relating to the software without the specific, prior written
permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This is a copy of CONTRIBUTORS file for the Pthreads-win32 library, downloaded from http://sourceware.org/cgi-bin/cvsweb.cgi/~checkout~/pthreads/CONTRIBUTORS?rev=1.32&cvsroot=pthreads-win32

Included here to compliment the Pthreads-win32 license header in wtf/ThreadingWin.cpp file. WebKit is using derived sources of ThreadCondition code from Pthreads-win32.

Contributors (in approximate order of appearance)

[See also the ChangeLog file where individuals are attributed in log entries. Likewise in the FAQ file.]

Ben Elliston bje at cygnus dot com
Initiated the project;
setup the project infrastructure (CVS, web page, etc.);
early prototype routines.
Ross Johnson
pj at callisto dot canberra dot edu dot au
early prototype routines;
ongoing project coordination/maintenance;
implementation of spin locks and barriers;
various enhancements;
bug fixes;
documentation;
testsuite.
Robert Colquhoun
rjc at trump dot net dot au
Early bug fixes.
John E. Bossom
John dot Bossom at cognos dot com
Contributed substantial original working implementation;
bug fixes;
ongoing guidance and standards interpretation.
Anders Norlander
norland at hem2 dot passagen dot se
Early enhancements and runtime checking for supported
Win32 routines.
Tor Lillqvist
tml at iki dot fi
General enhancements;
early bug fixes to condition variables.
Scott Lightner
scott at curriculum dot com
Bug fix.
Kevin Ruland
Kevin dot Ruland at anheuser-busch dot com
Various bug fixes.
Mike Russomiker
miker at eai dot com
Bug fix.
Mark E. Armstrong
avail at pacbell dot net
Bug fixes.
Lorin Hochstein
lmh at xiphos dot ca
genral bug fixes; bug fixes to condition variables.
Peter Slacik
Peter dot Slacik at tatamed dot sk
Bug fixes.
Mumit Khan
khan at xraylith dot wisc dot edu
Fixes to work with Mingw32.
Milan Gardian
mg at tatamed dot sk
Bug fixes and reports/analyses of obscure problems.
Aurelio Medina
aureliom at crt dot com
First implementation of read-write locks.
Graham Dumpleton
Graham dot Dumpleton at ra dot pad dot otc dot telstra dot com dot au
Bug fix in condition variables.
Tristan Savatier
tristan at mpegtv dot com
WinCE port.
Erik Hensema
erik at hensema dot xs4all dot nl
Bug fixes.
Rich Peters
rpeters at micro-magic dot com
Todd Owen at lucidcalm dot dropbear dot id dot au
Bug fixes to dll loading.
Jason Nyenye at nbnet dot nb dot ca
Implementation of async cancelation.
Fred Forester at eticomm dot net
Kevin D. Clark at cabletron dot com
David Baggett at itasoftware dot com
Bug fixes.
Paul Redondo at matchvision dot com
Scott McCaskill at 3dfx dot com
Bug fixes.
Jef Gearhart at tpssys dot com
Bug fix.
Arthur Kantor at bexusa dot com
Mutex enhancements.
Steven Reddies at essem-dot-com dot au
Bug fix.
Alexander Terekho at de dot ibm dot com
Re-implemented and improved read-write locks;
(with Louis Thomas) re-implemented and improved
condition variables;
enhancements to semaphores;
enhancements to mutexes;
new mutex implementation in 'futex' style;
suggested a robust implementation of pthread_once
similar to that implemented by V. Kliahtshko;
system clock change handling re CV timeouts;
bug fixes.
Thomas Pfaff at gmx dot net
Changes to make C version usable with C++ applications;
re-implemented mutex routines to avoid Win32 mutexes
and TryEnterCriticalSection;
procedure to fix Mingw32 thread-safety issues.
Franco Bez at gmx dot de
procedure to fix Mingw32 thread-safety issues.
Louis Thomas at arbitrade dot com
(with Alexander Terekho) re-implemented and improved
condition variables.
David Korn at research dot att dot com
Ported to UWIN.
Phil Frishie, Jr at hawksoft dot com
Bug fix.
Ralf Brase at pdb4 dot siemens dot de
Bug fix.
prionx at junot dot com
prionx at junot dot com
Bug fixes.
Max Woodbury at cds dot duke dot edu
POSIX versioning conditionals;
reduced namespace pollution;
idea to separate routines to reduce statically
linked image sizes.
Rob Fannerrfaner at stonethree dot com
Bug fix.
Michael Johnson michaelj at maine dot rr dot com
Bug fix.
Nicolas Barryboozai at yahoo dot com
Bug fixes.
Piet van Bruggenpietvb at newbridges dot nl
Bug fix.
Makoto Katoraven at oldskool dot jp
AM64 port.
Panagiotis E. Hadjidoukaspeh at hpclab dot ceid dot upatras dot gr
Contributed the QueueUserAPCEx package which
makes preemptive async cancelation possible.
Will Bryantwill dot bryant at ecosm dot com
Borland compiler patch and makefile.
Anuj Goyalanuj dot goyal at gmail dot com
Port to Digital Mars compiler.
Gottlob Fregegottlobfrege at gmail dot com
re-implemented pthread_once (version 2)
(pthread_once cancellation added by rpj).
Vladimir Kliatchkovladimir at kliatchko dot com
reimplemented pthread_once with the same form
as described by A.Terekhov (later version 2);
implementation of MCS (Mellor-Crummey/Scott) locks.
A. HISTORY OF THE SOFTWARE
=================================
Python was created in the early 1990s by Guido van Rossum at Stichting
Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands
as a successor of a language called ABC. Guido remains Python's
principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for
National Research Initiatives (CNRI, see http://www.cnri.reston.va.us)
in Reston, Virginia where he released several versions of the
software.

In May 2000, Guido and the Python core development team moved to
BeOpen.com to form the BeOpen PythonLabs team. In October of the same
year, the PythonLabs team moved to Digital Creations, which became
Zope Corporation. In 2001, the Python Software Foundation (PSF, see
https://www.python.org/psf/) was formed, a non-profit organization
created specifically to own Python-related Intellectual Property.
Zope Corporation was a sponsoring member of the PSF.
All Python releases are Open Source (see [http://www.opensource.org](http://www.opensource.org) for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

<table>
<thead>
<tr>
<th>Release</th>
<th>Derived from</th>
<th>Year</th>
<th>Owner</th>
<th>GPL-compatible? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9.0 thru 1.2</td>
<td>1.2</td>
<td>1991-1995</td>
<td>CWI</td>
<td>yes</td>
</tr>
<tr>
<td>1.3 thru 1.5.2</td>
<td>1.2</td>
<td>1995-1999</td>
<td>CNRI</td>
<td>yes</td>
</tr>
<tr>
<td>1.6</td>
<td>1.5.2</td>
<td>2000</td>
<td>CNRI</td>
<td>no</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>2000</td>
<td>BeOpen.com</td>
<td>no</td>
</tr>
<tr>
<td>1.6.1</td>
<td>1.6</td>
<td>2001</td>
<td>CNRI</td>
<td>yes (2)</td>
</tr>
<tr>
<td>2.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>no</td>
</tr>
<tr>
<td>2.0.1</td>
<td>2.0+1.6.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>2.1+2.0.1</td>
<td>2001</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>2.1.1</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.1.3</td>
<td>2.1.2</td>
<td>2002</td>
<td>PSF</td>
<td>yes</td>
</tr>
<tr>
<td>2.2 and above</td>
<td>2.1.1</td>
<td>2001-now</td>
<td>PSF</td>
<td>yes</td>
</tr>
</tbody>
</table>

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

---

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

---

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

---------------------------------------------

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an
office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the
Individual or Organization ("Licensee") accessing and otherwise using
this software in source or binary form and its associated
documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License
Agreement, BeOpen hereby grants Licensee a non-exclusive,
royalty-free, world-wide license to reproduce, analyze, test, perform
and/or display publicly, prepare derivative works, distribute, and
otherwise use the Software alone or in any derivative version,
provided, however, that the BeOpen Python License is retained in the
Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS"
basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT
INFRINGEMENT ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE
SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS
AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY
DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all
respects by the law of the State of California, excluding conflict of
law provisions. Nothing in this License Agreement shall be deemed to
create any relationship of agency, partnership, or joint venture
between BeOpen and Licensee. This License Agreement does not grant
permission to use BeOpen trademarks or trade names in a trademark
sense to endorse or promote products or services of Licensee, or any
third party. As an exception, the "BeOpen Python" logos available at
http://www.pythonlabs.com/logos.html may be used according to the
permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1. This LICENSE AGREEMENT is between the Corporation for National
Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions.
Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2
--------------------------------------------------

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

mapbox-gl-native copyright (c) 2014-2017 Mapbox.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in
  the documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===========================================================================

Mapbox GL uses portions of Android Gesture Detectors Framework.

Copyright (c) 2012, Almer Thie

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===========================================================================
Mapbox GL uses portions of Android Support Library.

Copyright (c) 2005-2013, The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

===========================================================================

Mapbox GL uses portions of Boost.

Distributed under the Boost Software License, Version 1.0.

http://www.boost.org/LICENSE_1_0.txt

===========================================================================

Mapbox GL uses portions of Clipper.

Author    : Angus Johnson
Version    : 6.1.3a
Date       : 22 January 2014
Website    : http://www.angusj.com
Copyright  : Angus Johnson 2010-2014

License:
Use, modification & distribution is subject to Boost Software License Ver 1.
http://www.boost.org/LICENSE_1_0.txt

Attributions:
The code in this library is an extension of Bala Vatti's clipping algorithm:
"A generic solution to polygon clipping"
http://portal.acm.org/citation.cfm?id=129906

Computer graphics and geometric modeling: implementation and algorithms
By Max K. Agoston
Springer; 1 edition (January 4, 2005)
http://books.google.com/books?q=vatti+clipping+agoston
See also:
"Polygon Offsetting by Computing Winding Numbers"
Paper no. DETC2005-85513 pp. 565-575
ASME 2005 International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference (IDETC/CIE2005)
September 24-28, 2005, Long Beach, California, USA
http://www.me.berkeley.edu/~mcmains/pubs/DAC05OffsetPolygon.pdf

===========================================================================

Mapbox GL uses portions of BugshotKit.

The MIT License (MIT)

Copyright (c) 2014 marcoarment

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===========================================================================

Mapbox GL uses portions of CSS Color Parser.

(c) Dean McNamee <dean@gmail.com>, 2012.
C++ port by Konstantin Kfer <mail@kkaefer.com>, 2014.

https://github.com/deann/css-color-parser-js
https://github.com/kkaefer/css-color-parser-cpp

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

===========================================================================

Mapbox GL uses portions of GLFW.

Copyright (c) 2002-2006 Marcus Geelnard
Copyright (c) 2006-2010 Camilla Berglund <elmindreda@elmindreda.org>

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would
be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not
be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

===========================================================================

Mapbox GL uses portions of libc++.

The libc++ library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license. As a user of this code you may choose
to use it under either license. As a contributor, you agree to allow your code
to be used under both.
Full text of the relevant licenses is included below.

====

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2015 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

====
Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===========================================================================

Mapbox GL uses portions of libcurl.

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2015, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

===========================================================================
Mapbox GL uses portions of libjpeg-turbo.

This software is based in part on the work of the Independent JPEG Group.

Copyright (C)2009-2015 D. R. Commander. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the libjpeg-turbo Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS", AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TurboJPEG/LJT: this implements the TurboJPEG API using libjpeg or libjpeg-turbo

===========================================================================

Mapbox GL uses portions of libpng.

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

This code is released under the libpng license.
libpng versions 1.0.7, July 1, 2000, through 1.6.18, July 23, 2015, are
Copyright (c) 2000-2002, 2004, 2006-2015 Glenn Randers-Pehrson, and are
distributed according to the same disclaimer and license as libpng-1.0.6
with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Mans Rullgard
Cosmin Truta
Gilles Vollant
James Yu

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the
library or against infringement. There is no warranty that our
efforts or the library will fulfill any of your particular purposes
or needs. This library is provided with all faults, and the entire
risk of satisfactory quality, performance, accuracy, and effort is with
the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are
Copyright (c) 1998-2000 Glenn Randers-Pehrson, and are distributed according
to the same disclaimer and license as libpng-0.96, with the following
individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are
Copyright (c) 1996-1997 Andreas Dilger, and are
distributed according to the same disclaimer and license as libpng-0.88,
with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors"
is defined as the following set of individuals:
The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

===========================================================================

Mapbox GL uses portions of libuv.

libuv is part of the Node project: http://nodejs.org/
libuv may be distributed alone under Node's license:

=====

Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

This license applies to all parts of libuv that are not externally maintained libraries.

The externally maintained libraries used by libuv are:

- tree.h (from FreeBSD), copyright Niels Provos. Two clause BSD license.

- inet_pton and inet_ntop implementations, contained in src/inet.c, are copyright the Internet Systems Consortium, Inc., and licensed under the ISC license.

- stdint-msvc2008.h (from msinttypes), copyright Alexander Chemeris. Three clause BSD license.

- pthread-fixes.h, pthread-fixes.c, copyright Google Inc. and Sony Mobile Communications AB. Three clause BSD license.


===========================================================================

Mapbox GL uses portions of libzip.

Copyright (C) 1999-2014 Dieter Baron and Thomas Klausner

The authors can be contacted at <libzip@nih.at>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===========================================================================

Mapbox GL uses portions of LOST.

Copyright (c) 2014 Mapzen

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

===========================================================================

Mapbox GL uses portions of the Mapbox iOS SDK, which was derived from the Route-Me open source project, including the Alpstein fork of it.

The Route-Me license appears below.
Copyright (c) 2008-2013, Route-Me Contributers
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===========================================================================
Mapbox GL uses portions of nunicode.

Copyright (c) 2013 Aleksey Tulinov <aleksey.tulinov@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Mapbox GL uses portions of OkHTTP.

Copyright 2014 Square, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Mapbox GL uses portions of OpenSSL.

LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License
---------------

Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an SSL implementation written
by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with Netscaps SSL.

This library is free for commercial and non-commercial use as long as The following conditions are aheared to. The following conditions
apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young’s, and as such any Copyright notices in the code are not to be removed.
If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used.
This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"
   The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:
   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]
Mapbox GL uses portions of RapidJSON.

Tencent is pleased to support the open source community by making RapidJSON available.

Copyright (C) 2015 THL A29 Limited, a Tencent company, and Milo Yip. All rights reserved.

If you have downloaded a copy of the RapidJSON binary from Tencent, please note that the RapidJSON binary is licensed under the MIT License. If you have downloaded a copy of the RapidJSON source code from Tencent, please note that RapidJSON source code is licensed under the MIT License, except for the third-party components listed below which are subject to different license terms. Your integration of RapidJSON into your own projects may require compliance with the MIT License, as well as the other licenses applicable to the third-party components included within RapidJSON. To avoid the problematic JSON license in your own projects, it's sufficient to exclude the bin/jsonchecker/ directory, as it's the only code under the JSON license. A copy of the MIT License is included in this file.

Other dependencies and licenses:

Open Source Software Licensed Under the BSD License:

The msinttypes r29
Copyright (c) 2006-2013 Alexander Chemeris
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Open Source Software Licensed Under the JSON License:
--------------------------------------------------------------------

json.org
Copyright (c) 2002 JSON.org
All Rights Reserved.

JSON_checker
Copyright (c) 2002 JSON.org
All Rights Reserved.

Terms of the JSON License:
--------------------------------------------------------------------

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Terms of the MIT License:
--------------------------------------------------------------------

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

===========================================================================

Mapbox GL uses portions of Reachability.

Copyright (c) 2011, Tony Million.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===========================================================================

Mapbox GL uses portions of SQLite.

2001 September 15

The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:
May you do good and not evil.
May you find forgiveness for yourself and forgive others.
May you share freely, never taking more than you give.

===========================================================================

Mapbox GL uses portions of SVPulsingAnnotationView.

Copyright (c) 2013, Sam Vermette <hello@samvermette.com>

Permission to use, copy, modify, and/or distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright notice
and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS
OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF
THIS SOFTWARE.

===========================================================================

Mapbox GL uses portions of zlib.

Acknowledgments:

The deflate format used by zlib was defined by Phil Katz. The deflate and
zlib specifications were written by L. Peter Deutsch. Thanks to all the
people who reported problems and suggested various improvements in zlib; they
are too numerous to cite here.

Copyright notice:

(C) 1995-2013 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly  Mark Adler
jlop@gzip.org  madler@alumni.caltech.edu

===========================================================================

Mapbox GL uses portions of Realm Objective-C.

Copyright 2015 Realm Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

****************************************************************************
**
** Copyright (C) YYYY The Qt Company Ltd.
** Contact: https://www.qt.io/licensing/
**
** This file is part of the documentation of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:FDL$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms
** and conditions see https://www.qt.io/terms-conditions. For further
** information use the contact form at https://www.qt.io/contact-us.
**
** GNU Free Documentation License Usage
** Alternatively, this file may be used under the terms of the GNU Free
** Documentation License version 1.3 as published by the Free Software
** Foundation and appearing in the file included in the packaging of
** this file. Please review the following information to ensure
** the GNU Free Documentation License version 1.3 requirements
Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

   i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;

   ii. moral rights retained by the original author(s) and/or performer(s);

   iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;

   iv. rights protecting against unfair competition in regards to a Work, subject to the limitations
v. rights protecting the extraction, dissemination, use and reuse of data in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.

b. Affirmer offers the Work as-is and makes no representations or warranties of any kind concerning
the Work, express, implied, statutory or otherwise, including without limitation warranties of
title, merchantability, fitness for a particular purpose, non infringement, or the absence of
latent or other defects, accuracy, or the present or absence of errors, whether or not
discoverable, all to the greatest extent permissible under applicable law.

c. Affirmer disclaims responsibility for clearing rights of other persons that may apply to the Work
or any use thereof, including without limitation any person's Copyright and Related Rights in the
Work. Further, Affirmer disclaims responsibility for obtaining any necessary consents,
permissions or other rights required for any use of the Work.

d. Affirmer understands and acknowledges that Creative Commons is not a party to this document and
has no duty or obligation with respect to this CC0 or use of the Work.

// Copyright 2015 The Chromium OS Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2014-2015 Calvin Metcalf and browserify-sign contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License (MIT)

Copyright (c) Ben Drucker <bvdrucker@gmail.com> (bendrucker.me)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/**************************************************************************
** Copyright (C) YYYY The Qt Company Ltd.
** Contact: https://www.qt.io/licensing/
**
** This file is part of the FOO module of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:LGPL3$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms
** and conditions see http://www.qt.io/terms-conditions. For further
** information use the contact form at http://www.qt.io/contact-us.
**
** GNU Lesser General Public License Usage
** Alternatively, this file may be used under the terms of the GNU Lesser
** General Public License version 3 as published by the Free Software
** Foundation and appearing in the file LICENSE.LGPLv3 included in the
** packaging of this file. Please review the following information to
** ensure the GNU Lesser General Public License version 3 requirements
** will be met: https://www.gnu.org/licenses/lgpl.html.
** */
** GNU General Public License Usage
** Alternatively, this file may be used under the terms of the GNU
** General Public License version 2.0 or later as published by the Free
** Software Foundation and appearing in the file LICENSE.GPL included in
** the packaging of this file. Please review the following information to
** ensure the GNU General Public License version 2.0 requirements will be
** met: http://www.gnu.org/licenses/gpl-2.0.html.
**
** $QT_END_LICENSE$
**
****************************************************************************/
PLY (Python Lex-Yacc)                   Version 3.4

Copyright (C) 2001-2011,
David M. Beazley (Dabeaz LLC)
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.
* Neither the name of the David Beazley or Dabeaz LLC may be used to
endorse or promote products derived from this software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Jed Watson <jed.watson@me.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MOZILLA PUBLIC LICENSE
Version 1.1

-----------

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the
Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to
the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original
Code, prior Modifications used by a Contributor, and the Modifications
made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the
combination of the Original Code and Modifications, in each case
including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally
accepted in the software development community for the electronic
transfer of data.

1.5. "Executable" means Covered Code in any form other than Source
Code.

1.6. "Initial Developer" means the individual or entity identified
as the Initial Developer in the Source Code notice required by Exhibit
A.
1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:
   A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.
   B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such
2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original
Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
   (a) Third Party Claims.
   If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

   (b) Contributor APIs.
   If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

   (c) Representations.
   Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than
any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients’ rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.
5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.
8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder
9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.


11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this
12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

```
The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is ______________________________________.

The Initial Developer of the Original Code is ____________________.
Portions created by ______________________ are Copyright (C) ______
________________________________. All Rights Reserved.

Contributor(s): ______________________________________.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [___] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file
```
under either the MPL or the [___] License.”

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND OTHER CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR OTHER CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the author nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND OTHER CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR OTHER CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----------------------------------------------------------------------------------

NASM is now licensed under the 2-clause BSD license, also known as the simplified BSD license.

Copyright 1996-2009 the NASM Authors - All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2014-2018 Omar Cornut

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:

* May you do good and not evil.
* May you find forgiveness for yourself and forgive others.
* May you share freely, never taking more than you give.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.
3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.
You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

The MIT License

Copyright (c) 2013 Max Ogden

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2016 Guillermo Rauch (@rauchg)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2013-2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2017 Garen J. Torikian

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011-2014 TJ Holowaychuk
Copyright (c) 2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# https://developers.google.com/open-source/cla/individual
# https://developers.google.com/open-source/cla/corporate
#
# Names should be added to this file as:
#   Name <email address>

David Beaumont <dbeaumont@google.com>
David Yonge-Mallo <davinci@google.com>
Dong Zhou <dongzhou@google.com>
(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License (MIT))

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
Copyright (c) 1987, 1988 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

Copyright 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Digital not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Copyright (c) 2001 Robert Penner
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the author nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
	node-http-proxy

Copyright (c) 2010-2016 Charlie Robbins, Jarrett Cruger & the Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License, Version 2.0
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide,

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and
- You must cause any modified files to carry prominent notices stating that You changed the files; and
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and where such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.
This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Copyright (C) 2012 Apple Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY APPLE, INC. "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APPLE, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (c) spdx-satisfies.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar licence but with different Copyright notices) all the files are:

Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
Copyright (c) 2018 Toru Nagashima

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2015 Gareth Jones (with contributions from many other people)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Condition of use and distribution are the same as zlib:

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgement in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Copyright (c) 2013-2014 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.
Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
// Copyright (c) 2012 The Chromium Authors. All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
//   notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
//   copyright notice, this list of conditions and the following disclaimer
//   in the documentation and/or other materials provided with the
//   distribution.
// * Neither the name of Google Inc. nor the names of its
//   contributors may be used to endorse or promote products derived from
//   this software without specific prior written permission.

// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2010 passive.ly LLC

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The libarchive distribution as a whole is Copyright by Tim Kientzle and is subject to the copyright notice reproduced at the bottom of this file.

Each individual file in this distribution should have a clear copyright/licensing statement at the beginning of the file. If any do not, please let me know and I will rectify it. The following is intended to summarize the copyright status of the individual files; the actual statements in the files are controlling.

* Except as listed below, all C sources (including .c and .h files) and documentation files are subject to the copyright notice reproduced at the bottom of this file.

* The following source files are also subject in whole or in part to a 3-clause UC Regents copyright; please read the individual source files for details:
  libarchive/archive_entry.c
  libarchive/archive_read_support_compression_compress.c
  libarchive/archive_write_set_compression_compress.c
  libarchive/mtree.5
  tar/matching.c

* The following source files are in the public domain:
  tar/getdate.c

* The build files---including Makefiles, configure scripts, and auxiliary scripts used as part of the compile process---have widely varying licensing terms. Please check individual files before distributing them to see if those restrictions apply to you.

I intend for all new source code to use the license below and hope over time to replace code with other licenses with new implementations that do use the license below. The varying licensing of the build scripts seems to be an unavoidable mess.

Copyright (c) 2003-2009 <author(s)>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer
in this position and unchanged.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR(S) "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2012 Colingo.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
This file contains a list of people who have made large contributions
to the public version of Protocol Buffers.

Original Protocol Buffers design and implementation:
Sanjay Ghemawat <sanjay@google.com>
Jeff Dean <jeff@google.com>
Daniel Dulitz <daniel@google.com>
Craig Silverstein
Paul Haahr <haahr@google.com>
Corey Anderson <corin@google.com>
(and many others)

Proto2 C++ and Java primary author:
Open Source Used In DNAC Platform Antelope

Kenton Varda <kenton@google.com>

Proto2 Python primary authors:
Will Robinson <robinson@google.com>
Petar Petrov <petar@google.com>

Java Nano primary authors:
Brian Duff <bduff@google.com>
Tom Chao <chaot@google.com>
Max Cai <maxtroy@google.com>
Ulas Kirazci <ulas@google.com>

Large code contributions:
Jason Hsueh <jasonh@google.com>
Joseph Schorr <jschorr@google.com>
Wenbo Zhu <wenboz@google.com>

Large quantity of code reviews:
Scott Bruce <sbruce@google.com>
Frank Yellin
Neal Norwitz <nnorwitz@google.com>
Jeffrey Yasskin <jyasskin@google.com>
Ambrose Feinstein <ambrose@google.com>

Documentation:
Lisa Carey <lcarey@google.com>

Maven packaging:
Gregory Kick <gak@google.com>

Patch contributors:
Kevin Ko <kevin.s.ko@gmail.com>
* Small patch to handle trailing slashes in --proto_path flag.
Johan Euphrosine <proppy@aminche.com>
* Small patch to fix Python CallMethod().
Ulrich Kunitz <kune@deine-taler.de>
* Small optimizations to Python serialization.
Leandro Lucarella <llucax@gmail.com>
* VI syntax highlighting tweaks.
* Fix compiler to not make output executable.
Dilip Joseph <dilip.antony.joseph@gmail.com>
* Heuristic detection of sub-messages when printing unknown fields in text format.
Brian Atkinson <nairb774@gmail.com>
* Added @Override annotation to generated Java code where appropriate.
Vincent Choinire <Choiniere.Vincent@hydro.qc.ca>
* Tru64 support.
Monty Taylor <monty.taylor@gmail.com>
* Solaris 10 + Sun Studio fixes.
Alek Storm <alek.storm@gmail.com>
  * Slicing support for repeated scalar fields for the Python API.
Oleg Smolsky <oleg.smolsky@gmail.com>
  * MS Visual Studio error format option.
  * Detect unordered_map in stl_hash.m4.
Brian Olson <brianolson@google.com>
  * gzip/zlib I/O support.
Michael Poole <mdpoole@trollus.org>
  * Fixed warnings about generated constructors not explicitly initializing
    all fields (only present with certain compiler settings).
  * Added generation of field number constants.
Wink Saville <wink@google.com>
  * Fixed initialization ordering problem in logging code.
Will Pierce <willp@nuclei.com>
  * Small patch improving performance of in Python serialization.
Alexandre Vassalotti <alexandre@peadrop.com>
  * Emacs mode for Protocol Buffers (editors/protobuf-mode.el).
Scott Stafford <scott.stafford@gmail.com>
  * Added Swap(), SwapElements(), and RemoveLast() to Reflection interface.
Alexander Melnikov <alm@sibmail.ru>
  * HPUX support.
Oliver Jowett <oliver.jowett@gmail.com>
  * Detect whether zlib is new enough in configure script.
  * Fixes for Solaris 10 32/64-bit confusion.
Evan Jones <evanj@mit.edu>
  * Optimize Java serialization code when writing a small message to a stream.
  * Optimize Java serialization of strings so that UTF-8 encoding happens only
    once per string per serialization call.
  * Clean up some Java warnings.
  * Fix bug with permanent callbacks that delete themselves when run.
Michael Kucharski <m.kucharski@gmail.com>
  * Added CodedInputStream.getTotalBytesRead().
Kacper Kowalik <xarthiusius.kk@gmail.com>
  * Fixed m4/axc_pthread.m4 problem for some Linux distributions.
William Orr <will@worrbase.com>
  * Fixed detection of sched_yield on Solaris.
  * Added atomicops for Solaris
Andrew Paprocki <andrew@ishiboo.com>
  * Fixed minor IBM xIC compiler build issues
  * Added atomicops for AIX (POWER)
The MIT License

Copyright (C) 2011-2019 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License

Copyright (c) 2010 Timothy Farrell

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2017, The CRC32C Authors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)
Copyright (c) 2017 Samuel Reed <samuel.trace.reed@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Licensed under the Academic Free License version 2.0 (below)
Or under the following terms:

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the
Free Software Foundation, Inc., 59 Temple Place - Suite 330,
Boston, MA 02111-1307, USA.
This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Academic Free License version 2.0

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:
   a) to reproduce the Original Work in copies;
   b) to prepare derivative works ("Derivative Works") based upon the Original Work;
   c) to distribute copies of the Original Work and Derivative Works to the public;
   d) to perform the Original Work publicly; and
   e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as
expressly stated herein. No patent license is granted to make, use, sell or
offer to sell embodiments of any patent claims other than the licensed claims
defined in Section 2. No right is granted to the trademarks of Licensor even if
such marks are included in the Original Work. Nothing in this License shall be
interpreted to prohibit Licensor from licensing under different terms from this
License any Original Work that Licensor otherwise would have a right to
license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative
Works that You create, all copyright, patent or trademark notices from the
Source Code of the Original Work, as well as any notices of licensing and any
descriptive text identified therein as an "Attribution Notice." You must cause
the Source Code for any Derivative Works that You create to carry a prominent
Attribution Notice reasonably calculated to inform recipients that You have
modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that
the copyright in and to the Original Work and the patent rights granted herein
by Licensor are owned by the Licensor or are sublicensed to You under the terms
of this License with the permission of the contributor(s) of those copyrights
and patent rights. Except as expressly stated in the immediately proceeding
sentence, the Original Work is provided under this License on an "AS IS" BASIS
and WITHOUT WARRANTY, either express or implied, including, without limitation,
the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU.
This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No
license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory,
whether in tort (including negligence), contract, or otherwise, shall the
Licensor be liable to any person for any direct, indirect, special, incidental,
or consequential damages of any character arising as a result of this License
or the use of the Original Work including, without limitation, damages for loss
of goodwill, work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses. This limitation of liability shall not
apply to liability for death or personal injury resulting from Licensor's
negligence to the extent applicable law prohibits such limitation. Some
jurisdictions do not allow the exclusion or limitation of incidental or
consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or
a Derivative Work, You must make a reasonable effort under the circumstances to
obtain the express assent of recipients to the terms of this License. Nothing
else but this License (or another written agreement between Licensor and You)
grants You permission to create Derivative Works based upon the Original Work
or to exercise any of the rights granted in Section 1 herein, and any attempt
to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, for patent infringement (i) against Licensor with respect to a patent applicable to software or (ii) against any entity with respect to a patent applicable to the Original Work (but excluding combinations of the Original Work with other software or hardware).

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise
restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

This license is Copyright (C) 2003 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

BoringSSL is a fork of OpenSSL. As such, large parts of it fall under OpenSSL licensing. Files that are completely new have a Google copyright and an ISC license. This license is reproduced at the bottom of this file.

Contributors to BoringSSL are required to follow the CLA rules for Chromium: https://cla.developers.google.com/clas

Files in third_party/ have their own licenses, as described therein. The MIT license, for third_party/fiat, which, unlike other third_party directories, is compiled into non-test libraries, is included below.

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

The following are Google-internal bug numbers where explicit permission from some authors is recorded for use of their work. (This is purely for our own record keeping.)
27287199
27287880
27287883

OpenSSL License
-------------

/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
 * *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 */
* 3. All advertising materials mentioning features or use of this
software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit. (http://www.openssl.org)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
endorse or promote products derived from this software without
prior written permission. For written permission, please contact
openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
nor may "OpenSSL" appear in their names without prior written
permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
acknowledgment:
"This product includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit (http://www.openssl.org)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
EXPressed OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.
*
Original SSLeay License
-----------------------------------------
*
This product includes cryptographic software written by Eric Young
eay@cryptsoft.com). This product includes software written by Tim
Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----------------------------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.
*
This package is an SSL implementation written
by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed.
If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used.
This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"
The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:
   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be
ISC license used for completely new code in BoringSSL:

/* Copyright (c) 2015, Google Inc.
 *
 * Permission to use, copy, modify, and/or distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
 * SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
 * OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
 * CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. */

The code in third_party/fiat carries the MIT license:

Copyright (c) 2015-2016 the fiat-crypto authors (see

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Licenses for support code
-------------------------
Parts of the TLS test suite are under the Go license. This code is not included in BoringSSL (i.e. libcrypto and libssl) when compiled, however, so distributing code linked against BoringSSL does not trigger this license:

Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BoringSSL uses the Chromium test infrastructure to run a continuous build, trybots etc. The scripts which manage this, and the script for generating build metadata, are under the Chromium license. Distributing code linked against BoringSSL does not trigger this license.

Copyright 2015 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# This is the official list of people who can contribute
# (and typically have contributed) code to the RE2 repository.
The AUTHORS file lists the copyright holders; this file lists people. For example, Google employees are listed here but not in AUTHORS, because Google holds the copyright.

The submission process automatically checks to make sure that people submitting code are listed in this file (by email address).

Names should be added to this file only after verifying that the individual or the individual's organization has agreed to the appropriate Contributor License Agreement, found here:

http://code.google.com/legal/individual-cla-v1.0.html

The agreement for individuals can be filled out on the web.

When adding J Random Contributor's name to this file, either J's name or J's organization's name should be added to the AUTHORS file, depending on whether the individual or corporate CLA was used.

Names should be added to this file like so:

Name <email address>

Please keep the list sorted.

Dominic Battre <battre@chromium.org>
Doug Kwan <dougkwan@gmail.com>
Dmitriy Vyukov <dvyukov@gmail.com>
John Millikin <jmillikin@gmail.com>
Mike Nazarewicz <mpn@gmail.com>
Nico Weber <thakis@chromium.org>
Pawel Hajdan <phajdan.jr@gmail.com>
Rob Pike <r@google.com>
Russ Cox <rsc@swtch.com>
Sanjay Ghemawat <sanjay@google.com>
Stefano Rivera <stefano.rivera@gmail.com>
Srinivasan Venkatachary <sri@swtch.com>
Viatcheslav Ostapenko <slostapenko@samsung.com>

MIT License

Copyright (c) 2012-2014 Chris Pettit

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017 Matt Lewis

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2006, 2008 Edward Rosten
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

*Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

*Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
*Neither the name of the University of Cambridge nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*
 * Copyright(c)1995,97 Mark Olesen <olesen@me.QueensU.CA>
 * Queen's Univ at Kingston (Canada)
 *
 * Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.
 *
 * THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR QUEEN'S UNIVERSITY AT KINGSTON MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.
 *
 * All of which is to say that you can do what you like with this source code provided you don't try to sell it as your own and you include an unaltered copy of this message (including the copyright).
 *
 * It is also implicitly understood that bug fixes and improvements should make their way back to the general Internet community so that everyone benefits.
 */

----------------------------------------------------------------

Notice that the following BSD-style license applies to this one file (valgrind.h) only. The rest of Valgrind is licensed under the terms of the GNU General Public License, version 2, unless otherwise indicated. See the COPYING file in the source distribution for details.
This file is part of Valgrind, a dynamic binary instrumentation framework.

Copyright (C) 2000-2007 Julian Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2012 Yahoo Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# This is the official list of people who can contribute
# (and who have contributed) code to the ANGLE project
# repository.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#

TransGaming Inc.
Nicolas Capens
Daniel Koch
Geoff Lang
Andrew Lewycky
Jamie Madill
Gavriel State
Shannon Woods

Google Inc.
Brent Austin
Michael Bai
John Bauman
Peter Beverloo
Steve Block
Rachel Blum
Eric Boren
Henry Bridge
Nat Duca
Courtney Goeltzenleuchter
Peter Kasting
Vangelis Kokkevis
Zhenyao Mo
Daniel Nicoara
Alastair Patrick
Alok Priyadarshri
Kenneth Russell
Brian Salomon
Gregg Tavares
Jeff Timanus
Ben Vanik
Adrienne Walker
thestig@chromium.org
Justin Schuh
Scott Graham
Corentin Wallez
Kai Ninomiya
Victor Costan
Shahbaz Youssefi

Adobe Systems Inc.
Alexandru Chiculita
Steve Minns
Max Vujovic

Autodesk, Inc.
Ranger Harke

Cloud Party, Inc.
Conor Dickinson

The Qt Company Ltd.
Andrew Knight

Imagination Technologies Ltd.
Gregoire Payen de La Garanderie

Intel Corporation
Jin Yang
Andy Chen
Josh Triplett
Sudarsana Nagineni
Jiajia Qin
Jiawei Shao
Jie Chen
Qiankun Miao
Bryan Bernhart
Yunchao He
Xinghua Cao
Brandon Jones
Enrico Galli

Klarlvdalens Datakonsult AB
Milian Wolff

Mozilla Corp.
Dzmitry Malysahu
Ehsan Akhgari
Edwin Flores
Jeff Gilbert
Mike Hommey
Benoit Jacob
Makoto Kato
Vladimir Vukićević
Josh Matthews

Turbulenz
Michael Braithwaite

Ulrik Persson (ddefrostt)
Mark Banner (standard8mbp)
David Kilzer
Jacek Caban
Tibor den Ouden
Rgis Fnon
Sebastian Bergstein
James Ross-Gowan
Andrei Volykhin
Jrme Duval
Till Rathmann
Nick Shaforostov

Microsoft Corporation
Cooper Partin
Austin Kinross
Mingyu Hu
Minmin Gong
Shawn Hargreaves
Rafael Cintron
James Clarke
Nathan Zabriskie

Microsoft Open Technologies, Inc.
Cooper Partin
Austin Kinross

NVIDIA Corporation
Olli Etuaho
Arun Patole
Qingqing Deng
Kimmo Kinnunen
Sami Visnen
Martin Radev
Joonatan Saarhelo
GNU LESSER GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2015 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt Toolkit under the terms of
GNU Lesser General Public License version 3, which is displayed below.
This license makes reference to the version 3 of the GNU General
Public License, which you can find in the LICENSE.GPLv3 file.

-------------------------------------------------------------------------
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this
licensedocument, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, this License refers to version 3 of the GNU Lesser
General Public License, and the GNU GPL refers to version 3 of the
GNU General Public License.

The Library refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An Application is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A Combined Work is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the Linked
Version.

The Minimal Corresponding Source for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The Corresponding Application Code for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.
If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.
The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License or any later version applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

/*
 * Written by Wilco Dijkstra, 1996. The following email exchange establishes the license.
 * *
 * From: Wilco Dijkstra <Wilco.Dijkstra@ntlworld.com>
 * Date: Fri, Jun 24, 2011 at 3:20 AM
 * Subject: Re: sqrt routine
 * To: Kevin Ma <kma@google.com>
 * Hi Kevin,
 * Thanks for asking. Those routines are public domain (originally posted to comp.sys.arm a long time ago), so you can use them freely for any purpose.
 * Cheers,
 * Wilco
 *
 * ----- Original Message -----*
 * From: "Kevin Ma" <kma@google.com>
 * To: <Wilco.Dijkstra@ntlworld.com>
 * Sent: Thursday, June 23, 2011 11:44 PM
 * Subject: Fwd: sqrt routine
 * Hi Wilco,
 * I saw your sqrt routine from several web sites, including http://www.finesse.demon.co.uk/steven/sqrt.html.
 * Just wonder if there's any copyright information with your Successive approximation routines, or if I can freely use it for any purpose.
 * Thanks.
 * Kevin
 */

Mozilla Public License, version 2.0
1. Definitions

1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"

means Covered Software of a particular Contributor.

1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"

means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"

means any form of the work other than Source Code Form.

1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. "License"

means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications"

means any of the following:

a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

b. any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

a. for any code that a Contributor has removed from Covered Software; or

b. for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation
Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

a. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

b. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered
Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing
basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty

Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may
not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of
the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

The code in lib/flat_buffers.dart is based on code that was released under the following license:

Copyright 2012, the Dart project authors. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

To the extent permissible, the changes to that code and the other assets in this package are licensed under the Apache2 license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2014 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2013 The Chromium Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (C) 2013 Vojta Jna.

Open Source Used In DNAC Platform Antelope 12514
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (C) 2011-2015 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (C) 2014 Wei Fanzhe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// Copyright 2019 The Fuchsia Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//  * Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
//  * Redistributions in binary form must reproduce the above
//    copyright notice, this list of conditions and the following disclaimer
//    in the documentation and/or other materials provided with the
//    distribution.
//  * Neither the name of Google Inc. nor the names of its
//    contributors may be used to endorse or promote products derived from
//    this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 1999 Serika Kurusugawa, All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS `"AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Copyright (c) 2017, crypto-browserify contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

W3C SOFTWARE NOTICE AND LICENSE

This license came from: https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document

This work is being provided by the copyright holders under the following license.

License
By obtaining and/or copying this work, you (the licensee) agree that you have read, understood, and will comply
with the following terms and conditions.

Permission to copy, modify, and distribute this work, with or without modification, for any purpose and without fee
or royalty is hereby granted, provided that you include the following on ALL copies of the work or portions thereof,
including modifications:

- The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
- Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C
  Software and Document Short Notice should be included.
- Notice of any changes or modifications, through a copyright statement on the new code or document such as "This
  software or document includes material copied from or derived from [title and URI of the W3C document].
  Copyright [YEAR] W3C (MIT, ERCIM, Keio, Beihang)."

Disclaimers
THIS WORK IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR
WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF
MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE
SOFTWARE OR DOCUMENT WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS,
TRADEMARKS OR OTHER RIGHTS.
COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENT.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the work
without specific, written prior permission. Title to copyright in this work will at all times remain with copyright
holders.

NOTE! The LGPL below is copyrighted by the Free Software Foundation, but
the instance of code that it refers to (the kde libraries) are copyrighted
by the authors who actually wrote it.

---------------------------------------------------------------------------
GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1301, USA.

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is
numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some
specially designated Free Software Foundation software, and to any
other libraries whose authors decide to use it. You can use it for
your libraries, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if
you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.
However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.
5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding
Open Source Used In DNAC Platform Antelope 12524

machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.
It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation.
Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

MIT License

Copyright (c) 2016 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
** packaging of this file. Please review the following information to
** ensure the GNU Lesser General Public License version 2.1 requirements
** will be met: http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html.
**
** If you have questions regarding the use of this file, please contact
** us via http://www.qt.io/contact-us/.
**
** $QT_END_LICENSE$
**
****************************************************************************/
Copyright (c) 2015-2016 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and/or associated documentation files (the
"Materials"), to deal in the Materials without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Materials, and to
permit persons to whom the Materials are furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Materials.

MODIFICATIONS TO THIS FILE MAY MEAN IT NO LONGER ACCURATELY REFLECTS
KHRONOS STANDARDS. THE UNMODIFIED, NORMATIVE VERSIONS OF KHRONOS
SPECIFICATIONS AND HEADER INFORMATION ARE LOCATED AT
https://www.khronos.org/registry/

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.
Copyright (c) 2012 Julian Berman

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

MIT License

Copyright (c) 2019 Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com), Paul Miller
(https://paulmillr.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2011 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2014-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2014, Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

amdefine is released under two licenses: new BSD, and MIT. You may pick the license that best suits your development needs. The text of both licenses are provided below.

The "New" BSD License:

Copyright (c) 2011-2016, The Dojo Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.
* Neither the name of the Dojo Foundation nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License
-----------

Copyright (c) 2011-2016, The Dojo Foundation

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2012-2018 Luke Plaster <notatestuser@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
(The MIT License)

Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

// Copyright 2018 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//  * Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
//  * Redistributions in binary form must reproduce the above
//    copyright notice, this list of conditions and the following disclaimer
//    in the documentation and/or other materials provided with the
//    distribution.
Copyright 2018 kumavis

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright (c) 2007-2010 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials.
THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

SGI FREE SOFTWARE LICENSE B (Version 2.0, Sept. 18, 2008)

Copyright (C) 1992 Silicon Graphics, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

## Glob Logo

Glob’s logo created by Tanya Brassie <http://tanyabrassie.com/> , licensed under a Creative Commons Attribution-ShareAlike 4.0 International License https://creativecommons.org/licenses/by-sa/4.0/

'zlib' general purpose compression library
version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented: you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu
Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the names of the Mozilla Foundation nor the names of project contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
(The MIT License)

Copyright (c) 2014-2015 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Permission is hereby granted, free of charge, to any person obtaining a
# copy of this software and associated documentation files (the "Software"),
# to deal in the Software without restriction, including without limitation
# the rights to use, copy, modify, merge, publish, distribute, sublicense,
# and/or sell copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
# OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
# THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
# OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
# ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.

The Clipper code library, the "Software" (that includes Delphi, C++ & C# source code, accompanying samples and documentation), has been released under the following license, terms and conditions:

Boost Software License - Version 1.0 - August 17th, 2003
http://www.boost.org/LICENSE_1_0.txt

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2016 Kirill Fomichev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (C) 2000-2017 Julian Seward. All rights reserved.
Copyright (C) 2003-2017 Josef Weidendorfer. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2007-2008, Apple, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Apple, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

****************************************************************************
**
** Copyright (C) YYYY The Qt Company Ltd.
** Contact: https://www.qt.io/licensing/
**
** This file is part of the FOO module of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:BSD$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms
** and conditions see https://www.qt.io/terms-conditions. For further
** information use the contact form at https://www.qt.io/contact-us.
**
** BSD License Usage
** Alternatively, you may use this file under the terms of the BSD license
** as follows:
**
** "Redistribution and use in source and binary forms, with or without
** modification, are permitted provided that the following conditions are
** met:
** * Redistributions of source code must retain the above copyright
**   notice, this list of conditions and the following disclaimer.
** * Redistributions in binary form must reproduce the above copyright
**   notice, this list of conditions and the following disclaimer in
**   the documentation and/or other materials provided with the
**   distribution.
** * Neither the name of The Qt Company Ltd nor the names of its
**   contributors may be used to endorse or promote products derived
**   from this software without specific prior written permission.
**
** THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
** "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
** LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
** A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
** OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
** SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
** LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
** DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
** THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
** (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
** OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."
**
** $QT_END_LICENSE$
**

The ISC License

Copyright (c) 2015, 2019 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Copyright (c) 2005, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ESLint
Copyright jQuery Foundation and other contributors, https://jquery.org/

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the assimp team, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of the assimp team.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

If you want to write an option parser, and have it be good, there are two ways to do it. The Right Way, and the Wrong Way.

The Wrong Way is to sit down and write an option parser. We've all done that.

The Right Way is to write some complex configurable program with so many options that you hit the limit of your frustration just trying to manage them all, and defer it with duct-tape solutions until you see exactly to the core of the problem, and finally snap and write an awesome option parser.
If you want to write an option parser, don't write an option parser. Write a package manager, or a source control system, or a service restarter, or an operating system. You probably won't end up with a good one of those, but if you don't give up, and you are relentless and diligent enough in your procrastination, you may just end up with a very nice option parser.

## USAGE

```javascript
// my-program.js
var nopt = require("nopt")
, Stream = require("stream").Stream
, path = require("path")
, knownOpts = { "foo" : [String, null]
, "bar" : [Stream, Number]
, "baz" : path
, "bloo" : [ "big", "medium", "small" ]
, "flag" : Boolean
, "pick" : Boolean
, "many1" : [String, Array]
, "many2" : [path, Array]
}
, shortHands = { "foofoo" : ["--foo", "Mr. Foo"]
, "b7" : ["--bar", "7"]
, "m" : ["--bloo", "medium"]
, "p" : ["--pick"]
, "f" : ["--flag"]
}
// everything is optional.
// knownOpts and shorthands default to {}
// arg list defaults to process.argv
// slice defaults to 2
, parsed = nopt(knownOpts, shortHands, process.argv, 2)
console.log(parsed)
```

This would give you support for any of the following:

```console
$ node my-program.js --foo "blerp" --no-flag
{ "foo" : "blerp", "flag" : false }

$ node my-program.js ---bar 7 --foo "Mr. Hand" --flag
{ bar: 7, foo: "Mr. Hand", flag: true }

$ node my-program.js --foo "blerp" -f -----p
{ foo: "blerp", flag: true, pick: true }

$ node my-program.js -fp --foofoo
{ foo: "Mr. Foo", flag: true, pick: true }

$ node my-program.js --foofoo -- -fp # -- stops the flag parsing.
{ foo: "Mr. Foo", argv: { remain: ["-fp"] } }

$ node my-program.js --blatzk -fp # unknown opts are ok.
{ blatzk: true, flag: true, pick: true }

$ node my-program.js --blatzk=1000 -fp # but you need to use = if they have a value
{ blatzk: 1000, flag: true, pick: true }

$ node my-program.js --no-blatzk -fp # unless they start with "no-
{ blatzk: false, flag: true, pick: true }

$ node my-program.js --baz b/a/z # known paths are resolved.
{ baz: "/Users/isaacs/b/a/z" }

# if Array is one of the types, then it can take many
# values, and will always be an array. The other types provided
# specify what types are allowed in the list.

$ node my-program.js --many1 5 --many1 null --many1 foo
{ many1: ["5", "null", "foo"] }

$ node my-program.js --many2 foo --many2 bar
{ many2: "/path/to/foo", "path/to/bar" }

Read the tests at the bottom of `lib/nopt.js` for more examples of
what this puppy can do.

## Types

The following types are supported, and defined on `nopt.typeDefs`

* String: A normal string. No parsing is done.
* path: A file system path. Gets resolved against cwd if not absolute.
* url: A url. If it doesn't parse, it isn't accepted.
* Number: Must be numeric.
* Date: Must parse as a date. If it does, and `Date` is one of the options,
then it will return a Date object, not a string.
* Boolean: Must be either `true` or `false`. If an option is a boolean,
then it does not need a value, and its presence will imply `true` as
the value. To negate boolean flags, do `--no-whatever` or `--whatever
false`
* NaN: Means that the option is strictly not allowed. Any value will fail.
* Stream: An object matching the "Stream" class in node. Valuable for use when validating programmatically. (npm uses this to let you supply any WriteStream on the `outfd` and `logfd` config options.)
* Array: If `Array` is specified as one of the types, then the value will be parsed as a list of options. This means that multiple values can be specified, and that the value will always be an array.

If a type is an array of values not on this list, then those are considered valid values. For instance, in the example above, the `--bloo` option can only be one of `"big"`, `"medium"`, or `"small"`, and any other value will be rejected.

When parsing unknown fields, `"true"`, `"false"`, and `"null"` will be interpreted as their JavaScript equivalents.

You can also mix types and values, or multiple types, in a list. For instance `{ blah: [Number, null] }` would allow a value to be set to either a Number or null. When types are ordered, this implies a preference, and the first type that can be used to properly interpret the value will be used.

To define a new type, add it to `nopt.typeDefs`. Each item in that hash is an object with a `type` member and a `validate` method. The `type` member is an object that matches what goes in the type list. The `validate` method is a function that gets called with `validate(data, key, val)`. Validate methods should assign `data[key]` to the valid value of `val` if it can be handled properly, or return boolean `false` if it cannot.

You can also call `nopt.clean(data, types, typeDefs)` to clean up a config object and remove its invalid properties.

## Error Handling

By default, nopt outputs a warning to standard error when invalid values for known options are found. You can change this behavior by assigning a method to `nopt.invalidHandler`. This method will be called with the offending `nopt.invalidHandler(key, val, types)`.

If no `nopt.invalidHandler` is assigned, then it will console.error its whining. If it is assigned to boolean `false` then the warning is suppressed.

## Abbreviations

Yes, they are supported. If you define options like this:
```javascript
{
  "foolhardyelephants": Boolean,
  "pileofmonkeys": Boolean
}
```

Then this will work:

```bash
node program.js --foolhar --pil
node program.js --no-f --pileofmon
# etc.
```

## Shorthands

Shorthands are a hash of shorter option names to a snippet of args that they expand to.

If multiple one-character shorthands are all combined, and the combination does not unambiguously match any other option or shorthand, then they will be broken up into their constituent parts. For example:

```json
{
  "s": ["--loglevel", "silent"],
  "g": "--global",
  "f": "--force",
  "p": "--parseable",
  "l": "--long"
}
```

```bash
npm ls -sgflp
# just like doing this:
npm ls --loglevel silent --global --force --long --parseable
```

## The Rest of the args

The config object returned by nopt is given a special member called `argv`, which is an object with the following fields:

* `remain`: The remaining args after all the parsing has occurred.
* `original`: The args as they originally appeared.
* `cooked`: The args after flags and shorthands are expanded.

## Slicing
Node programs are called with more or less the exact argv as it appears in C land, after the v8 and node-specific options have been plucked off. As such, `argv[0]` is always `node` and `argv[1]` is always the JavaScript program being run.

That's usually not very useful to you. So they're sliced off by default. If you want them, then you can pass in `0` as the last argument, or any other number that you'd like to slice off the start of the list.

MIT License

Copyright (c) 2016, 2018 Linus Unnebck

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2012 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2013 Thorsten Lorenz.
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**libjpeg-turbo Licenses**

libjpeg-turbo is covered by three compatible BSD-style open source licenses:

- The IJG (Independent JPEG Group) License, which is listed in [README.ijg](README.ijg) (readme.ijg)

  This license applies to the libjpeg API library and associated programs (any code inherited from libjpeg, and any modifications to that code.)

- The Modified (3-clause) BSD License, which is listed below

  This license covers the TurboJPEG API library and associated programs, as well as the build system.

- The [zlib License](https://opensource.org/licenses/Zlib)
This license is a subset of the other two, and it covers the libjpeg-turbo SIMD extensions.

Complying with the libjpeg-turbo Licenses
========================================

This section provides a roll-up of the libjpeg-turbo licensing terms, to the best of our understanding.

1. If you are distributing a modified version of the libjpeg-turbo source, then:

   1. You cannot alter or remove any existing copyright or license notices from the source.

   **Origin**
   - Clause 1 of the IJG License
   - Clause 1 of the Modified BSD License
   - Clauses 1 and 3 of the zlib License

   2. You must add your own copyright notice to the header of each source file you modified, so others can tell that you modified that file (if there is not an existing copyright header in that file, then you can simply add a notice stating that you modified the file.)

   **Origin**
   - Clause 1 of the IJG License
   - Clause 2 of the zlib License

   3. You must include the IJG README file, and you must not alter any of the copyright or license text in that file.

   **Origin**
   - Clause 1 of the IJG License

2. If you are distributing only libjpeg-turbo binaries without the source, or if you are distributing an application that statically links with libjpeg-turbo, then:

   1. Your product documentation must include a message stating:

   This software is based in part on the work of the Independent JPEG Group.

   **Origin**
   - Clause 2 of the IJG license
2. If your binary distribution includes or uses the TurboJPEG API, then your product documentation must include the text of the Modified BSD License (see below.)

**Origin**
- Clause 2 of the Modified BSD License

3. You cannot use the name of the IJG or The libjpeg-turbo Project or the contributors thereof in advertising, publicity, etc.

**Origin**
- IJG License
- Clause 3 of the Modified BSD License

4. The IJG and The libjpeg-turbo Project do not warrant libjpeg-turbo to be free of defects, nor do we accept any liability for undesirable consequences resulting from your use of the software.

**Origin**
- IJG License
- Modified BSD License
- zlib License

The Modified (3-clause) BSD License
===================================

Copyright (C)2009-2019 D. R. Commander. All Rights Reserved.
Copyright (C)2015 Viktor Szathmry. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the libjpeg-turbo Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS", AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

Why Three Licenses?
===================

The zlib License could have been used instead of the Modified (3-clause) BSD
License, and since the IJG License effectively subsumes the distribution
conditions of the zlib License, this would have effectively placed
libjpeg-turbo binary distributions under the IJG License. However, the IJG
License specifically refers to the Independent JPEG Group and does not extend
attribution and endorsement protections to other entities. Thus, it was
desirable to choose a license that granted us the same protections for new code
that were granted to the IJG for code derived from their software.

// Copyright 2018 The ANGLE Project Authors.
// All rights reserved.
/
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//
// Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
// with the distribution.
//
// Neither the name of TransGaming Inc., Google Inc., 3DLabs Inc.
// Ltd., nor the names of their contributors may be used to endorse
// or promote products derived from this software without specific
// prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
// FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
// COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
// BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
// LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
// CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
// LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
Open Source Used In DNAC Platform Antelope 12561
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2014-present Olivier Lalonde <olalonde@gmail.com>, James Talmage <james@talmage.io>, Ruben Verborgh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2017 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 hemanth

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2011-2018 JS Foundation and contributors, https://js.foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
/* Copyright (c) 2008-2009, Google Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 * met:
 * *
 * * Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * * Neither the name of Google Inc. nor the names of its
 * contributors may be used to endorse or promote products derived from
 * this software without specific prior written permission.
 * *
 * * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 * A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 * OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
* ---
* Author: Kostya Serebryany
* /

Copyright 2011 Kevin Burke unless otherwise noted.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some content is copyrighted by Twitter, Inc., and also released under an
Apache License; these sections are noted in the source.
# This is the official list of people who can contribute
# (and who have contributed) code to the ANGLE project
# repository.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#

TransGaming Inc.
Nicolas Capens
Daniel Koch
Geoff Lang
Andrew Lewycky
Jamie Madill
Gavriel State
Shannon Woods

Google Inc.
Brent Austin
Michael Bai
John Bauman
Peter Beverloo
Steve Block
Rachel Blum
Eric Boren
Henry Bridge
Nat Duca
Peter Kasting
Vangelis Kokkevis
Zhenyao Mo
Daniel Nicoara
Alastair Patrick
Alok Priyadarshi
Kenneth Russell
Brian Salomon
Gregg Tavares
Jeff Timanus
Ben Vanik
Adrienne Walker
thestig@chromium.org
Justin Schuh
Scott Graham
Corentin Wallez
Kai Ninomiya

Adobe Systems Inc.
Alexandru Chiculita
Steve Minns
Max Vujovic

Autodesk, Inc.
Ranger Harke

Cloud Party, Inc.
Conor Dickinson

The Qt Company Ltd.
Andrew Knight

Imagination Technologies Ltd.
Gregoire Payen de La Garanderie

Intel Corporation
Jin Yang
Andy Chen
Josh Triplett
Sudarsana Nagineni
Jiajia Qin
Jiawei Shao
Jie Chen
Qiankun Miao
Bryan Bernhart
Yunchao He
Xinghua Cao
Brandon Jones

Klarlvdalens Datakonsult AB
Milian Wolff

Mozilla Corp.
Ehsan Akhgari
Edwin Flores
Jeff Gilbert
Mike Hommey
Benoit Jacob
Makoto Kato
Vladimir Vukicevic

Turbulenz
Michael Braithwaite

Ulrik Persson (ddefrostt)
Mark Banner (standard8mbp)
David Kilzer
Jacek Caban
Tibor den Ouden
Rgis Fnon
Sebastian Bergstein
James Ross-Gowan
Andrei Volykhin
Jrme Duval

Microsoft Corporation
Cooper Partin
Austin Kinross
Minmin Gong
Shawn Hargreaves

Microsoft Open Technologies, Inc.
Cooper Partin
Austin Kinross

NVIDIA Corporation
Olli Etuaho
Arun Patole
Qingqing Deng
Kimmo Kinnunen
Sami Visnen
Martin Radev

Open Source Used In DNAC Platform Antelope 12567
Open Source Used In DNAC Platform Antelope 12568

Opera Software ASA
Daniel Bratell
Tomasz Moniuszko
David Landell

Advanced Micro Devices, Inc.
Russ Lind
The MIT License (MIT)

Copyright (c) 2014 Arnout Kazemier

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2013 Linux Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the Linux Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This BSD License conforms to the Open Source Initiative "Simplified
BSD License" as published at:
http://www.opensource.org/licenses/bsd-license.php
License information regarding the data obtained from National Land Survey of
Finland http://www.maanmittauslaitos.fi/en
- topographic model from Elevation model 2 m (U4421B, U4421D, U4422A and
U4422C) 08/2014
- map image extracted from Topographic map raster 1:50 000 (U442) 08/2014

National Land Survey open data licence - version 1.0 - 1 May 2012

1. General information

The National Land Survey of Finland (hereinafter the Licensor), as the holder
of the immaterial rights to the data, has granted on the terms mentioned below
the right to use a copy (hereinafter data or dataset(s)) of the data (or a part
of it).

The Licensee is a natural or legal person who makes use of the data covered by
this licence. The Licensee accepts the terms of this licence by receiving the
dataset(s) covered by the licence.

This Licence agreement does not create a co-operation or business relationship
between the Licensee and the Licensor.

2. Terms of the licence

2.1. Right of use

This licence grants a worldwide, free of charge and irrevocable parallel right
of use to open data. According to the terms of the licence, data received by
the Licensee can be freely:
- copied, distributed and published,
- modified and utilised commercially and non-commercially,
- inserted into other products and
- used as a part of a software application or service.

2.2. Duties and responsibilities of the Licensee

Through reasonable means suitable to the distribution medium or method which is used in conjunction with a product containing data or a service utilising data covered by this licence or while distributing data, the Licensee shall:
- mention the name of the Licensor, the name of the dataset(s) and the time when the National Land Survey has delivered the dataset(s) (e.g.: contains data from the National Land Survey of Finland Topographic Database 06/2012)
- provide a copy of this licence or a link to it, as well as
- require third parties to provide the same information when granting rights to copies of dataset(s) or products and services containing such data and
- remove the name of the Licensor from the product or service, if required to do so by the Licensor.

The terms of this licence do not allow the Licensee to state in conjunction with the use of dataset(s) that the Licensor supports or recommends such use.

2.3. Duties and responsibilities of the Licensor

The Licensor shall ensure that
- the Licensor has the right to grant rights to the dataset(s) in accordance with this licence.

The data has been licensed "as is" and the Licensor
- shall not be held responsible for any errors or omissions in the data,
  disclaims any warranty for the validity or up to date status of the data and shall be free from liability for direct or consequential damages arising from the use of data provided by the Licensor,
- and is not obligated to ensure the continuous availability of the data, nor to announce in advance the interruption or cessation of availability, and the Licensor shall be free from liability for direct or consequential damages arising from any such interruption or cessation.

3. Jurisdiction

Finnish law shall apply to this licence.

4. Changes to this licence

The Licensor may at any time change the terms of the licence or apply a different licence to the data. The terms of this licence shall, however, still apply to such data that has been received prior to the change of the terms of the licence or the licence itself.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

Open Source Used In DNAC Platform Antelope  12577
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright 2011 The LibYuv Project Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright 2014 PDFium Authors. All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and
Copyright (C) 2014-2017 by Vitaly Puzrin and Andrei Tuputcy

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:

    May you do good and not evil.
    May you find forgiveness for yourself and forgive others.
    May you share freely, never taking more than you give.

// Copyright 2014 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

// This file has been auto-generated from the Jinja2 template
// {{jinja_template_filename}}
// by the script {{code_generator}}.
// DO NOT MODIFY!

// clang-format off
COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2011, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

The MIT License (MIT)

Copyright (c) 2015 Daniel Cousens

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Bodymovin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*******************************************************************************
# License headers for subpackages
*******************************************************************************

Transformation Matrix v2.0
(c) Epistemex 2014-2015
www.epistemex.com
By Ken Fyrstenberg
Contributions by leoniya.
License: MIT, header required.

*******************************************************************************

Copyright 2014 David Bau.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. 
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY 
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, 
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE 
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

 Bez:str: Easing - use bezier curve for transition easing function 
by Gatan Renaudeau 2014 - 2015  MIT License

Credits: is based on Firefox's nsSMILKeySpline.cpp 
Usage: 
var spline = BezierEasing([ 0.25, 0.1, 0.25, 1.0 ]); 
spline.get(x) => returns the easing value | x must be in [0, 1] range 
Copyright (c) 2011 IETF Trust and the persons identified as 
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or 
without modification, are permitted provided that the following 
conditions are met:

- Redistributions of source code must retain the above 
copyright notice, this list of conditions and 
the following disclaimer.

- Redistributions in binary form must reproduce the above 
copyright notice, this list of conditions and the following 
disclaimer in the documentation and/or other materials provided 
with the distribution.

- Neither the name of Internet Society, IETF or IETF Trust, nor 
the names of specific contributors, may be used to endorse or 
promote products derived from this software without specific 
prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND 
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, 
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2015-2019 The Khronos Group Inc.
Copyright (c) 2015-2019 LunarG, Inc.
Copyright (c) 2015-2019 Valve Corporation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on as "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Unless otherwise noted in the LICENSE.txt file in the install folder, all components of the Vulkan Runtime are licensed under the above license. Licenses for any components not so licensed are listed in the LICENSE.txt file.

==================================================================================================
==================================================================================================

Copyright (c) 2009 Dave Gamble
Copyright (c) 2015-2016 The Khronos Group Inc.
Copyright (c) 2015-2016 Valve Corporation
Copyright (c) 2015-2016 LunarG, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

==================================================================================================================================================================================================================

Copyright (c) 2014 joseph werle <joseph.werle@gmail.com>
Copyright (c) 2015-2016 The Khronos Group Inc.
Copyright (c) 2015-2016 Valve Corporation
Copyright (c) 2015-2016 LunarG, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions:

The above copyright notice(s) and this permission notice shall be included in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.

Copyright (c) 2014, Hugh Kennedy
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the nor the names of its contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2014-2017 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2010, The WebM Project authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of Google, nor the WebM Project, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/*
 * Copyright (c) 2008 NVIDIA, Corporation
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the Software is
 * furnished to do so, subject to the following conditions:
 * */
* The above copyright notice and this permission notice (including the next
* paragraph) shall be included in all copies or substantial portions of the
* Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
* AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
* OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*/

KWSys - Kitware System Library
Copyright 2000-2009 Kitware, Inc., Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

* Neither the names of Kitware, Inc., the Insight Software Consortium,
nor the names of their contributors may be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2010 by Armin Ronacher and contributors. See AUTHORS
for more details.

Some rights reserved.
Redistribution and use in source and binary forms of the software as well as documentation, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* The names of the contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE AND DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

CMake - Cross Platform Makefile Generator
Copyright 2000-2018 Kitware, Inc. and Contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Kitware, Inc. nor the names of Contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following individuals and institutions are among the Contributors:

* Aaron C. Meadows <cmake@shadowguarddev.com>
* Adriaan de Groot <groot@kde.org>
* Aleksey Avdeev <solo@altlinux.ru>
* Alexander Neundorf <neundorf@kde.org>
* Alexander Smorkalov <alexander.smorkalov@itseez.com>
* Alexey Sokolov <sokolov@gmail.com>
* Alex Turbov <i.zaufi@gmail.com>
* Andreas Pakulat <apaku@gmx.de>
* Andreas Schneider <asn@cryptomilk.org>
* Andr Rigland Brodtkorb <Andre.Brodtkorb@ifi.uio.no>
* Axel Huebl, Helmholtz-Zentrum Dresden - Rossendorf
* Benjamin Eikel
* Bjoern Ricks <bjoern.ricks@gmail.com>
* Brad Hards <bradh@kde.org>
* Christopher Harvey
* Christoph Grninger <foss@grueninger.de>
* Clement Creusot <creusot@es.york.ac.uk>
* Daniel Blezek <blezek@gmail.com>
* Daniel Pfeifer <daniel@pfeifer-mail.de>
* Enrico Scholz <enrico.scholz@informatik.tu-chemnitz.de>
* Eran Ifrah <eran.ifrah@gmail.com>
* Esben Mose Hansen, Ange Optimization ApS
* Geoffrey Viola <geoffrey.viola@asirobots.com>
* Google Inc
* Gregor Jasny
* Helio Chissini de Castro <helio@kde.org>
* Ilya Lavrenov <ilya.lavrenov@itseez.com>
* Insight Software Consortium <insightsoftwareconsortium.org>
* Jan Woetzeli
* Kelly Thompson <kgt@lanl.gov>
* Konstantin Podsvirov <konstantin@podsvirov.pro>
* Mario Bensi <mbensi@ipsquad.net>
* Mathieu Malaterre <mathieu.malaterre@gmail.com>
See version control history for details of individual contributions.

The above copyright and license notice applies to distributions of CMake in source and binary form. Third-party software packages supplied with CMake under compatible licenses provide their own copyright notices documented in corresponding subdirectories or source files.

CMake was initially developed by Kitware with the following sponsorship:

* National Library of Medicine at the National Institutes of Health
as part of the Insight Segmentation and Registration Toolkit (ITK).

* US National Labs (Los Alamos, Livermore, Sandia) ASC Parallel Visualization Initiative.

* National Alliance for Medical Image Computing (NAMIC) is funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149.

* Kitware, Inc.

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2009, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

MIT License

Copyright (c) 2014 Petka Antonov
2015 Sindre Sorhus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [2007] Neal Norwitz
provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER(S) AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER(S) OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Note: The license for SPIRV-Tools changed to Apache 2.0 via this commit: https://github.com/KhronosGroup/SPIRV-Tools/commit/9fc8658ef301b0f03b2173d274c52f011b5c73e5

// Copyright (c) 2015-2016 The Khronos Group Inc.
// Copyright (c) 2015-2016 Google Inc.
// Copyright (c) 2016 LunarG Inc.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

****************************************************************************
** Copyright (C) YYYY The Qt Company Ltd.
** Contact: https://www.qt.io/licensing/
**
** This file is part of the FOO module of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:GPL-EXCEPT$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms

Open Source Used In DNAC Platform Antelope 12607
** and conditions see https://www.qt.io/terms-conditions. For further
** information use the contact form at https://www.qt.io/contact-us.
**
** GNU General Public License Usage
** Alternatively, this file may be used under the terms of the GNU
** General Public License version 3 as published by the Free Software
** Foundation with exceptions as appearing in the file LICENSE.GPL3-EXCEPT
** included in the packaging of this file. Please review the following
** information to ensure the GNU General Public License requirements will
** be met: https://www.gnu.org/licenses/gpl-3.0.html.
**
** $QT_END_LICENSE$
**
****************************************************************************/

The MIT License (MIT)

Copyright (c) 2014 Cory Benfield

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
(Copied from the README.)

--------------------------------------------------------------------------

LICENSE extracted from IJG's jpeg distribution:
--------------------------------------------------------------------------

In plain English:

1. We don't promise that this software works. (But if you find any bugs,
   please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a
program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:
(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

GNU GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2015 The Qt Company Ltd. Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt Toolkit under the terms of GNU General Public License version 2, which is displayed below.

-------------------------------------------------------------------------
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this
license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom
to share and change it. By contrast, the GNU General Public License is
intended to guarantee your freedom to share and change free software
--to make sure the software is free for all its users. This General
Public License applies to most of the Free Software Foundation's
software and to any other program whose authors commit to using it.
(Some other Free Software Foundation software is covered by the GNU
Lesser General Public License instead.) You can apply it to your
programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have the
freedom to distribute copies of free software (and charge for this
service if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone
to deny you these rights or to ask you to surrender the rights. These
restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis
or for a fee, you must give the recipients all the rights that you
have. You must make sure that they, too, receive or can get the source
code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original

Open Source Used In DNAC Platform Antelope 12610
Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further restrictions
on the recipients’ exercise of the rights granted herein. You are not
responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent license
would not permit royalty-free redistribution of the Program by all
those who receive copies directly or indirectly through you, then the
only way you could satisfy both it and this License would be to refrain
entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License may
add an explicit geographical distribution limitation excluding those
countries, so that distribution is permitted only in or among countries
not thus excluded. In such case, this License incorporates the limitation
as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail
to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
MIT License

Copyright (c) 2018 Nikita Skovoroda <chalkerx@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

reporting:
   reports:
- lcov
Copyright Joyent, Inc. and other Node contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Xiph.org Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ISC License (ISC)
Copyright 2018 Stefan Penner

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1996 John Maloney and Mario Wolczko
Copyright 2003-2005 Tom Wu
Copyright 2005-2007 Sam Stephenson
Copyright 2006-2010 the V8 project authors

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

raytrace.js contains code from the the Prototype JavaScript framework, version 1.5.0:

Prototype JavaScript framework, version 1.5.0
Copyright (c) 2005-2007 Sam Stephenson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The majority of files in this project use the Apache 2.0 License. There are a few exceptions and their license can be found in the source. Any license deviations from Apache 2.0 are "more permissive" licenses.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE file from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

## -*-makefile-*-
## Copyright (C) 2016 and later: Unicode, Inc. and others.
## License & terms of use: http://www.unicode.org/copyright.html
## BSD-specific setup (FreeBSD, OpenBSD, NetBSD, *BSD)
## Copyright (c) 1999-2013, International Business Machines Corporation and
## others. All Rights Reserved.

## Commands to generate dependency files
GEN_DEPS.c=$(CC) -E -MM $(DEFS) $(CPPFLAGS)
GEN_DEPS.cc=$(CXX) -E -MM $(DEFS) $(CPPFLAGS)

## Flags for position independent code
SHAREDLIBCFLAGS = -fPIC
SHAREDLIBCXXFLAGS = -fPIC
SHAREDLIBCPPFLAGS = -DPIC

## Additional flags when building libraries and with threads
THREADSCPPFLAGS = -D_REENTRANT
LIBCPPFLAGS =

## Compiler switch to embed a runtime search path
LD_RPATH=
LD_RPATH_PRE=-Wl,-rpath,

## Compiler switch to embed a library name
LD_SONAME = -Wl,-soname -Wl,$(notdir $(MIDDLE_SO_TARGET))

## Shared library options
LD_SOOPTIONS= -Wl,-Bsymbolic

## Shared object suffix
SO = so

## Non-shared intermediate object suffix
STATIC_O = ao

## Compilation rules
%.S(STATIC_O): $(srcdir)/%.c
$(COMPILE.c) $(STATICCPPFLAGS) $(STATICCFLAGS) -o $@ $<
%.o: $(srcdir)/%.c
$(COMPILE.c) $(DYNAMICCPPFLAGS) $(DYNAMICCFLAGS) -o $@ $<

%.S(STATIC_O): $(srcdir)/%.cpp
$(COMPILE.cc) $(STATICCPPFLAGS) $(STATICCXXFLAGS) -o $@ $<
%.o: $(srcdir)/%.cpp
$(COMPILE.cc) $(DYNAMICCPPFLAGS) $(DYNAMICCXXFLAGS) -o $@ $<

## Dependency rules
%.d: $(srcdir)/%.c
@echo "generating dependency information for $<"
@$($(SHELL)) -ec "$(GEN_DEPS.c) $< \n | sed "s/\(\*\)\.o[ :]\+/\1.o $@ : /g" > $@; \n [ -s $@ ] || rm -f $@"

%.d: $(srcdir)/%.cpp
@echo "generating dependency information for $<"
@$($(SHELL)) -ec "$(GEN_DEPS.cc) $< \n | sed "s/\(\*\)\.o[ :]\+/\1.o $@ : /g" > $@; \n [ -s $@ ] || rm -f $@"

## Versioned libraries rules
%.S(SO).$(SO_TARGET_VERSION_MAJOR): %.S(SO).$(SO_TARGET_VERSION)
$(RM) $@ && ln -s ${<F} $@
%.S(SO): %.S(SO).$(SO_TARGET_VERSION_MAJOR)
$(RM) $@ && ln -s ${*F}.$(SO).$(SO_TARGET_VERSION) $@

## Bind internal references

# LDflags that pkgdata will use
BIR_LDFLAGS= -Wl,-Bsymbolic

# Dependencies [i.e. map files] for the final library
BIR_DEPS=

## Remove shared library 's'
STATIC_PREFIX_WHEN_USED =
STATIC_PREFIX =

## End BSD-specific setup
Copyright 2019 Inria
Author: Nicolas Roussel (nicolas.roussel@inria.fr)

BSD License https://opensource.org/licenses/BSD-3-Clause
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holders, nor those of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2017 Intel Corporation
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// Copyright (c) 2012 The Polymer Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(Copied from src/license.txt)
The source code of UnRAR utility is freeware. This means:

1. All copyrights to RAR and the utility UnRAR are exclusively owned by the author - Alexander Roshal.

2. UnRAR source code may be used in any software to handle RAR archives without limitations free of charge, but cannot be used to develop RAR (WinRAR) compatible archiver and to re-create RAR compression algorithm, which is proprietary. Distribution of modified UnRAR source code in separate form or as a part of other software is permitted, provided that full text of this paragraph, starting from "UnRAR source code" words, is included in license, or in documentation if license is not available, and in source code comments of resulting package.

3. The UnRAR utility may be freely distributed. It is allowed to distribute UnRAR inside of other software packages.

4. THE RAR ARCHIVER AND THE UnRAR UTILITY ARE DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.

5. Installing and using the UnRAR utility signifies acceptance of these terms and conditions of the license.

6. If you don't agree with terms of the license you must remove UnRAR files from your storage devices and cease to use the utility.

Thank you for your interest in RAR and UnRAR.

Alexander L. Roshal

// Copyright 2014 The Chromium OS Authors. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
// *
// Redistributions of source code must retain the above copyright
This open source software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

**Copyright (c) Rod Vagg (the "Original Author") and additional contributors**

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

// Copyright (c) 2005, Google Inc.
// All rights reserved.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
//   * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2000  Ming-Che Chuang
Copyright (C) 2001, 2002  James Su, Turbolinux Inc.
Copyright (C) 2002  WU Yi, HancomLinux Inc.
Copyright (C) 2001, 2002  Anthony Fok, ThizLinux Laboratory Ltd.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

MIT License

Copyright (c) DC <threedeecce@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2012 Irakli Gozalishvili. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

GNU GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2015 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt GUI Toolkit under the terms of
GNU Lesser General Public License version 3. That license references
the General Public License version 3, that is displayed below. Other
portions of the Qt Toolkit may be licensed directly under this license.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of
protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding
The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.
6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.
A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for
unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further
restrictions” within the meaning of section 10. If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of
this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.
To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program>  Copyright (C) <year>  <name of author>  
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.

For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

#!/usr/bin/env node
var nopt = require("../lib/nopt")
```javascript
var path = require("path");
var types = {
  num: Number,
  bool: Boolean,
  help: Boolean,
  list: Array,
  "num-list": [Number, Array],
  "str-list": [String, Array],
  "bool-list": [Boolean, Array],
  str: String,
  clear: Boolean,
  config: Boolean,
  length: Number,
  file: path
};

var shorthands = {
  s: [ "--str", "astring" ],
  b: [ "--bool" ],
  nb: [ "--no-bool" ],
  tft: [ "--bool-list", "--no-bool-list", "--bool-list", "true" ],
  "?": ["--help"],
  h: ["--help"],
  H: ["--help"],
  n: [ "--num", "125" ],
  c: ["--config"],
  l: ["--length"],
  f: ["--file"]
};

var parsed = nopt( types, shorthands, process.argv, 2 );

console.log("parsed", parsed);

if (parsed.help) {
  console.log(""
  console.log("nopt cli tester")
  console.log(""
  console.log("types")
  console.log(Object.keys(types).map(function M (t) {
    var type = types[t];
    if (Array.isArray(type)) {
      return [t, type.map(function (type) { return type.name })]
    } else {
      return [t, type && type.name]
    }
  }).reduce(function (s, i) {
    s[i[0]] = i[1]
    return s
  }, {}));
```
The "Clarified Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Distribution fee" is a fee you charge for providing a copy of this Package to another party.

"Freely Available" means that no fee is charged for the right to use the item, though there may be fees involved in handling the item.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain, or those made Freely Available, or from the Copyright Holder. A Package modified in such a way shall still be
considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site allowing unrestricted access to them, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

e) permit and encourage anyone who receives a copy of the modified Package permission to make your modifications Freely Available in some specific way.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

e) offer the machine-readable source of the Package, with your modifications, by mail order.

5. You may charge a distribution fee for any distribution of this Package. If you offer support for this Package, you may charge any fee you choose
for that support. You may not charge a license fee for the right to use this Package itself. You may distribute this Package in aggregate with other (possibly commercial and possibly nonfree) programs as part of a larger (possibly commercial and possibly nonfree) software distribution, and charge license fees for other parts of that software distribution, provided that you do not advertise this Package as a product of your own. If the Package includes an interpreter, You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of the Standard Version of the Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End
(Copied from the COPYRIGHT file of https://code.google.com/p/sctp-refimpl/source/browse/trunk/COPYRIGHT)
Copyright (c) 2001, 2002 Cisco Systems, Inc.
Copyright (c) 2002-12 Randall R. Stewart
Copyright (c) 2002-12 Michael Tuexen
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

C++ Big Integer Library
(see ChangeLog for version)

http://mattmccutchen.net/bigint/

Written and maintained by Matt McCutchen <matt@mattmccutchen.net>

You can use this library in a C++ program to do arithmetic on integers of size limited only by your computer's memory. The library provides BigUnsigned and BigInteger classes that represent nonnegative integers and signed integers, respectively. Most of the C++ arithmetic operators are overloaded for these classes, so big-integer calculations are as easy as:

```
#include "BigIntegerLibrary.hh"

BigInteger a = 65536;
cout << (a * a * a * a * a * a * a);  
```

(prints 340282366920938463463374607431768211456)
The code in `sample.cc' demonstrates the most important features of the library. To get started quickly, read the code and explanations in that file and run it. If you want more detail or a feature not shown in `sample.cc', consult the consult the actual header and source files, which are thoroughly commented.

This library emphasizes ease of use and clarity of implementation over speed; some users will prefer GMP (http://swox.com/gmp/), which is faster. The code is intended to be reasonably portable across computers and modern C++ compilers; in particular, it uses whatever word size the computer provides (32-bit, 64-bit, or otherwise).

Compiling programs that use the library
---------------------------------------
The library consists of a folder full of C++ header files (`.hh') and source files (`.cc'). Your own programs should `#include' the necessary header files and link with the source files. A makefile that builds the sample program (`sample.cc') is included; you can adapt it to replace the sample with your own program.

Alternatively, you can use your own build system or IDE. In that case, you must put the library header files where the compiler will find them and arrange to have your program linked with the library source files; otherwise, you will get errors about missing header files or "undefined references". To learn how to do this, consult the documentation for the build system or IDE; don't bother asking me. Adding all the library files to your project will work in many IDEs but may not be the most desirable approach.

Resources
---------
The library's Web site (above) provides links to released versions, the current development version, and a mailing list for release announcements, questions, bug reports, and other discussion of the library. I would be delighted to hear from you if you like this library and/or find a good use for it.

Bugs and enhancements
-----------------------
The library has been tested by me and others but is by no means bug-free. If you find a bug, please report it, whether it comes in the form of compiling trouble, a mathematically inaccurate result, or a memory-management blooper (since I use Java, these are altogether too common in my C++). I generally fix all reported bugs. You are also welcome to request enhancements, but I am unlikely to do substantial amounts of work on enhancements at this point.

Legal
-----
I, Matt McCutchen, the sole author of the original Big Integer Library, waive my copyright to it, placing it in the public domain. The library comes with absolutely no warranty.
The MIT License (MIT)

Copyright (c) 2014-2017 createECDH contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it...
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

------------------------------------------------------------------------------

Gradle Subcomponents:

------------------------------------------------------------------------------

License for the slf4j package

------------------------------------------------------------------------------

SLF4J License
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

These terms are identical to those of the MIT License, also called the X License or the X11 License, which is a simple, permissive non-copyleft free software license. It is deemed compatible with virtually all types of licenses, commercial or otherwise. In particular, the Free Software Foundation has declared it compatible with GNU GPL. It is also known to be approved by the Apache Software Foundation as compatible with Apache Software License.

---------------------------------------------
License for the JUnit package
---------------------------------------------
THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;
where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution 'originates' from a
Contributor if it was added to the Program by such Contributor itself or anyone
acting on such Contributor's behalf. Contributions do not include additions to
the Program which: (i) are separate modules of software distributed in
conjunction with the Program under their own license agreement, and (ii) are not
derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when
combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free copyright license to
reproduce, prepare derivative works of, publicly display, publicly perform,
distribute and sublicense the Contribution of such Contributor, if any, and such
derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed
Patents to make, use, sell, offer to sell, import and otherwise transfer the
Contribution of such Contributor, if any, in source code and object code form.
This patent license shall apply to the combination of the Contribution and the
Program if, at the time the Contribution is added by the Contributor, such
addition of the Contribution causes such combination to be covered by the
Licensed Patents. The patent license shall not apply to any other combinations
which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses
to its Contributions set forth herein, no assurances are provided by any
Contributor that the Program does not infringe the patent or other intellectual
property rights of any other entity. Each Contributor disclaims any liability to
Recipient for claims brought by any other entity based on infringement of
intellectual property rights or otherwise. As a condition to exercising the
rights and licenses granted hereunder, each Recipient hereby assumes sole
responsibility to secure any other intellectual property rights needed, if any.
For example, if a third party patent license is required to allow Recipient to
distribute the Program, it is Recipient's responsibility to acquire that license.
before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS
7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

------------------------------------------------------------------------------
License for the JCIFS package
------------------------------------------------------------------------------

JCIFS License

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.
For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.
For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)
"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has
a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2011-2017 by Yehuda Katz

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

# Files that don't get a copyright, or which are taken from elsewhere.
#
# All lines in this file are patterns, including the comment lines; this
# means that e.g. `FTL.TXT' matches all files that have this string in
# the file name (including the path relative to the current directory,
# always starting with `./').
#
# Don't put empty lines into this file!
#
.gitignore
#
builds/unix/pkg.m4
#
docs/FTL.TXT
docs/GPLv2.TXT
#
include/freetype/internal/fthash.h
#
src/base/fthash.c
src/base/md5.c
src/base/md5.h
#
src/bdf/bdf.c
src/bdf/bdf.h
src/bdf/bdfdrivr.c
src/bdf/bdfdrivr.h
src/bdf/bdferror.h
src/bdf/bdflib.c
src/bdf/module.mk
src/bdf/README
src/bdf/rules.mk
#
src/pcf/module.mk
src/pcf/pcf.c
src/pcf/pcf.h
src/pcf/pcfdrivr.c
src/pcf/pcfdrivr.h
src/pcf/pcferror.h
src/pcf/pcfread.c
src/pcf/pcfread.h
src/pcf/pcfutil.c
src/pcf/pcfutil.h
src/pcf/README
src/pcf/rules.mk
#
src/gzip/adler32.c
src/gzip/infblock.c
Copyright (c) 2015 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Original Library
- Copyright (c) Marak Squires

Additional Functionality
- Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Fonts are (c) Bitstream (see below). DejaVu changes are in public domain. Glyphs imported from Arev fonts are (c) Tavmjong Bah (see below)

Bitstream Vera Fonts Copyright
--------------------------------
Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Bitstream" or the word "Vera".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Bitstream Vera" names.

The Font Software may be sold as part of a larger software package but no
copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

Arev Fonts Copyright
----------------------

Copyright (c) 2006 by Tavmjong Bah. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license ("Fonts") and associated documentation files (the "Font Software"), to reproduce and distribute the modifications to the Bitstream Vera Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words "Tavmjong Bah" or the word "Arev".

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the "Tavmjong Bah Arev" names.

The Font Software may be sold as part of a larger software package but
no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT. IN NO EVENT SHALL TAVMJONG BAH BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the name of Tavmjong Bah shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from Tavmjong Bah. For further information, contact: tavmjong @ free.fr.

TeX Gyre DJV Math
--------------
Fonts are (c) Bitstream (see below). DejaVu changes are in public domain.

Math extensions done by B. Jackowski, P. Strzelczyk and P. Pianowski (on behalf of TeX users groups) are in public domain.

Letters imported from Euler Fraktur from AMSfonts are (c) American Mathematical Society (see below).

Bitstream Vera Fonts Copyright
Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license (Fonts) and associated documentation files (the Font Software), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and
additional
glyphs or characters may be added to the Fonts, only if the fonts are
renamed
to names not containing either the words Bitstream or the word Vera.

This License becomes null and void to the extent applicable to Fonts or
Font Software
that has been modified and is distributed under the Bitstream Vera
names.

The Font Software may be sold as part of a larger software package but
no copy
of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT,
TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME
FOUNDATION
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL,
SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN
ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR
INABILITY TO USE
THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.
Except as contained in this notice, the names of GNOME, the GNOME
Foundation,
and Bitstream Inc., shall not be used in advertising or otherwise to promote
the sale, use or other dealings in this Font Software without prior written
authorization from the GNOME Foundation or Bitstream Inc., respectively.
For further information, contact: fonts at gnome dot org.

AMSFonts (v. 2.2) copyright

The PostScript Type 1 implementation of the AMSFonts produced by and
previously distributed by Blue Sky Research and Y&Y, Inc. are now freely
available for general use. This has been accomplished through the
cooperation
of a consortium of scientific publishers with Blue Sky Research and Y&Y.
Members of this consortium include:

Elsevier Science IBM Corporation Society for Industrial and Applied
Mathematics (SIAM) Springer-Verlag American Mathematical Society (AMS)

In order to assure the authenticity of these fonts, copyright will be
held by
the American Mathematical Society. This is not meant to restrict in any way
the legitimate use of the fonts, such as (but not limited to) electronic
distribution of documents containing these fonts, inclusion of these fonts into other public domain or commercial font collections or computer applications, use of the outline data to create derivative fonts and/or faces, etc. However, the AMS does require that the AMS copyright notice be removed from any derivative versions of the fonts which have been altered in any way. In addition, to ensure the fidelity of TeX documents using Computer Modern fonts, Professor Donald Knuth, creator of the Computer Modern faces, has requested that any alterations which yield different font metrics be given a different name.

$Id$
// sigslot.h: Signal/Slot classes
//
// Written by Sarah Thompson (sarah@telergy.com) 2002.
//
// License: Public domain. You are free to use this code however you like, with
// the proviso that the author takes on no responsibility or liability for any
// use.
Copyright (c) 2006, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2011, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.

Copyright (c) 1999 Apple Computer, Inc. All rights reserved.

@APPLE_LICENSE_HEADER_START@

This file contains Original Code and/or Modifications of Original Code
as defined in and that are subject to the Apple Public Source License
Version 2.0 (the 'License'). You may not use this file except in
compliance with the License. Please obtain a copy of the License at
http://www.opensource.apple.com/apsl/ and read it before using this
file.

The Original Code and all software distributed under the License are
distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER
EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES,
INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT.
Please see the License for the specific language governing rights and
limitations under the License.

@APPLE_LICENSE_HEADER_END@

Copyright 2007-2008 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

(MIT)

Copyright (c) 2013 Julian Gruber <julian@juliangruber.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

All files in this directory are provided by the following license if not stated otherwise in the individual file:

=================================
This file is provided under a dual BSD/GPLv2 license. When using or redistributing this file, you may do so under either license.

GPL LICENSE SUMMARY

Copyright (c) 2005-2012 Intel Corporation. All rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St - Fifth Floor, Boston, MA 02110-1301 USA. The full GNU General Public License is included in this distribution in the file called LICENSE.GPL.

Contact Information:

BSD LICENSE

Copyright (c) 2005-2012 Intel Corporation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

================================
Copyright (c) 1985, 1986 The Regents of the University of California.
All rights reserved.

This code is derived from software contributed to Berkeley by James A. Woods, derived from original work by Spencer Thomas and Joseph Orost.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
   This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ISC License

Copyright (c) 2017, Mapbox

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

JSZip is dual licensed. You may use it under the MIT license *or* the GPLv3 license.

The MIT License
Copyright (c) 2009-2012 Stuart Knightley, David Duponchel, Franz Buchinger, Antnio Afonso

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GPL version 3

==

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.
TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the
product that is covered by this License, on a durable physical 
medium customarily used for software interchange, for a price no 
more than your reasonable cost of physically performing this 
conveying of source, or (2) access to copy the 
Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the 
written offer to provide the Corresponding Source. This 
alternative is allowed only occasionally and noncommercially, and 
only if you received the object code with such an offer, in accord 
with subsection 6b.

d) Convey the object code by offering access from a designated 
place (gratis or for a charge), and offer equivalent access to the 
Corresponding Source in the same way through the same place at no 
further charge. You need not require recipients to copy the 
Corresponding Source along with the object code. If the place to 
copy the object code is a network server, the Corresponding Source 
may be on a different server (operated by you or a third party) 
that supports equivalent copying facilities, provided you maintain 
clear directions next to the object code saying where to find the 
Corresponding Source. Regardless of what server hosts the 
Corresponding Source, you remain obligated to ensure that it is 
available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided 
you inform other peers where the object code and Corresponding 
Source of the work are being offered to the general public at no 
charge under subsection 6d.

A separable portion of the object code, whose source code is excluded 
from the Corresponding Source as a System Library, need not be 
included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any 
tangible personal property which is normally used for personal, family, 
or household purposes, or (2) anything designed or sold for incorporation 
into a dwelling. In determining whether a product is a consumer product, 
doubtful cases shall be resolved in favor of coverage. For a particular 
product received by a particular user, "normally used" refers to a 
typical or common use of that class of product, regardless of the status 
of the particular user or of the way in which the particular user 
actually uses, or expects or is expected to use, the product. A product 
is a consumer product regardless of whether the product has substantial 
commercial, industrial or non-consumer uses, unless such uses represent 
the only significant mode of use of the product.

"Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the
terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or
author attributions in that material or in the Appropriate Legal
Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or
requiring that modified versions of such material be marked in
reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or
authors of the material; or

e) Declining to grant rights under trademark law for use of some
trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that
material by anyone who conveys the material (or modified versions of
it) with contractual assumptions of liability to the recipient, for
any liability that these contractual assumptions directly impose on
those licensors and authors.

All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10. If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term. If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.
8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an
organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.
13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

NASM is now licensed under the 2-clause BSD license, also known as the simplified BSD license.

Copyright 1996-2010 the NASM Authors - All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

cubemaps: See cubemaps/miramar/README.TXT

gltf: See gltf/LICENSE.md
The "Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
The MIT License (MIT)
Copyright (c) 2015-2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Kevin Roark

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2015 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt GUI Toolkit under the terms of GNU Lesser General Public License version 3, which is displayed below.

-------------------------------------------------------------------------
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this
licensedocument, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, this License refers to version 3 of the GNU Lesser
General Public License, and the GNU GPL refers to version 3 of the
GNU General Public License.

The Library refers to a covered work governed by this License,
another than an Application or a Combined Work as defined below.

An Application is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A Combined Work is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the Linked
Version.

The Minimal Corresponding Source for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The Corresponding Application Code for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.
If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.
The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License or any later version applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Copyright 2006-2011, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.
Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jlop@gzip.org           madler@alumni.caltech.edu

If you use the zlib library in a product, we would appreciate *not* receiving lengthy legal documents to sign. The sources are provided for free but without warranty of any kind. The library has been entirely written by Jean-loup Gailly and Mark Adler; it does not include third-party code.

If you redistribute modified sources, we would appreciate that you include in the file ChangeLog history information documenting your changes. Please read the FAQ for more information on the distribution of modified source versions.

The MIT License (MIT)

Copyright (c) 2014-2017 browserify-aes contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2010
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# This file contains a list of people who've made non-trivial
# contribution to the Google C++ Mocking Framework project. People
# who commit code to the project are encouraged to add their names
# here. Please keep the list sorted by first names.

Benoit Sigoure <tsuna@google.com>
Bogdan Piloca <boo@google.com>
Chandler Carruth <chandlerc@google.com>
Dave MacLachlan <dmaclach@gmail.com>
David Anderson <danderson@google.com>
Dean Sturtevant
Gene Volovich <gv@cite.com>
Hal Burch <gmock@hburch.com>
Jeffrey Yasskin <jyasskin@google.com>
Jim Keller <jimkeller@google.com>
Joe Walnes <joe@truemesh.com>
Jon Wray <jwray@google.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kostya Serebryany <kcc@google.com>
The Qt Company GPL Exception 1.0

Exception 1:

As a special exception you may create a larger work which contains the output of this application and distribute that work under terms of your choice, so long as the work is not otherwise derived from or based on this application and so long as the work does not in itself generate output that contains the output from this application in its original or modified form.

Exception 2:

As a special exception, you have permission to combine this application with Plugins licensed under the terms of your choice, to produce an executable, and to copy and distribute the resulting executable under the terms of your choice. However, the executable must be accompanied by a prominent notice offering all users of the executable the entire source code to this application, excluding the source code of the independent modules, but including any changes you have made to this application, under the terms of this license.
GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a
computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-mercial terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that
material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under
this License. If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.


You are not required to accept this License in order to receive or
run a copy of the Program. Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance. However,
nothing other than this License grants you permission to propagate or
modify any covered work. These actions infringe copyright if you do
not accept this License. Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License. You are not responsible
for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations. If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License. For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based. The
work thus licensed is called the contributor's "contributor version”.

A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software
Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License. But first, please read

Copyright (c) 1998-2013, Brian Gladman, Worcester, UK. All rights reserved.

The redistribution and use of this software (with or without changes)
is allowed without the payment of fees or royalties provided that:

source code distributions include the above copyright notice, this list of conditions and the following disclaimer;

binary distributions include the above copyright notice, this list of conditions and the following disclaimer in their documentation.

This software is provided ‘as is’ with no explicit or implied warranties
in respect of its operation, including, but not limited to, correctness
and fitness for purpose.

GNU LESSER GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2017 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt Toolkit under the terms of
GNU Lesser General Public License version 3, which is displayed below.
This license makes reference to the version 3 of the GNU General
Public License, which you can find in the LICENSE.GPLv3 file.

-------------------------------------------------------------------------

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this licencedocument, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, this License refers to version 3 of the GNU Lesser
General Public License, and the GNU GPL refers to version 3 of the
GNU General Public License.

The Library refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An Application is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A Combined Work is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the Linked Version.

The Minimal Corresponding Source for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The Corresponding Application Code for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this
license document.


You may convey a Combined Work under terms of your choice that, taken
together, effectively do not restrict modification of the portions of
the Library contained in the Combined Work and reverse engineering for
debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this
license document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

  0) Convey the Minimal Corresponding Source under the terms of
this License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

  1) Use a suitable shared library mechanism for linking with
the Library. A suitable mechanism is one that (a) uses at run
time a copy of the Library already present on the user's
computer system, and (b) will operate properly with a modified
version of the Library that is interface-compatible with the
Linked Version.

e) Provide Installation Information, but only if you would
otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License or any later version applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
GNU GENERAL PUBLIC LICENSE

The Qt Toolkit is Copyright (C) 2016 The Qt Company Ltd.
Contact: http://www.qt.io/licensing/

You may use, distribute and copy the Qt GUI Toolkit under the terms of GNU Lesser General Public License version 3. That license references the General Public License version 3, that is displayed below. Other portions of the Qt Toolkit may be licensed directly under this license.

-------------------------------------------------------------------------

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so. This is fundamentally incompatible with the aim of
protecting users' freedom to change the software. The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable. Therefore, we
have designed this version of the GPL to prohibit the practice for those
products. If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary. To prevent this, the GPL assures that
patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and
modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this
License. Each licensee is addressed as "you". "Licensees" and
"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.
The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities. However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work. For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

The Corresponding Source for a work in source code form is that
same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met. This License explicitly affirms your unlimited
permission to run the unmodified Program. The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work. This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force. You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright. Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under
the conditions stated below. Sublicensing is not allowed; section 10
makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.
No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-imagressive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the
Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply.
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed. Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law. If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it. (Additional permissions may be written to require their own
removal in certain cases when you modify the work.) You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the
terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or
author attributions in that material or in the Appropriate Legal
Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or
requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.
Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for
11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.
The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.
If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License. But first, please read
# Rollup core license
Rollup is released under the MIT license:

The MIT License (MIT)

Copyright (c) 2017 [these people](https://github.com/rollup/rollup/graphs/contributors)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

# Licenses of bundled dependencies
The published Rollup artifact additionally contains code with the following licenses:
MIT, ISC, Apache-2.0

# Bundled dependencies:
## acorn-import-meta
License: MIT
By: Adrian Heine
Repository: https://github.com/adrianheine/acorn-import-meta
Copyright (C) 2017-2018 by Adrian Heine

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----------------------------

# acorn-walk
License: MIT
By: Marijn Haverbeke, Ingvar Stepanyan, Adrian Heine
Repository: https://github.com/acornjs/acorn.git

Copyright (C) 2012-2018 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
# braces
License: MIT
By: Jon Schlinkert, Brian Woodward, Elan Shanker, Eugene Sharygin, hemanth.hm
Repository: micromatch/braces

> The MIT License (MIT)
>
> Copyright (c) 2014-2018, Jon Schlinkert.
>
> Permission is hereby granted, free of charge, to any person obtaining a copy
> of this software and associated documentation files (the "Software"), to deal
> in the Software without restriction, including without limitation the rights
> to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
> copies of the Software, and to permit persons to whom the Software is
> furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in
> all copies or substantial portions of the Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
> AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
> LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
> OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
> THE SOFTWARE.

# date-time
License: MIT
By: Sindre Sorhus
Repository: sindresorhus/date-time

# fill-range
License: MIT
By: Jon Schlinkert, Edo Rivai, Paul Miller, Rouven Weling
Repository: jonschlinkert/fill-range

> The MIT License (MIT)
>
> Copyright (c) 2014-present, Jon Schlinkert.
>
> Permission is hereby granted, free of charge, to any person obtaining a copy
> of this software and associated documentation files (the "Software"), to deal
> in the Software without restriction, including without limitation the rights
> to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
> copies of the Software, and to permit persons to whom the Software is
> furnished to do so, subject to the following conditions:
>
> > The above copyright notice and this permission notice shall be included in
> > all copies or substantial portions of the Software.
>
> > THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> > IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> > FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
> > AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
> > LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
> > OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
> > THE SOFTWARE.

### hash.js
License: MIT
By: Fedor Indutny
Repository: git@github.com:indutny/hash.js

### immutable
License: MIT
By: Lee Byron
Repository: git://github.com/facebook/immutable-js.git

> MIT License
>
> Copyright (c) 2014-present, Facebook, Inc.
>
> Permission is hereby granted, free of charge, to any person obtaining a copy
> of this software and associated documentation files (the "Software"), to deal
> in the Software without restriction, including without limitation the rights
> to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
> copies of the Software, and to permit persons to whom the Software is
> furnished to do so, subject to the following conditions:
>
> > The above copyright notice and this permission notice shall be included in all
> > copies or substantial portions of the Software.
>
> > THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> > IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> > FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### inherits
License: ISC
Repository: git://github.com/isaacs/inherits

The ISC License

Copyright (c) Isaac Z. Schlueter

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

### is-number
License: MIT
By: Jon Schlinkert, Olsten Larck, Rouven Weling
Repository: jonschlinkert/is-number

The MIT License (MIT)

Copyright (c) 2014-present, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
> AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
> LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
> OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
> THE SOFTWARE.

---------------------------------------

## is-reference
License: MIT
By: Rich Harris
Repository: git+https://github.com/Rich-Harris/is-reference.git

---------------------------------------

## locate-character
License: MIT
By: Rich Harris
Repository: Rich-Harris/locate-character

---------------------------------------

## magic-string
License: MIT
By: Rich Harris
Repository: https://github.com/rich-harris/magic-string

> Copyright 2018 Rich Harris
>
> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---------------------------------------
## micromatch
License: MIT
By: Jon Schlinkert, Amila Welihinda, Bogdan Chadkin, Brian Woodward, Devon Govett, Elan Shanker, Fabrcio Matt, Martin Kolrik, Olsten Larck, Paul Miller, Tom Byrer, Tyler Akins, Peter Bright
Repository: micromatch/micromatch

> The MIT License (MIT)
>
> Copyright (c) 2014-present, Jon Schlinkert.
>
> Permission is hereby granted, free of charge, to any person obtaining a copy
> of this software and associated documentation files (the "Software"), to deal
> in the Software without restriction, including without limitation the rights
> to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
> copies of the Software, and to permit persons to whom the Software is
> furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in
> all copies or substantial portions of the Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
> AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
> LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
> OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
> THE SOFTWARE.

# minimalistic-assert
License: ISC
Repository: https://github.com/calvinmetcalf/minimalistic-assert.git

> Copyright 2015 Calvin Metcalf
>
> Permission to use, copy, modify, and/or distribute this software for any purpose
> with or without fee is hereby granted, provided that the above copyright notice
> and this permission notice appear in all copies.
>
> THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
> REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
> AND
> FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
> INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
> LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
> OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
### minimist
License: MIT
By: James Halliday
Repository: git://github.com/substack/minimist.git

> This software is released under the MIT license:
> 
> Permission is hereby granted, free of charge, to any person obtaining a copy of
> this software and associated documentation files (the "Software"), to deal in
> the Software without restriction, including without limitation the rights to
> use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
> the Software, and to permit persons to whom the Software is furnished to do so,
> subject to the following conditions:
> 
> The above copyright notice and this permission notice shall be included in all
> copies or substantial portions of the Software.
> 
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
> FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
> COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
> IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
> CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### parse-ms
License: MIT
By: Sindre Sorhus
Repository: sindresorhus/parse-ms

### picomatch
License: MIT
By: Jon Schlinkert
Repository: micromatch/picomatch

> The MIT License (MIT)
>
> Copyright (c) 2017-present, Jon Schlinkert.
>
> Permission is hereby granted, free of charge, to any person obtaining a copy
> of this software and associated documentation files (the "Software"), to deal
> in the Software without restriction, including without limitation the rights
> to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
> copies of the Software, and to permit persons to whom the Software is
> furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in
> all copies or substantial portions of the Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
> AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
> LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
> OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
> THE SOFTWARE.

## pretty-bytes
License: MIT
By: Sindre Sorhus
Repository: sindresorhus/pretty-bytes

## pretty-ms
License: MIT
By: Sindre Sorhus
Repository: sindresorhus/pretty-ms

## require-relative
License: MIT
By: Valerio Proietti
Repository: git://github.com/kamicane/require-relative.git

## rollup-pluginutils
License: MIT
By: Rich Harris
Repository: rollup/rollup-pluginutils

## signal-exit
License: ISC
By: Ben Coe
Repository: https://github.com/tapjs/signal-exit.git

> The ISC License
>
> Copyright (c) 2015, Contributors
>
> Permission to use, copy, modify, and/or distribute this software
> for any purpose with or without fee is hereby granted, provided
> that the above copyright notice and this permission notice
> appear in all copies.
>
> THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
> WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
> OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
> LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
> OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
> WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
> ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

---------------------------------------

## sourcemap-codec

License: MIT
By: Rich Harris
Repository: https://github.com/Rich-Harris/sourcemap-codec

> The MIT License
>
> Copyright (c) 2015 Rich Harris
>
> Permission is hereby granted, free of charge, to any person obtaining a copy
> of this software and associated documentation files (the "Software"), to deal
> in the Software without restriction, including without limitation the rights
> to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
> copies of the Software, and to permit persons to whom the Software is
> furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in
> all copies or substantial portions of the Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
> AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
> LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
> OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
> THE SOFTWARE.
### time-zone
License: MIT
By: Sindre Sorhus
Repository: sindresorhus/time-zone

### to-regex-range
License: MIT
By: Jon Schlinkert, Rouven Weling
Repository: micromatch/to-regex-range

> The MIT License (MIT)
>
> Copyright (c) 2015-present, Jon Schlinkert.
>
> Permission is hereby granted, free of charge, to any person obtaining a copy
> of this software and associated documentation files (the "Software"), to deal
> in the Software without restriction, including without limitation the rights
> to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
> copies of the Software, and to permit persons to whom the Software is
> furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in
> all copies or substantial portions of the Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
> AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
> LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
> OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
> THE SOFTWARE.

### tslib
License: Apache-2.0
By: Microsoft Corp.
Repository: https://github.com/Microsoft/tslib.git

> Apache License
>
> Version 2.0, January 2004
>
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants
to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

> 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

> 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

> You must give any other recipients of the Work or Derivative Works a copy of this License; and

> You must cause any modified files to carry prominent notices stating that You changed the files; and

> You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

> If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

> 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

> 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
> 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
>
> 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
>
> 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
>
> END OF TERMS AND CONDITIONS

```
## turbocolor
License: MIT
By: Jorge Bucaran
Repository: jorgebucaran/turbocolor

> Copyright 2015-present [Jorge Bucaran](https://github.com/jorgebucaran)
>
> Permission is hereby granted, free of charge, to any person obtaining a copy
> of this software and associated documentation files (the "Software"), to deal
> in the Software without restriction, including without limitation the rights
> to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
> copies of the Software, and to permit persons to whom the Software is
> furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in all
> copies or substantial portions of the Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
> IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
> FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
> AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
> LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
```
 MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) MapBox
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) Elan Shanker and Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

  0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

  1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Copyright 2019 "David Mark Clements <david.mark.clements@gmail.com>"

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The MIT License (MIT)

Copyright (c) 2014-2017 Calvin Metcalf & contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2012-2015, JP Richardson <jprichardson@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2011-2013 Jake Luer jake@alogicalparadox.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

thestig@chromium.org

Copyright (c) 2015, Valve Corporation
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TLS Lite includes code from different sources. All code is either dedicated to the public domain by its authors, or available under a BSD-style license. In particular:

- Code written by Trevor Perrin, Kees Bos, Sam Rushing, Dimitris Moraitis, Marcelo Fernandez, Martin von Loewis, Dave Baggett, and Yngve Pettersen is available under the following terms:

  This is free and unencumbered software released into the public domain.

  Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

  In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- Code written by Bram Cohen (rijndael.py) was dedicated to the public domain by its author. See rijndael.py for details.

- Code written by Google is available under the following terms:

Copyright (c) 2008, The Chromium Authors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2012-2016 Mickael Daniel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without

Open Source Used In DNAC Platform Antelope  12774
restriction, including without limitation the rights to use, 
copy, modify, merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom the 
Software is furnished to do so, subject to the following 
conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES 
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND 
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT 
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, 
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING 
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR 
OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without 
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright 
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright 
  notice, this list of conditions and the following disclaimer in the 
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" 
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE 
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE 
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY 
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES 
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; 
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND 
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT 
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF 
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (c) 2015 John Hiesey

Permission is hereby granted, free of charge, 
to any person obtaining a copy of this software and 
associated documentation files (the "Software"), to 
deal in the Software without restriction, including 
without limitation the rights to use, copy, modify, 
merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom 
the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Stefan Thomas

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2012-2016 Kristian Faeldt <faeldt_kristian@cyberagent.co.jp>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The ISC License

Copyright (c) 2015 Elan Shanker

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The MIT License

Copyright (c) 2013 Dominic Tarr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright 2012-2015 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Yahoo! Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2017 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2017 Bart Muzzin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Derived from:

The MIT License (MIT)

Copyright (c) 2015 mojmir svoboda

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Copyright (c) 2013, Ethan Furman.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

Redistributions of source code must retain the above
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name Ethan Furman nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NOTE! This copyright does *not* cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does *not* fall under the heading of "derived work". Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the linux kernel) is copyrighted by me and others who actually wrote it.

Linus Torvalds

----------------------------------------

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free software— to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy  <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.


Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Opus is subject to the royalty-free patent licenses which are specified at:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Microsoft Corporation:
https://datatracker.ietf.org/ipr/1914/

Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/
(Copied from the README.)

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this
software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.)
So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that
"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

jconfig.h is distributed under the MPL 1.1/GPL 2.0/LGPL 2.1 tri-license.

jmorecfg.h contains modifications, which are distributed under the Netscape Public License.

# Copyright (c) 2015 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

Copyright JS Foundation and other contributors, https://js.foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# https://developers.google.com/open-source/cla/individual
# https://developers.google.com/open-source/cla/corporate
#
# Names should be added to this file as:
#   Name <email address>

Google Inc.
Nicolas Capens <capn@google.com>
Alexis Htu <sugoi@google.com>
Shannon Woods <shannonwoods@google.com>
Corentin Wallez <cwallez@google.com>

Open Source Used In DNAC Platform Antelope 12791
Greg Hartman <ghartman@google.com>
Ping-Hao Wu <pinghao@google.com>
Maxime Grgoire <mgregoire@google.com>
Veranika Liaukevich <veranika@google.com>
John Bauman <jbauman@google.com>
Keun Soo Yim <yim@google.com>
John Sheu <sheu@google.com>
Philippe Hamel <hamelphi@google.com>
Daniel Toyama <kenjitoyama@google.com>
Meng-Lin Wu <marleymoo@google.com>
Krzysztof Kosiski <krzysio@google.com>
Chris Forbes <chrisforbes@google.com>
Ben Clayton <bclayton@google.com>
Hernan Liatis <hliatis@google.com>
Logan (Tzu-hsiang) Chien <loganchien@google.com>
Stephen White <senorblanco@google.com>
Raymond Chiu <chiur@google.com>

TransGaming Inc.
Nicolas Capens
Gavriel State
Jim MacArthur
Daniel Koch
Luther Johnson
Rob Stepinski
(The MIT License)

Copyright (c) 2017, Ryan Zimmerman <opensrc@ryanzim.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the 'Software'), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
This license is for the example glTF code, not the [specification](specification/README.md) and
[schema](specification).
Original work Copyright (c) 2012, Motorola Mobility, Inc.
All Rights Reserved.
BSD License.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Motorola Mobility nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional contributions copyright their respective contributors (also BSD License):
* Fabrice Robinet
* Analytical Graphics, Inc.
The MIT License (MIT)

Copyright (c) 2014 KARASZI Istvn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Yasm is Copyright (c) 2001-2014 Peter Johnson and other Yasm developers.

Yasm developers and/or contributors include:
Peter Johnson
Michael Urman
Brian Gladman (Visual Studio build files, other fixes)
Stanislav Karchebny (options parser)
Mathieu Monnier (SSE4 instruction patches, NASM preprocessor additions)
Anonymous "NASM64" developer (NASM preprocessor fixes)
Stephen Polkowski (x86 instruction patches)
Henryk Richter (Mach-O object format)
Ben Skeggs (patches, bug reports)
Alexei Svitkine (GAS preprocessor)
Samuel Thibault (TASM parser and frontend)

-----------------------------------
Yasm licensing overview and summary
-----------------------------------

Note: This document does not provide legal advice nor is it the actual license of any part of Yasm. See the individual licenses for complete details. Consult a lawyer for legal advice.

The primary license of Yasm is the 2-clause BSD license. Please use this license if you plan on submitting code to the project.

Yasm has absolutely no warranty; not even for merchantibility or fitness for a particular purpose.

-------
Libyasm
-------
Libyasm is 2-clause or 3-clause BSD licensed, with the exception of bitvect, which is triple-licensed under the Artistic license, GPL, and LGPL. Libyasm is thus GPL and LGPL compatible. In addition, this also means that libyasm is free for binary-only distribution as long as the terms of the 3-clause BSD license and Artistic license (as it applies to bitvect) are fulfilled.

-------
Modules
-------
The modules are 2-clause or 3-clause BSD licensed.
Frontends

The frontends are 2-clause BSD licensed.

License Texts

The full text of all licenses are provided in separate files in the source distribution. Each source file may include the entire license (in the case of the BSD and Artistic licenses), or may reference the GPL or LGPL license file.

- BSD.txt - 2-clause and 3-clause BSD licenses
- Artistic.txt - Artistic license
- GNU_GPL-2.0 - GNU General Public License
- GNU_LGPL-2.0 - GNU Library General Public License

Copyright (C) 2005, 2006, 2007, 2008, 2009 Apple Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY APPLE INC. AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APPLE INC. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The "New" BSD License

Copyright (c) 2004-2015, The Dojo Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.
* Neither the name of the Dojo Foundation nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2014 Guillermo Rauch

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.
For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)
"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library’s complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.
(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENTERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Copyright (C) 2016 by Marijn Haverbeke <marijnh@gmail.com> and others

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

language: node_js
node_js:
  - '0.12'
  - '4'
  - '6'
  - '7'
before_install:
  - npm install -g npm@latest
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015-2016 Douglas Christopher Wilson <doug@somethingdoug.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) James Talmage <james@talmage.io> (github.com/jamestalmage)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2008 Fair Oaks Labs, Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

// Copyright (c) 2017 The Chromium OS Authors. All rights reserved.

// Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

// * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
// * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

validate-npm-package-license

Give me a string and I'll tell you if it's a valid npm package license string.

```javascript
var valid = require('validate-npm-package-license');
```

SPDX license identifiers are valid license strings:

```javascript
```
var assert = require('assert');
var validSPDXExpression = {
  validForNewPackages: true,
  validForOldPackages: true,
  spdx: true
};

assert.deepEqual(valid('MIT'), validSPDXExpression);
assert.deepEqual(valid('BSD-2-Clause'), validSPDXExpression);
assert.deepEqual(valid('Apache-2.0'), validSPDXExpression);
assert.deepEqual(valid('ISC'), validSPDXExpression);

The function will return a warning and suggestion for nearly-correct license identifiers:

```
assert.deepEqual(
  valid('Apache 2.0'),
  {
    validForOldPackages: false,
    validForNewPackages: false,
    warnings: [
      'license should be ' +
      'a valid SPDX license expression (without "LicenseRef"), ' +
      '"UNLICENSED", or ' +
      '"SEE LICENSE IN <filename>"',
      'license is similar to the valid expression "Apache-2.0"
    ]
  }
);
```

SPDX expressions are valid, too ...

```
// Simple SPDX license expression for dual licensing
assert.deepEqual(
  valid('GPL-3.0-only OR BSD-2-Clause'),
  validSPDXExpression
);
```

... except if they contain `LicenseRef`:

```
var warningAboutLicenseRef = {
  validForOldPackages: false,
  validForNewPackages: false,
```

```
If you can't describe your licensing terms with standardized SPDX identifiers, put the terms in a file in the package and point users there:

```javascript
assert.deepEqual(
    valid('SEE LICENSE IN LICENSE.txt'),
    {
        validForNewPackages: true,
        validForOldPackages: true,
        inFile: 'LICENSE.txt'
    }
);

assert.deepEqual(
    valid('SEE LICENSE IN license.md'),
    {
        validForNewPackages: true,
        validForOldPackages: true,
        inFile: 'license.md'
    }
);
```

If there aren't any licensing terms, use `UNLICENSED`:

```javascript
var unlicensed = {
    validForNewPackages: true,
```
validForOldPackages: true,
unlicensed: true
};
assert.deepEqual(valid('UNLICENSED'), unlicensed);
assert.deepEqual(valid('UNLICENCED'), unlicensed);
```

APPLE PUBLIC SOURCE LICENSE
Version 2.0 - August 6, 2003

Please read this License carefully before downloading this software. By downloading or using this software, you are agreeing to be bound by the terms of this License. If you do not or cannot agree to the terms of this License, please do not download or use the software.

1. General; Definitions. This License applies to any program or other work which Apple Computer, Inc. ("Apple") makes publicly available and which contains a notice placed by Apple identifying such program or work as "Original Code" and stating that it is subject to the terms of this Apple Public Source License version 2.0 ("License"). As used in this License:

1.1 "Applicable Patent Rights" mean: (a) in the case where Apple is the grantor of rights, (i) claims of patents that are now or hereafter acquired, owned by or assigned to Apple and (ii) that cover subject matter contained in the Original Code, but only to the extent necessary to use, reproduce and/or distribute the Original Code without infringement; and (b) in the case where You are the grantor of rights, (i) claims of patents that are now or hereafter acquired, owned by or assigned to You and (ii) that cover subject matter in Your Modifications, taken alone or in combination with Original Code.

1.2 "Contributor" means any person or entity that creates or contributes to the creation of Modifications.

1.3 "Covered Code" means the Original Code, Modifications, the combination of Original Code and any Modifications, and/or any respective portions thereof.

1.4 "Externally Deploy" means: (a) to sublicense, distribute or otherwise make Covered Code available, directly or indirectly, to anyone other than You; and/or (b) to use Covered Code, alone or as part of a Larger Work, in any way to provide a service, including but not limited to delivery of content, through electronic communication with a client other than You.

1.5 "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.
1.6 "Modifications" mean any addition to, deletion from, and/or change to, the substance and/or structure of the Original Code, any previous Modifications, the combination of Original Code and any previous Modifications, and/or any respective portions thereof. When code is released as a series of files, a Modification is: (a) any addition to or deletion from the contents of a file containing Covered Code; and/or (b) any new file or other representation of computer program statements that contains any part of Covered Code.

1.7 "Original Code" means (a) the Source Code of a program or other work as originally made available by Apple under this License, including the Source Code of any updates or upgrades to such programs or works made available by Apple under this License, and that has been expressly identified by Apple as such in the header file(s) of such work; and (b) the object code compiled from such Source Code and originally made available by Apple under this License.

1.8 "Source Code" means the human readable form of a program or other work that is suitable for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an executable (object code).

1.9 "You" or "Your" means an individual or a legal entity exercising rights under this License. For legal entities, "You" or "Your" includes any entity which controls, is controlled by, or is under common control with, You, where "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

2. Permitted Uses; Conditions & Restrictions. Subject to the terms and conditions of this License, Apple hereby grants You, effective on the date You accept this License and download the Original Code, a world-wide, royalty-free, non-exclusive license, to the extent of Apple's Applicable Patent Rights and copyrights covering the Original Code, to do the following:

2.1 Unmodified Code. You may use, reproduce, display, perform, internally distribute within Your organization, and Externally Deploy verbatim, unmodified copies of the Original Code, for commercial or non-commercial purposes, provided that in each instance:

(a) You must retain and reproduce in all copies of Original Code the copyright and other proprietary notices and disclaimers of Apple as they appear in the Original Code, and keep intact all notices in the Original Code that refer to this License; and
(b) You must include a copy of this License with every copy of Source Code of Covered Code and documentation You distribute or Externally Deploy, and You may not offer or impose any terms on such Source Code that alter or restrict this License or the recipients' rights hereunder, except as permitted under Section 6.

2.2 Modified Code. You may modify Covered Code and use, reproduce, display, perform, internally distribute within Your organization, and Externally Deploy Your Modifications and Covered Code, for commercial or non-commercial purposes, provided that in each instance You also meet all of these conditions:

(a) You must satisfy all the conditions of Section 2.1 with respect to the Source Code of the Covered Code;

(b) You must duplicate, to the extent it does not already exist, the notice in Exhibit A in each file of the Source Code of all Your Modifications, and cause the modified files to carry prominent notices stating that You changed the files and the date of any change; and

(c) If You Externally Deploy Your Modifications, You must make Source Code of all Your Externally Deployed Modifications either available to those to whom You have Externally Deployed Your Modifications, or publicly available. Source Code of Your Externally Deployed Modifications must be released under the terms set forth in this License, including the license grants set forth in Section 3 below, for as long as you Externally Deploy the Covered Code or twelve (12) months from the date of initial External Deployment, whichever is longer. You should preferably distribute the Source Code of Your Externally Deployed Modifications electronically (e.g. download from a web site).

2.3 Distribution of Executable Versions. In addition, if You Externally Deploy Covered Code (Original Code and/or Modifications) in object code, executable form only, You must include a prominent notice, in the code itself as well as in related documentation, stating that Source Code of the Covered Code is available under the terms of this License with information on how and where to obtain such Source Code.

2.4 Third Party Rights. You expressly acknowledge and agree that although Apple and each Contributor grants the licenses to their respective portions of the Covered Code set forth herein, no assurances are provided by Apple or any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Apple and each Contributor disclaim any liability to You for claims brought by any other entity based on
infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow You to distribute the Covered Code, it is Your responsibility to acquire that license before distributing the Covered Code.

3. Your Grants. In consideration of, and as a condition to, the licenses granted to You under this License, You hereby grant to any person or entity receiving or distributing Covered Code under this License a non-exclusive, royalty-free, perpetual, irrevocable license, under Your Applicable Patent Rights and other intellectual property rights (other than patent) owned or controlled by You, to use, reproduce, display, perform, modify, sublicense, distribute and Externally Deploy Your Modifications of the same scope and extent as Apple’s licenses under Sections 2.1 and 2.2 above.

4. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In each such instance, You must make sure the requirements of this License are fulfilled for the Covered Code or any portion thereof.

5. Limitations on Patent License. Except as expressly stated in Section 2, no other patent rights, express or implied, are granted by Apple herein. Modifications and/or Larger Works may require additional patent licenses from Apple which Apple may grant in its sole discretion.

6. Additional Terms. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations and/or other rights consistent with the scope of the license granted herein (“Additional Terms”) to one or more recipients of Covered Code. However, You may do so only on Your own behalf and as Your sole responsibility, and not on behalf of Apple or any Contributor. You must obtain the recipient’s agreement that any such Additional Terms are offered by You alone, and You hereby agree to indemnify, defend and hold Apple and every Contributor harmless for any liability incurred by or claims asserted against Apple or such Contributor by reason of any such Additional Terms.

7. Versions of the License. Apple may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Once Original Code has been published under a particular version of this License, You may continue to use it under the terms of that version. You may also choose to use such Original Code under the terms of any subsequent version of this
License published by Apple. No one other than Apple has the right to modify the terms applicable to Covered Code created under this License.

8. NO WARRANTY OR SUPPORT. The Covered Code may contain in whole or in part pre-release, untested, or not fully tested works. The Covered Code may contain errors that could cause failures or loss of data, and may be incomplete or contain inaccuracies. You expressly acknowledge and agree that use of the Covered Code, or any portion thereof, is at Your sole and entire risk. THE COVERED CODE IS PROVIDED "AS IS" AND WITHOUT WARRANTY, UPGRADES OR SUPPORT OF ANY KIND AND APPLE AND APPLE'S LICENSOR(S) (COLLECTIVELY REFERRED TO AS "APPLE" FOR THE PURPOSES OF SECTIONS 8 AND 9) AND ALL CONTRIBUTORS EXPRESSLY DISCLAIM ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY, OF SATISFACTORY QUALITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY, OF QUIET ENJOYMENT, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. APPLE AND EACH CONTRIBUTOR DOES NOT WARRANT AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE COVERED CODE, THAT THE FUNCTIONS CONTAINED IN THE COVERED CODE WILL MEET YOUR REQUIREMENTS, THAT THE OPERATION OF THE COVERED CODE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE COVERED CODE WILL BE CORRECTED. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY APPLE, AN APPLE AUTHORIZED REPRESENTATIVE OR ANY CONTRIBUTOR SHALL CREATE A WARRANTY. You acknowledge that the Covered Code is not intended for use in the operation of nuclear facilities, aircraft navigation, communication systems, or air traffic control machines in which case the failure of the Covered Code could lead to death, personal injury, or severe physical or environmental damage.

9. LIMITATION OF LIABILITY. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT SHALL APPLE OR ANY CONTRIBUTOR BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE OR YOUR USE OR INABILITY TO USE THE COVERED CODE, OR ANY PORTION THEREOF, WHETHER UNDER A THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCTS LIABILITY OR OTHERWISE, EVEN IF APPLE OR SUCH CONTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY REMEDY. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OF LIABILITY OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION MAY NOT APPLY TO YOU. In no event shall Apple's total liability to You for all damages (other than as may be required by applicable law) under this License exceed the amount of fifty dollars ($50.00).

10. Trademarks. This License does not grant any rights to use the trademarks or trade names "Apple", "Apple Computer", "Mac", "Mac OS", "QuickTime", "QuickTime Streaming Server" or any other trademarks, service marks, logos or trade names belonging to Apple (collectively
"Apple Marks") or to any trademark, service mark, logo or trade name belonging to any Contributor. You agree not to use any Apple Marks in or as part of the name of products derived from the Original Code or to endorse or promote products derived from the Original Code other than as expressly permitted by and in strict compliance at all times with Apple's third party trademark usage guidelines which are posted at http://www.apple.com/legal/guidelinesfor3rdparties.html.

11. Ownership. Subject to the licenses granted under this License, each Contributor retains all rights, title and interest in and to any Modifications made by such Contributor. Apple retains all rights, title and interest in and to the Original Code and any Modifications made by or on behalf of Apple ("Apple Modifications"), and such Apple Modifications will not be automatically subject to this License. Apple may, at its sole discretion, choose to license such Apple Modifications under this License, or on different terms from those contained in this License or may choose not to license them at all.

12. Termination.

12.1 Termination. This License and the rights granted hereunder will terminate:

(a) automatically without notice from Apple if You fail to comply with any term(s) of this License and fail to cure such breach within 30 days of becoming aware of such breach;

(b) immediately in the event of the circumstances described in Section 13.5(b); or

(c) automatically without notice from Apple if You, at any time during the term of this License, commence an action for patent infringement against Apple; provided that Apple did not first commence an action for patent infringement against You in that instance.

12.2 Effect of Termination. Upon termination, You agree to immediately stop any further use, reproduction, modification, sublicensing and distribution of the Covered Code. All sublicenses to the Covered Code which have been properly granted prior to termination shall survive any termination of this License. Provisions which, by their nature, should remain in effect beyond the termination of this License shall survive, including but not limited to Sections 3, 5, 8, 9, 10, 11, 12.2 and 13. No party will be liable to any other for compensation, indemnity or damages of any sort solely as a result of terminating this License in accordance with its terms, and termination of this License will be without prejudice to any other right or remedy of any party.

13.1 Government End Users. The Covered Code is a "commercial item" as defined in FAR 2.101. Government software and technical data rights in the Covered Code include only those rights customarily provided to the public as defined in this License. This customary commercial license in technical data and software is provided in accordance with FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for Department of Defense purchases, DFAR 252.227-7015 (Technical Data -- Commercial Items) and 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation). Accordingly, all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

13.2 Relationship of Parties. This License will not be construed as creating an agency, partnership, joint venture or any other form of legal association between or among You, Apple or any Contributor, and You will not represent to the contrary, whether expressly, by implication, appearance or otherwise.

13.3 Independent Development. Nothing in this License will impair Apple's right to acquire, license, develop, have others develop for it, market and/or distribute technology or products that perform the same or similar functions as, or otherwise compete with, Modifications, Larger Works, technology or products that You may develop, produce, market or distribute.

13.4 Waiver; Construction. Failure by Apple or any Contributor to enforce any provision of this License will not be deemed a waiver of future enforcement of that or any other provision. Any law or regulation which provides that the language of a contract shall be construed against the drafter will not apply to this License.

13.5 Severability. (a) If for any reason a court of competent jurisdiction finds any provision of this License, or portion thereof, to be unenforceable, that provision of the License will be enforced to the maximum extent permissible so as to effect the economic benefits and intent of the parties, and the remainder of this License will continue in full force and effect. (b) Notwithstanding the foregoing, if applicable law prohibits or restricts You from fully and/or specifically complying with Sections 2 and/or 3 or prevents the enforceability of either of those Sections, this License will immediately terminate and You must immediately discontinue any use of the Covered Code and destroy all copies of it that are in your possession or control.

13.6 Dispute Resolution. Any litigation or other dispute resolution between You and Apple relating to this License shall take place in the
Northern District of California, and You and Apple hereby consent to
the personal jurisdiction of, and venue in, the state and federal
courts within that District with respect to this License. The
application of the United Nations Convention on Contracts for the
International Sale of Goods is expressly excluded.

13.7 Entire Agreement; Governing Law. This License constitutes the
total agreement between the parties with respect to the subject
matter hereof. This License shall be governed by the laws of the
United States and the State of California, except that body of
California law concerning conflicts of law.

Where You are located in the province of Quebec, Canada, the following
clause applies: The parties hereby confirm that they have requested
that this License and all related documents be drafted in English. Les
parties ont exige que le present contrat et tous les documents
connexes soient rediges en anglais.

EXHIBIT A.

"Portions Copyright (c) 1999-2003 Apple Computer, Inc. All Rights
Reserved.

This file contains Original Code and/or Modifications of Original Code
as defined in and that are subject to the Apple Public Source License
Version 2.0 (the 'License'). You may not use this file except in
compliance with the License. Please obtain a copy of the License at
http://www.opensource.apple.com/apsl/ and read it before using this
file.

The Original Code and all software distributed under the License are
distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER
EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES,
INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT.
Please see the License for the specific language governing rights and
limitations under the License."

1.2955 crc 3.8.0
1.2955.1 Available under license :
The MIT License (MIT)

Copyright 2014 Alex Gorbatchev

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2956 https-proxy-agent 3.0.1

1.2956.1 Available under license:

No license file was found, but licenses were detected in source scan.

https-proxy-agent

===============

### An HTTP(s) proxy `http.Agent` implementation for HTTPS

[![Build Status](https://github.com/TooTallNate/node-https-proxy-agent/workflows/Node%20CI/badge.svg)](https://github.com/TooTallNate/node-https-proxy-agent/actions?workflow=Node+CI)

This module provides an `http.Agent` implementation that connects to a specified
HTTP or HTTPS proxy server, and can be used with the built-in `https` module.

Specifically, this `Agent` implementation connects to an intermediary "proxy"
server and issues the [CONNECT HTTP method][CONNECT], which tells the proxy to
open a direct TCP connection to the destination server.

Since this agent implements the CONNECT HTTP method, it also works with other
protocols that use this method when connecting over proxies (i.e. WebSockets).
See the "Examples" section below for more.

Installation

```
```

Install with `npm`:

```bash
```
Examples
--------

#### `https` module example

```js
var url = require('url');
var https = require('https');
var HttpsProxyAgent = require('https-proxy-agent');

// HTTP/HTTPS proxy to connect to
var proxy = process.env.http_proxy || 'http://168.63.76.32:3128';
console.log('using proxy server %j', proxy);

// HTTPS endpoint for the proxy to connect to
console.log('attempting to GET %j', endpoint);
var options = url.parse(endpoint);

// create an instance of the HttpsProxyAgent class with the proxy server information
var agent = new HttpsProxyAgent(proxy);
options.agent = agent;

https.get(options, function (res) {
    console.log("response" event!, res.headers);
    res.pipe(process.stdout);
});
```

#### `ws` WebSocket connection example

```js
var url = require('url');
var WebSocket = require('ws');
var HttpsProxyAgent = require('https-proxy-agent');

// HTTP/HTTPS proxy to connect to
var proxy = process.env.http_proxy || 'http://168.63.76.32:3128';
console.log('using proxy server %j', proxy);

// WebSocket endpoint for the proxy to connect to
var endpoint = process.argv[2] || 'ws://echo.websocket.org';
var parsed = url.parse(endpoint);
console.log('attempting to connect to WebSocket %j', endpoint);
```
// create an instance of the `HttpsProxyAgent` class with the proxy server information
var options = url.parse(proxy);

var agent = new HttpsProxyAgent(options);

// finally, initiate the WebSocket connection
var socket = new WebSocket(endpoint, { agent: agent });

socket.on('open', function () {
  console.log("open" event!);
  socket.send('hello world');
});

socket.on('message', function (data, flags) {
  console.log("message" event! %j %j", data, flags);
  socket.close();
});

```

API
---

### new HttpsProxyAgent(Object options)

The `HttpsProxyAgent` class implements an `http.Agent` subclass that connects to the specified "HTTP(s) proxy server" in order to proxy HTTPS and/or WebSocket requests. This is achieved by using the [HTTP `CONNECT` method][CONNECT].

The `options` argument may either be a string URI of the proxy server to use, or an "options" object with more specific properties:

* `host` - String - Proxy host to connect to (may use `hostname` as well). Required.
* `port` - Number - Proxy port to connect to. Required.
* `protocol` - String - If `https:` then use TLS to connect to the proxy.
* `headers` - Object - Additional HTTP headers to be sent on the HTTP CONNECT method.
* Any other options given are passed to the `net.connect()`/`tls.connect()` functions.

License
-------

(The MIT License)

Copyright (c) 2013 Nathan Rajlich &lt;nathan@tootallnate.net&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[CONNECT]: http://en.wikipedia.org/wiki/HTTP_tunnel#HTTP_CONNECT_Tunneling

Found in path(s):
* /opt/cola/permits/1186854399_1628187065.73/0/https-proxy-agent-3-0-1-5-tgz/package/README.md

1.2957 netty 4.1.61.Final
1.2957.1 Available under license :
Copyright (c) 2003-2008 Yuta Mori All Rights Reserved.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License

Copyright (c) 2009 William Kinney

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Copyright 2009-2010 Ning, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/*
* Copyright (c) 2004-2006, 2008, 2009, 2011 Apple Inc. All rights reserved.
* @APPLE_LICENSE_HEADER_START@
* This file contains Original Code and/or Modifications of Original Code
* as defined in and that are subject to the Apple Public Source License
* Version 2.0 (the "License"). You may not use this file except in
* compliance with the License. Please obtain a copy of the License at
* https://www.opensource.apple.com/apsl/ and read it before using this
* file.
*
* The Original Code and all software distributed under the License are
* distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER
* EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES,
* INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT.
* Please see the License for the specific language governing rights and
* limitations under the License.
*
* @APPLE_LICENSE_HEADER_END@
*/

The MIT License (MIT)

Copyright (c) 2014 Cory Benfield

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2010-2011 Matthew J. Francis and Contributors of the jbzip2 Project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the domicitors holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to
the detriment of the Dedicator's heirs and successors. Dedicator intends this
dedication to be an overt act of relinquishment in perpetuate of all present
and future rights under copyright law, whether vested or contingent, in the
Work. Dedicator understands that such relinquishment of all rights includes
the relinquishment of all rights to enforce (by lawsuit or otherwise) those
copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be
freely reproduced, distributed, transmitted, used, modified, built upon, or
otherwise exploited by anyone for any purpose, commercial or non-commercial,
and in any way, including by methods that have not yet been invented or
conceived.

The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa
Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Netty Project
=================

Please visit the Netty web site for more information:

* https://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:
https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

-------------------------------------------------------------------------------

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * https://www.slf4j.org/

This product contains a modified portion of 'Apache Harmony', an open source
Java SE, which can be obtained at:

* NOTICE:
  * license/NOTICE.harmony.txt
* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * https://archive.apache.org/dist/harmony/

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://github.com/y-256/libdivsufsort

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/protobuf

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * https://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (Apache License 2.0)
  * HOMEPAGE:
  * https://github.com/jboss-remoting/jboss-marshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
  * HOMEPAGE:
  * https://github.com/google/caliper

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
  * HOMEPAGE:
  * https://commons.apache.org/logging/

This product optionally depends on 'Apache Log4j', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
  * HOMEPAGE:
  * https://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
  * HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Cory Benfield. It can be obtained at:

* LICENSE:
  * license/LICENSE.hyper-hpack.txt (MIT License)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Tatsuhiro Tsujikawa. It can be obtained at:

* LICENSE:
  * license/LICENSE.nghttp2-hpack.txt (MIT License)
* HOMEPAGE:
  * https://github.com/nghttp2/nghttp2

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang

This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user has everything necessary to run the Maven build.

* LICENSE:
  * license/LICENSE.mvn-wrapper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/takari/maven-wrapper

This product contains the dnsinfo.h header file, that provides a way to retrieve the system DNS configuration on MacOS.
This private header is also used by Apple's open source mDNSResponder (https://opensource.apple.com/tarballs/mDNSResponder/).

* LICENSE:
  * license/LICENSE.dnsinfo.txt (Apple Public Source License 2.0)
* HOMEPAGE:
  * Apache License
    * Version 2.0, January 2004
    * https://www.apache.org/licenses/
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
(BSD License: https://www.opensource.org/licenses/bsd-license)

Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the
following conditions are met:

* Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other
materials provided with the distribution.

* Neither the name of the Webbit nor the names of
its contributors may be used to endorse or promote products
derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(https://www.bouncycastle.org)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*
 * Copyright (c) 2004-2007 QOS.ch
 * All rights reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/

Protocol Buffers - Google's data interchange format
Copyright 2013 Google Inc. All rights reserved.
https://developers.google.com/protocol-buffers/

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

This copy of Aalto XML processor is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

https://www.apache.org/licenses/

A copy is also included with both the the downloadable source code package and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/asl"; and in jar "META-INF/"

Apache Harmony

Copyright 2006, 2010 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

1.2958 external-editor 3.1.0

1.2958.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Kevin Gravier

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2959 nanoid 1.0.7
1.2959.1 Available under license:

The MIT License (MIT)

Copyright 2017 Andrey Sitnik <andrey@sitnik.ru>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2960 deep-is 0.1.3 0.1.3
1.2960.1 Available under license:

Copyright (c) 2012, 2013 Thorsten Lorenz <thlorenz@gmx.de>
Copyright (c) 2012 James Halliday <mail@substack.net>
Copyright (c) 2009 Thomas Robinson <280north.com>

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2961 fill-range 4.0.0
1.2961.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2962 netty-handler 4.1.59.Final
1.2962.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/SslContextBuilder.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/OpenSslSessionTicketKey.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolAccessor.java

No license file was found, but licenses were detected in source scan.

/
* Copyright 2019 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/OpenSslPrivateKeyMethod.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/4util/KeyManagerFactoryWrapper.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/SslMasterKeyHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/address/package-info.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/util/X509KeyManagerWrapper.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/PseudoRandomFunction.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/address/DynamicAddressConnectHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/TrustManagerFactoryWrapper.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/util/SimpleKeyManagerFactory.java

No license file was found, but licenses were detected in source scan.

/
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/ApplicationProtocolUtil.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/LazyX509Certificate.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/CipherSuiteFilter.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ipfilter/IpFilterRule.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/OpenSslEngine.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/BouncyCastleSelfSignedCertGenerator.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/OpenSslNpnApplicationProtocolNegotiator.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/JdkApplicationProtocolNegotiator.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/OpenSslServerSessionContext.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/JettyNpnSslEngine.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/logging/LogLevel.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/SslHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/stream/ChunkedWriteHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/stream/package-info.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/ReadTimeoutHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/logging/LoggingHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/WriteTimeoutException.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/stream/ChunkedFile.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/IdleStateHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/NotSslRecordException.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/IdleStateEvent.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/package-info.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/IdleState.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/ReadTimeoutException.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/WriteTimeoutHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/package-info.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/stream/ChunkedInput.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/timeout/TimeoutException.java
jar/io/netty/handler/stream/ChunkedStream.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/logging/package-info.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 *   https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/SslHandshakeCompletionEvent.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2018 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 *   https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/OpenSslCachingX509KeyManagerFactory.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/OpenSslKeyMaterialProvider.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
Open Source Used In DNAC Platform Antelope 12863

/*
 * Copyright 2011 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
  jar/io/netty/handler/traffic/AbstractTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 * 
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * 
 * https://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
  jar/io/netty/handler/ssl/oqs/p/package-info.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
  jar/io/netty/handler/ssl/Conscrypt.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
  jar/io/netty/handler/ssl/AbstractSniHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
  jar/io/netty/handler/ssl/JdkAlpnSslUtils.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
  jar/io/netty/handler/ssl/SslClientHelloHandler.java
/opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/SslCompletionEvent.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/SniCompletionEvent.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/ocsp/OcspClientHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/OptionalSslHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/ConscryptAlpnSslEngine.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version
 * 2.0 (the "License"); you may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations under
 * the License.
 *
* /

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/flow/FlowControlHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2020 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
  * /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/pcap/EthernetPacket.java
  * /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/pcap/IPPacket.java
  * /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/pcap/UDPPacket.java
  * /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/pcap/TCPPacket.java
  * /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/pcap/PcapWriter.java
  * /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/pcap/package-info.java
  * /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/pcap/PcapHeaders.java
  * /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/pcap/PcapWriteHandler.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

Found in path(s):

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2021 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/SslContextOption.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/OpenSslContextOption.java
No license file was found, but licenses were detected in source scan.

# The Netty Project licenses this file to you under the Apache License,
# version 2.0 (the "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at:
# distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/META-INF/native-
image/io.netty/handler/native-image.properties
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
// Try the OpenJDK's proprietary implementation.

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2020 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/util/FingerprintTrustManagerFactoryBuilder.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ipfilter/IpSubnetFilterRuleComparator.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/logging/ByteBufFormat.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/address/ResolveAddressHandler.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/SslClosedEngineException.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-
jar/io/netty/handler/ssl/SslHandshakeTimeoutException.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/traffic/GlobalChannelTrafficShapingHandler.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/util/X509TrustManagerWrapper.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/OpenSslKeyMaterialManager.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/PemValue.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/PemPrivateKey.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/PemX509Certificate.java
* /opt/cola/permits/1145449572_1616156172.9/0/netty-handler-4-1-59-final-sources-jar/io/netty/handler/ssl/Java8SslUtils.java
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

1.2963 jansi-native 1.8
1.2963.1 Available under license:

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2964 jbig-kit 2.1-3.1

1.2964.1 Available under license:

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.
When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it.
either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY
11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Open Source Used In DNAC Platform Antelope 12880
You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

On Debian systems, the complete text of the GNU General Public License version 2 can be found in "/usr/share/common-licenses/GPL-2".

Files: debian/*

Copyright: 2011 Michael van der Kolff <mvanderkolff@gmail.com>
License: GPL-2+
This package is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This package is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

On Debian systems, the complete text of the GNU General Public License version 2 can be found in "/usr/share/common-licenses/GPL-2".

1.2965 jcip-annotation 1.0 1

1.2965.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2013 Stephen Connolly.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
1.2966 supports-color 2.0.0

1.2966.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2967 find-cache-dir 2.1.0

1.2967.1 Available under license:
MIT License

Copyright (c) James Talmage <james@talmage.io> (github.com/jamestalmage)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2968 read-pkg 2.0.0
1.2968.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2969 word-wrap 1.2.3 1.2.3
1.2969.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2970 etag 1.8.1 1.8.1
1.2970.1 Available under license :
(The MIT License)

Copyright (c) 2014-2016 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2971 term-size 2.2.1
1.2971.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (https://sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2972 ajv-keywords 3.4.1 3.4.1

1.2972.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.2973 jscs 0.5.0

1.2973.1 Available under license:

Copyright Mathias Bynens <http://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2974 zlib 1.2.3

1.2974.1 Available under license:

/* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.11, January 15th, 2017

Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu

* /
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

1.2975 isobject 3.0.1
1.2975.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2976 prelude.ls 1.1.2 1.1.2

1.2976.1 Available under license:

Copyright (c) George Zahariev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2977 calcite-linq4j 1.21.0-drill-r2

1.2977.1 Available under license:

# Licensed to the Apache Software Foundation (ASF) under one or more # contributor license agreements. See the NOTICE file distributed with # this work for additional information regarding copyright ownership. # The ASF licenses this file to you under the Apache License, Version 2.0 # (the "License"); you may not use this file except in compliance with # the License. You may obtain a copy of the License at # # http://www.apache.org/licenses/LICENSE-2.0 #
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Database of contributors to Apache Calcite.
# Pages such as developer.md use this data.
# List must be sorted by first name, last name.
#
- name: Alan Gates
  emeritus: 2018/05/04
  apacheId: gates
  githubId: alanfgates
  org: Hortonworks
  role: PMC
- name: Aman Sinha
  apacheId: amansinha
  githubId: amansinha100
  org: MapR
  role: PMC
- name: Andrei Sereda
  apacheId: sereda
  githubId: asereda-gs
  org:
  role: Committer
- name: Ashutosh Chauhan
  apacheId: hashutosh
  githubId: ashutoshc
  org: Hortonworks
  role: PMC
- name: Chris Baynes
  apacheId: cbaynes
  githubId: chris-baynes
  org: Contiamo
  role: Committer
- name: Christian Beikov
  apacheId: cbeikov
  githubId: beikov
  org: Blazebit
  role: Committer
- name: Chunwei Lei
  apacheId: chunwei
  githubId: chunweilei
  org: Alibaba
  role: Committer
- name: Danny Chan
  apacheId: danny0405
githubId: danny0405
org: Alibaba
role: Committer
- name: Edmon Begoli
apacheId: ebegoli
githubId: ebegoli
org: Oak Ridge National Laboratory
role: Committer
- name: Francis Chuang
apacheId: francischuang
githubId: F21
org: Boostport
role: PMC Chair
- name: Gian Merlino
apacheId: gian
githubId: gianm
org: Imply
role: Committer
- name: Haisheng Yuan
apacheId: hyuan
githubId: hsyuan
org: Alibaba
role: Committer
- name: Hongze Zhang
apacheId: hongze
githubId: zhztheplayer
org: Tencent
role: Committer
- name: James R. Taylor
apacheId: jamestaylor
githubId: JamesRTaylor
org: Salesforce
role: PMC
- name: Jacques Nadeau
apacheId: jacques
githubId: jacques-n
org: Dremio
role: PMC
- name: Jess Balint
apacheId: jbalint
githubId: jbalint
org: Stardog
role: Committer
- name: Jess Camacho Rodriguez
apacheId: jcamacho
githubId: jcamachor
org: Hortonworks
role: PMC
- name: Jinfeng Ni  
apacheId: jni  
githubId: jinfengni  
org: MapR  
role: PMC  
- name: John Pullokkaran  
apacheId: jpullokk  
githubId: jpullokkaran  
org: Hortonworks  
role: PMC  
- name: Josh Elser  
apacheId: elserj  
githubId: joshelser  
org: Hortonworks  
role: PMC  
- name: Julian Hyde  
apacheId: jhyde  
githubId: julianhyde  
org: Looker  
role: PMC  
homepage: http://people.apache.org/~jhyde  
- name: Kevin Liew  
apacheId: kliew  
githubId: kliewkliew  
org:  
role: Committer  
- name: Kevin Risden  
apacheId: krisden  
githubId: risdenk  
org:  
role: PMC  
- name: Laurent Goujon  
apacheId: laurent  
githubId: laurentgo  
org: Dremio  
role: PMC  
- name: Maryann Xue  
apacheId: maryannxue  
githubId: maryannxue  
org: Intel  
role: PMC  
- name: Michael Mior  
apacheId: mmior  
githubId: michaelmior  
org: Rochester Institute of Technology  
role: PMC  
homepage: https://michael.mior.ca/  
- name: Milinda Pathirage
- name: MinJi Kim
  apacheId: minji
  githubId: minji-kim
  org: Oracle
  role: Committer
  homepage: http://www.n10k.com

- name: Nishant Bangarwa
  apacheId: nishant
  githubId: nishantmonu51
  org: Hortonworks
  role: Committer

- name: Sergey Nuyanzin
  apacheId: snuyanzin
  githubId: snuyanzin
  org: EPAM
  role: Committer

- name: Shuyi Chen
  apacheId: shuyichen
  githubId: suez1224
  org: Uber
  role: Committer

- name: Slim Bouguerra
  apacheId: bslim
  githubId: b-slim
  org: Hortonworks
  role: Committer

- name: Stamatis Zampetakis
  apacheId: zabetak
  githubId: zabetak
  org: TIBCO
  role: PMC

- name: Steven Noels
  apacheId: stevenn
githubId: steven
org: NGData
role: PMC
- name: Ted Dunning
apacheId: tdunning
githubId: tdunning
org: MapR
role: PMC
avatar: https://mapr.com/blog/author/ted-dunning/assets/tdunning-panama.jpg
- name: Vladimir Sitnikov
apacheId: vladimirsitnikov
githubId: vlsi
org: NetCracker
role: PMC
- name: Volodymyr Vysotskyi
apacheId: volodymyr
githubId: vvysotskyi
org:
role: PMC
- name: Zhen Wang
apacheId: zhenw
githubId: zinking
org:
role: Committer
- name: Zhiqiang He
apacheId: zhiqianghe
githubId: Zhiqiang-He
org: Huawei
role: Committer
- name: Zhiwei Peng
apacheId: zhiwei
githubId: pengzhiwei2018
org: Ant Financial Group
role: Committer
- name: Zoltan Haindrich
apacheId: kgyrtkir
githubId: kgyrtkir
org: Hortonworks
role: Committer

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

-----------------------------------------------------------------------

APACHE CALCITE SUBCOMPONENTS:

The Apache Calcite project contains subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the following licenses.

-----------------------------------------------------------------------

The MIT License
-----------------------------------------------------------------------

The Apache Calcite project bundles the following files under the MIT License:

- site
  Parts of the web site generated by Jekyll (http://jekyllrb.com/)
  Copyright (c) 2008-2015 Tom Preston-Werner
- site/_sass/_font-awesome.scss
  Font-awesome css files v4.1.0 (http://fortawesome.github.io/Font-Awesome/)
  Copyright (c) 2013 Dave Gandy
- site/_sass/_normalize.scss
  normalize.css v3.0.2 | git.io/normalize
  Copyright (c) Nicolas Gallagher and Jonathan Neal
- site/_sass/_gridism.scss
  Gridism: A simple, responsive, and handy CSS grid by @cobyism
  https://github.com/cobyism/gridism
  Copyright (c) 2013 Coby Chapple
- site/js/html5shiv.min.js
  HTML5 Shiv 3.7.2 | @afarkas @jdalton @jon_neal @rem
- site/js/respond.min.js
  Respond.js v1.4.2: min/max-width media query polyfill
  Copyright 2013 Scott Jehl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----------------------------------------------------------------------

The Open Font License

The Apache Calcite project bundles the following fonts under the
SIL Open Font License (OFL) - http://scripts.sil.org/OFL/

- site/fonts/fontawesome-webfont.*
  Font-awesome font files v4.0.3 (http://fortawesome.github.io/Font-Awesome/)
  Apache Calcite
  Copyright 2012-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product is based on source code originally developed
by DynamoBI Corporation, LucidEra Inc., SQLstream Inc. and others
under the auspices of the Eigenbase Foundation
and released as the LucidDB project.

The web site includes files generated by Jekyll.

1.2978 ion-java 1.0.2
1.2978.1 Available under license :

Amazon Ion Java

Copyright 2007-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2979 is 3.3.0
1.2979.1 Available under license :
(The MIT License)

Copyright (c) 2013 Enrico Marino
Copyright (c) 2014 Enrico Marino and Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2980 @babel/plugin-transform-duplicate-keys 7.12.13

1.2980.1 Available under license:
MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2981 set-blocking 2.0.0 2.0.0

1.2981.1 Available under license:
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
1.2982 get-value 2.0.6

1.2982.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014-2016, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2983 platform 1.3.1

1.2983.1 Available under license:
Copyright 2014-2016 Benjamin Tan <https://d10.github.io/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2984 passport-local 1.0.0

1.2984.1 Available under license :

The MIT License (MIT)

Copyright (c) 2011-2014 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2985 nodemon 2.0.7

1.2985.1 Available under license :

(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (C) 2011 Peter Zotov <whitequark@whitequark.org>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE.
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)
Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

// MIT License

Copyright (C) Roman Shtylman <shtylman@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2013-2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2013 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014 Nathan LaFreniere and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors
The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Copyright (c) 2010 - present, Remy Sharp, https://remysharp.com <remy@remysharp.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright 2014 Alex Gorbatchev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2012 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schlueter

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)
Copyright (c) 2012 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2986 avatica-core 1.15.0
1.2986.1 Available under license:

Apache Calcite Avatica
Copyright 2012-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.2987 protostuff 1.7.1

1.2987.1 Available under license:

Thanks to

============

Michael Donohue

* J2ME module

Alexander Dvorkovyy

* Deployment to maven central

Ivan Prisyazhniy

* `proto_extender` (compiler/codegen)
  * `java_beans_model` (compiler/codegen)

Igor Scherbak

* `proto_extender` (compiler/codegen)

John Koepi

* `ProtoToProtoCompiler`
  * `JavaBeanModelCompiler`

Patch contributors
J.D. Zamfirescu

* `protostuff-parser` - supplied code to parse protobuf "syntax2" header. (Issue 6)

Philippe Laflamme

* `protostuff-parser` - supplied code to parse protobuf extensions and allow generated code to access its metadata. (Issue 25)

Joseph Gentle

* `protostuff-model` - fixed bug in the reflection compiler with nested protobuf types. (Issue 26)

Tim Underwood

* `protostuff-api` - ArrayIndexOutOfBoundsException in StringSerializer.writeUTF8 (Issue 27)

nordlig.ulv

* `protostuff-compiler` - Native methods for repeated fields (Issue 35)
* `protostuff-compiler` - Merge method for json protobuf classes (Issue 64)

avaskys

* `protostuff-parser`, `protostuff-compiler` - cannot import more than one .proto file from the same package (Issue 38)

sergei.s.ivanov

* `protostuff-compiler` Use FQCN everywhere in classes generated by java_v2protoc_schema (Issue 70). Maven plugin to attach generated source directory to the project compile source roots (Issue 69).

Thomas Kock

* `protostuff-uberjar` - a bundle created primarily for osgi.

thierry.momney

* `protostuff-me` - Preverification fails (Issue 92)

Leo Romanoff

* `protostuff-runtime` - user definable mapping: FQNC's --> integer, for serialized stream size-reduction (Issue 96)

Brice Jaglin
* `protostuff-runtime` - annotations on pojo's fields to have explicit control of field numbers.

baetz.alexander

* `protostuff-compiler` - Issue 128 patch
* `protostuff-maven-plugin` - Issue 128 patch

Max Lunin

* `protostuff-core` - Issue 138

Ryszard Winiewski

* `protostuff-runtime` - Issue 140 patch

lzh0379

* `protostuff-runtime` - Issue 141 patch

eedzjee

* `protostuff-runtime-md` - Issue 156 patch

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only...
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==============================================================
protostuff
Copyright 2009 David Yu dyuproject@gmail.com

 ================================================================

protobuf is copyright Google inc unless otherwise noted.
It is licensed under the BSD license.

jackson-core-asl is copyright FasterXml unless otherwise noted.
It is licensed under the apache 2.0 license.

antlr is copyright Terence Parr unless otherwise noted.
It is licensed under the BSD license.

stringtemplate is copyright Terence Parr unless otherwise noted.
It is licensed under the BSD license.
velocity is licensed under the apache 2.0 license.

B64Code.java is copyright Mort Bay Consulting Pty Ltd unless otherwise noted. It is licensed under the apache 2.0 license.

jarjar is copyright Google inc unless otherwise noted. It is licensed under the apache 2.0 license.

guava is copyright Google inc unless otherwise noted. It is licensed under the apache 2.0 license.

1.2988 cookiejar 2.0.6
1.2988.1 Available under license:
The MIT License (MIT)
Copyright (c) 2013 Bradley Meck

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2989 jsreport-studio 2.10.1
1.2989.1 Available under license:
The MIT License (MIT)
Copyright (c) 2015 Erik Rasmussen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2990 jersey-client 2.32

1.2990.1 Available under license:

# Notice for Jersey
This content is produced and maintained by the Eclipse Jersey project.

* Project home: https://projects.eclipse.org/projects/ee4j.jersey

## Trademarks
Eclipse Jersey is a trademark of the Eclipse Foundation.

## Copyright
All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code
The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/jersey

## Third-party Content
Angular JS, v1.6.6
* License MIT (http://www.opensource.org/licenses/mit-license.php)
* Project: http://angularjs.org
* Copyright: (c) 2010-2017 Google, Inc.

aopalliance Version 1
* License: all the source code provided by AOP Alliance is Public Domain.
* Copyright: Material in the public domain is not protected by copyright

Bean Validation API 2.0.2
* License: Apache License, 2.0
* Project: http://beanvalidation.org/1.1/
* Copyright: 2009, Red Hat, Inc. and/or its affiliates, and individual contributors
  * by the @authors tag.

Hibernate Validator CDI, 6.1.2.Final
* License: Apache License, 2.0
* Project: https://beanvalidation.org/
* Repackaged in org.glassfish.jersey.server.validation.internal.hibernate

Bootstrap v3.3.7
* License: MIT license (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* Project: http://getbootstrap.com
* Copyright: 2011-2016 Twitter, Inc

Google Guava Version 18.0
* License: Apache License, 2.0
* Copyright (C) 2009 The Guava Authors

javax.inject Version: 1
* License: Apache License, 2.0
* Copyright (C) 2009 The JSR-330 Expert Group

Javassist Version 3.25.0-GA
* License: Apache License, 2.0
* Project: http://www.javassist.org/
* Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.

Jackson JAX-RS Providers Version 2.10.1
* License: Apache License, 2.0
* Project: https://github.com/FasterXML/jackson-jaxrs-providers
* Copyright: (c) 2009-2011 FasterXML, LLC. All rights reserved unless otherwise indicated.

jQuery v1.12.4
* License: jquery.org/license
* Project: jquery.org
jQuery Barcode plugin 0.3
* Project: http://www.pasella.it/projects/jquery/barcode
* Copyright: (c) 2009 Antonello Pasella antonello.pasella@gmail.com

JSR-166 Extension - JEP 266
* License: CC0
* No copyright
* Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

KineticJS, v4.7.1
* License: MIT license (http://www.opensource.org/licenses/mit-license.php)
* Copyright: Eric Rowell

org.objectweb.asm Version 8.0
* License: Modified BSD (http://asm.objectweb.org/license.html)
* Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

org.osgi.core version 6.0.0
* License: Apache License, 2.0
* Copyright (c) 2005, 2008. All Rights Reserved.

org.glassfish.jersey.server.internal.monitoring.core
* License: Apache License, 2.0
* Copyright (c) 2015-2018 Oracle and/or its affiliates. All rights reserved.
* Copyright 2010-2013 Coda Hale and Yammer, Inc.

W3.org documents
* License: W3C License
* Copyright: Copyright (c) 1994-2001 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.
http://www.w3.org/Consortium/Legal/
# Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content
Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial
2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS
3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

   iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

   iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION
Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY
EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT
PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS
SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST
PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of
the remainder of the terms of this Agreement, and without further
action by the parties hereto, such provision shall be reformed to the
minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity
(including a cross-claim or counterclaim in a lawsuit) alleging that the
Program itself (excluding combinations of the Program with other software
or hardware) infringes such Recipient's patent(s), then such Recipient's
rights granted under Section 2(b) shall terminate as of the date such
litigation is filed.

All Recipient's rights under this Agreement shall terminate if it
fails to comply with any of the material terms or conditions of this
Agreement and does not cure such failure in a reasonable period of
time after becoming aware of such noncompliance. If all Recipient's
rights under this Agreement terminate, Recipient agrees to cease use
and distribution of the Program as soon as reasonably practicable.
However, Recipient's obligations under this Agreement and any licenses
granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement,
but in order to avoid inconsistency the Agreement is copyrighted and
may only be modified in the following manner. The Agreement Steward
reserves the right to publish new versions (including revisions) of
this Agreement from time to time. No one other than the Agreement
Steward has the right to modify this Agreement. The Eclipse Foundation
is the initial Agreement Steward. The Eclipse Foundation may assign the
responsibility to serve as the Agreement Steward to a suitable separate
entity. Each new version of the Agreement will be given a distinguishing
version number. The Program (including Contributions) may always be
Distributed subject to the version of the Agreement under which it was
received. In addition, after a new version of the Agreement is published,
Contributor may elect to Distribute the Program (including its
Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General
Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. 
   (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based on
the Program, the distribution of the whole must be on the terms of this
License, whose permissions for other licensees extend to the entire
whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of a
storage or distribution medium does not bring the other work under the
scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable
      source code, which must be distributed under the terms of Sections 1
      and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three
      years, to give any third party, for a charge no more than your cost
      of physically performing source distribution, a complete
      machine-readable copy of the corresponding source code, to be
      distributed under the terms of Sections 1 and 2 above on a medium
      customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to
distribute corresponding source code. (This alternative is allowed
only for noncommercial distribution and only if you received the
program in object code or executable form with such an offer, in
accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source code
means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to control
compilation and installation of the executable. However, as a special
exception, the source code distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies the
executable.
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.
It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND,
EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.
If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

---

## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to
do so. If you do not wish to do so, delete this exception statement from your version.

1.2991 archiver-utils 2.1.0

1.2991.1 Available under license:

Copyright (c) 2015 Chris Talkington.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2992 parquet-encoding 1.8.1

1.2992.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

---------------------------------------------------------------------
This project includes code from Daniel Lemire's JavaFastPFOR project in this binary artifact. The "Lemire" bit packing classes produced by parquet-generator are derived from the JavaFastPFOR project.

Copyright: 2013 Daniel Lemire
Home page: http://lemire.me/en/
Project page: https://github.com/lemire/JavaFastPFOR
License: Apache License Version 2.0 http://www.apache.org/licenses/LICENSE-2.0

Apache Parquet Encodings
Copyright 2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.2993 parseuri 0.0.6
1.2993.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Gal Koren

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.2994 drafter.js 2.6.7 2.6.7
1.2994.1 Available under license :
(The MIT License)

Copyright (c) 2015 Apiary Inc. <support@apiary.io>.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2995 aws-java-sdk-kms 1.11.704
1.2995.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with
 * the License. A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions
 * and limitations under the License.
 */

Found in path(s):
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015-2020 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with
 * the License. A copy of the License is located at
 * http://aws.amazon.com/apache2.0
 * or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
 * CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions
 * and limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/RevokeGrantRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/InvalidImportTokenException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/CreateAliasResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/GrantListEntryJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/PutKeyPolicyResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/DeleteCustomKeyStoreRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/GetKeyRotationStatusResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/EncryptResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/DisableKeyRotationResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/UpdateKeyDescriptionRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/UpdateCustomKeyStoreRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/GenerateDataKeyPairResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/ConnectCustomKeyStoreResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/RetireGrantRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/GrantListEntry.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/DecryptRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/GrantListEntry.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/RetireGrantRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources.jar/com/amazonaws/services/kms/model/transform/UpdateKeyDescriptionRequestProtocolMarshaller.java

Open Source Used In DNAC Platform Antelope 12952
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/DataKeySpec.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/TagMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/UnsupportedOperationExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/RevokeGrantResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/EnableKeyRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/DescribeCustomKeyStoresResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/UpdateAliasResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/CustomKeyStoreNotFoundException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/AliasListEntry.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/ScheduleKeyDeletionRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/UpdateKeyDescriptionResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/IncorrectTrustAnchorExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/RetireGrantRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/GrantOperation.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/CloudHsmClusterInUseExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/DisableKeyRotationRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/InvalidArnExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/GenerateDataKeyPairWithoutPlaintextRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/AWSKMSAsync.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/ListKeysResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/SignRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/DisconnectCustomKeyStoreResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/VerifyRequest.java

Open Source Used In DNAC Platform Antelope 12953
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/ListGrantsResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/GenerateDataKeyPairResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/CloudHsmClusterNotActiveException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/CreateGrantRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/DisconnectCustomKeyStoreResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/EnableKeyResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/CloudHsmClusterInvalidConfigurationException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/ImportKeyMaterialRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/EnableKeyRotationResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/RetireGrantResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/AWSKMSClientBuilder.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/AWSKMSAsyncClient.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/ListKeyPoliciesRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/CancelKeyDeletionResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/CustomKeyStoreNameInUseException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/transform/DisableKeyRotationResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/ConnectCustomKeyStoreRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-jar/com/amazonaws/services/kms/model/GenerateDataKeyPairWithoutPlaintextResult.java
<table>
<thead>
<tr>
<th>File Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jar/com/amazonaws/services/kms/model/DisableKeyResult.java</td>
<td>* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/CreateGrantResult.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/ReEncryptRequest.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/transform/UpdateCustomKeyStoreRequest.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/transform/DescribeCustomKeyStoresRequestMarshaller.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/transform/KeyMetadataJsonUnmarshaller.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/transform/CustomKeyStoresListEntryMarshaller.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/transform/ConnectionErrorCodeType.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/transform/GenerateDataKeyRequestMarshaller.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/EnableKeyRequest.java</td>
</tr>
<tr>
<td>/opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-</td>
<td>jar/com/amazonaws/services/kms/model/transform/ConnectionStateType.java</td>
</tr>
</tbody>
</table>
| /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources- | jar/com/antelope/12959/12/0
jar/com/amazonaws/services/kms/model/ExpirationModelType.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/AlreadyExistsException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/KMSInternalExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/WrappingKeySpec.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/DescribeKeyResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/Message.Type.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/UpdateKeyDescriptionRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/GenerateDataKeyWithoutPlaintextResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DisableKeyRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateGrantRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CorrectKeyMaterialExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/ListGrantsRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/GetParametersForImportResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DescribeKeyRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DeleteImportedKeyMaterialResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/DeleteCustomKeyStoreRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/KeyListEntry.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateAliasResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/LimitExceededExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/ListRetirableGrantsRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/RevokeGrantRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/KeyListEntry.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateGrantRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/IncorrectKeyMaterialExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/ListGrantsRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/GetImportRequestUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DescribeKeyRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateAliasRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/InvalidMarkerException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/GenerateDataKeyWithoutPlaintextResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/UpdateKeyDescriptionRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DeleteImportedKeyMaterialResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/GetParametersForImportResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DisableKeyRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CorrectKeyMaterialExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DescribeKeyRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/IncorrectKeyMaterialExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DescribeKeyRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CorrectKeyMaterialExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DescribeKeyRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CorrectKeyMaterialExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DescribeKeyRequestMarshaller.java

jar/com/amazonaws/services/kms/model/transform/ReEncryptResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/GetKeyPolicyRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/UntagResourceRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/ListKeysResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/InvalidMarkerExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/GetPublicKeyResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/ListAliasesRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DeleteImportedKeyMaterialRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/VerifyResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/DeleteCustomKeyStoreResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/InvalidAliasNameException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/ListResourceTagsRequestProtocolMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/InvalidGrantIdException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/Tag.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/IncorrectTrustAnchorException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/ListKeyPoliciesResult.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateCustomKeyStoreResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/InvalidAliasNameExceptionUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateCustomKeyStoreRequestMarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateKeyResultJsonUnmarshaller.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/GenerateDataKeyPairRequest.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/InvalidArnException.java
* /opt/cola/permits/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/ListResourceTagsRequest.java
jar/com/amazonaws/services/kms/model/ListGrantsRequest.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateAliasRequestProtocolMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DisableKeyRequestMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/ListKeysRequestProtocolMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CustomKeyStoreHasCMKsExceptionUnmarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/GrantConstraintsMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/AWSKMSException.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/EncryptRequest.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/GrantConstraintsJsonUnmarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/RevokeGrantRequestMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DescribeKeyRequestProtocolMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/SignRequest.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/KMSInvalidSignatureExceptionUnmarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/TagResourceResult.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateKeyRequestProtocolMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CreateKeyRequestMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/ListResourceTagsRequest.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/GetKeyPolicyResult.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/CancelKeyDeletionResultJsonUnmarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/TagResourceRequestMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/VerifyResult.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/DescribeKeyResultJsonUnmarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/transform/PutKeyPolicyRequestProtocolMarshaller.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
jar/com/amazonaws/services/kms/model/PutKeyPolicyResult.java
* /opt/cola/permissions/1137154742_1614023258.66/0/aws-java-sdk-kms-1-11-704-sources-
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

1.2998 private 0.1.8

1.2998.1 Available under license :

Copyright (c) 2014 Ben Newman <bn@cs.stanford.edu>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
1.2999.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3000 fstream 1.0.12
1.3000.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.3001 blob 0.0.5
1.3001.1 Available under license:
MIT License

Copyright (C) 2014 Rase-

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3002.1 Available under license:
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
1.3003 define-property 2.0.2

1.3003.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3004 de.huxhorn.lilith.sender 0.9.44

1.3004.1 Available under license:
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.
As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.3005 @babel/helper-module-transforms 7.13.14
1.3005.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3006 netty-codec 4.1.59.Final
1.3006.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
* */

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/DefaultHeadersImpl.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/CharSequenceValueConverter.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/ValueConverter.java

No license file was found, but licenses were detected in source scan.

/ *
* Copyright 2012 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
* */

/ *
* Written by Robert Harder and released to the public domain, as explained at
* https://creativecommons.org/licenses/publicdomain
* /
/**
* Enumeration of supported Base64 dialects.
* *
* The internal lookup tables in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/">Robert Harder's Public Domain
* Base64 Encoder/Decoder</a>.
* */

Found in path(s):
No license file was found, but licenses were detected in source scan.

~ Copyright 2012 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.

*/

/*
* Copyright 2012 The Netty Project
* 
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
* 
* https://www.apache.org/licenses/LICENSE-2.0
* 
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

/*
* Written by Robert Harder and released to the public domain, as explained at
* https://creativecommons.org/licenses/publicdomain
*/

/*
* Utility class for [@link ByteBuf] that encodes and decodes to and from
* <a href="https://en.wikipedia.org/wiki/Base64">Base64</a> notation.
* 
* The encoding and decoding algorithm in this class has been derived from
* /

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/base64/Base64.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2019 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/compression/Lz4XXHash32.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
Open Source Used In DNAC Platform

Antelope 12978
jar/io/netty/handler/codec/compression/SnappyFrameEncoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/ByteToMessageCodec.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/compression/Base64/Base64Encoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/bytes/ByteArrayDecoder.java
jar/io/netty/handler/codec/compression/JZlibDecoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/MessageToMessageEncoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/serialization/ClassResolver.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/DateFormatter.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/CodecOutputList.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/string/LineSeparator.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-jar/io/netty/handler/codec/string/LineEncoder.java

No license file was found, but licenses were detected in source scan.
/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/HeadersUtils.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/protobuf/ProtobufVarint32FrameDecoder.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/unsupported/UnsupportedValueConverter.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/protobuf/ProtobufEncoder.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/protobuf/ProtobufVarint32LengthFieldPrepender.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/ProtocolDetectionResult.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/protobuf/protobufEncoder.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/protobuf/protobufDecoderNano.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/protobuf/protobufEncoderNano.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/protobuf/protobufDecoder.java
  * /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
    jar/io/netty/handler/codec/ProtocolDetectionState.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 */
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* A decoder that splits the received {@link ByteBuf}s dynamically by the
* value of the length field in the message. It is particularly useful when you
* decode a binary message which has an integer header field that represents the
* length of the message body or the whole message.
* <p>
* {@link LengthFieldBasedFrameDecoder} has many configuration parameters so
* that it can decode any message with a length field, which is often seen in
* proprietary client-server protocols. Here are some example that will give
* you the basic idea on which option does what.
*
* <h3>2 bytes length field at offset 0, do not strip header</h3>
* <pre>
* <b>lengthFieldOffset</b>   = <b>0</b>
* <b>lengthFieldLength</b>   = <b>2</b>
* lengthAdjustment    = 0
* initialBytesToStrip = 0 (= do not strip header)
* </pre>
* The value of the length field in this example is <tt>12 (0x0C)</tt> which
* represents the length of "HELLO, WORLD". By default, the decoder assumes
* that the length field represents the number of the bytes that follows the
* length field. Therefore, it can be decoded with the simplistic parameter
* combination.
* <pre>
* BEFORE DECODE (14 bytes)         AFTER DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content |----->| Length | Actual Content |
* | 0x000C | "HELLO, WORLD" |      | 0x000C | "HELLO, WORLD" |
* +--------+----------------+      +--------+----------------+
* </pre>
* <h3>2 bytes length field at offset 0, strip header</h3>
* Because we can get the length of the content by calling
* {@link ByteBuf#readableBytes()}, you might want to strip the length
* field by specifying <tt>initialBytesToStrip</tt>. In this example, we
* specified <tt>2</tt>, that is same with the length of the length field, to
* strip the first two bytes.
* <pre>
* lengthFieldOffset = 0
* lengthFieldLength = 2
* lengthAdjustment = 0
*/
* <b>initialBytesToStrip</b> = <b>2</b> (= the length of the Length field)

* BEFORE DECODE (14 bytes)  AFTER DECODE (12 bytes)
* +--------+----------------+      +----------------+
* | Length | Actual Content |----->| Actual Content |
* | 0x000C | "HELLO, WORLD" |      | "HELLO, WORLD" |
* +--------+----------------+      +----------------+
* </pre>

* BEFORE DECODE (14 bytes)  AFTER DECODE (14 bytes)
* +----------+----------+----------------+      +----------+----------+----------------+
* | Header 1 |  Length  | Actual Content |----->| Header 1 |  Length  | Actual Content |
* | 0xCAFE   | 0x00000C | "HELLO, WORLD" |      | 0xCAFE   | 0x00000C | "HELLO, WORLD" |

* In most cases, the length field represents the length of the message body
* only, as shown in the previous examples. However, in some protocols, the
* length field represents the length of the whole message, including the
* message header. In such a case, we specify a non-zero
* <tt>lengthAdjustment</tt>. Because the length value in this example message
* is always greater than the body length by <tt>2</tt>, we specify <tt>-2</tt>
* as <tt>lengthAdjustment</tt> for compensation.

* BEFORE DECODE (14 bytes)  AFTER DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content |----->| Length | Actual Content |
* | 0x000E | "HELLO, WORLD" |      | 0x000E | "HELLO, WORLD" |

* The following message is a simple variation of the first example. An extra
* header value is prepended to the message. <tt>lengthAdjustment</tt> is zero
* again because the decoder always takes the length of the prepended data into
* account during frame length calculation.

* BEFORE DECODE (17 bytes)  AFTER DECODE (17 bytes)
* +----------+----------+----------------+      +----------+----------+----------------+
* | Header 1 |  Length  | Actual Content |----->| Header 1 |  Length  | Actual Content |
* | 0xCAFE   | 0x00000C | "HELLO, WORLD" |      | 0xCAFE   | 0x00000C | "HELLO, WORLD" |
This is an advanced example that shows the case where there is an extra header between the length field and the message body. You have to specify a positive `<tt>lengthAdjustment</tt>` so that the decoder counts the extra header into the frame length calculation.

This is a combination of all the examples above. There are the prepended header before the length field and the extra header after the length field. The prepended header affects the `<tt>lengthFieldOffset</tt>` and the extra header affects the `<tt>lengthAdjustment</tt>`. We also specified a non-zero `<tt>initialBytesToStrip</tt>` to strip the length field and the prepended header from the frame. If you don't want to strip the prepended header, you could specify `<tt>0</tt>` for `<tt>initialBytesToSkip</tt>`.

This is a combination of all the examples above. There are the prepended header before the length field and the extra header after the length field. The prepended header affects the `<tt>lengthFieldOffset</tt>` and the extra header affects the `<tt>lengthAdjustment</tt>`. We also specified a non-zero `<tt>initialBytesToStrip</tt>` to strip the length field and the prepended header from the frame. If you don't want to strip the prepended header, you could specify `<tt>0</tt>` for `<tt>initialBytesToSkip</tt>`.

This is a combination of all the examples above. There are the prepended header before the length field and the extra header after the length field. The prepended header affects the `<tt>lengthFieldOffset</tt>` and the extra header affects the `<tt>lengthAdjustment</tt>`. We also specified a non-zero `<tt>initialBytesToStrip</tt>` to strip the length field and the prepended header from the frame. If you don't want to strip the prepended header, you could specify `<tt>0</tt>` for `<tt>initialBytesToSkip</tt>`.
Let's give another twist to the previous example. The only difference from
the previous example is that the length field represents the length of the
whole message instead of the message body, just like the third example.
We have to count the length of HDR1 and Length into \texttt{lengthAdjustment}.
Please note that we don't need to take the length of HDR2 into account
because the length field already includes the whole header length.

\begin{verbatim}
\begin{small}
  lengthFieldOffset = 1
  lengthFieldLength = 2
  \texttt{lengthAdjustment} = \texttt{-3} (= the length of HDR1 + LEN, negative)
  \texttt{initialBytesToStrip} = \texttt{3}

  BEFORE DECODE (16 bytes)                       AFTER DECODE (13 bytes)
  \begin{tabular}{llllll}
    | HDR1 | Length | HDR2 | Actual Content | --- | HDR2 | Actual Content |
    \hline
    | 0xCA | 0x0010 | 0xFE | "HELLO, WORLD" |    | \textbf{0xFE} | "HELLO, WORLD" |
    \hline
  \end{tabular}
\end{small}
\end{verbatim}

\@see LengthFieldPrepender

\*/

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/LengthFieldBasedFrameDecoder.java

No license file was found, but licenses were detected in source scan.

\/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* \url{https://www.apache.org/licenses/LICENSE-2.0}
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*
*/

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/Crc32c.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/JdkZlibDecoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/xml/XmlFrameDecoder.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 *  https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/LzfDecoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/Lz4FrameEncoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/FastLz.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/FastLzFrameEncoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/FastLzFrameDecoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/Lz4Constants.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/AsciiHeadersEncoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/Bzip2HuffmanAllocator.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/Bzip2BlockDecompressor.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/Bzip2MoveToFrontTable.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/SnappyFramedDecoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/Bzip2Encoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/SnappyFramedEncoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/compression/Bzip2Decoder.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License, version 2.0 (the
 * "License"); you may not use this file except in compliance with the License. You may obtain a
 * copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License
 * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/EmptyHeaders.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/DefaultHeaders.java
* /opt/cola/permits/1145449520_1616155795.37/0/netty-codec-4-1-59-final-sources-1-
jar/io/netty/handler/codec/Headers.java

1.3007 debug 2.6.9
1.3007.1 Available under license :
Copyright (c) 2004-2015 Fabien Potencier

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished
to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3008 es6-set 0.1.5
1.3008.1 Available under license :
Copyright (C) 2013 Mariusz Nowak (www.medikoo.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3009 hk2-api 2.6.1

1.3009.1 Available under license:

# Eclipse Public License - v 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and

b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.
"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is
c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

   ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) does not attempt to limit or alter the recipients' rights
in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any
party to be under a license that satisfies the requirements
of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the
Program (i) is combined with other material in a separate file or
files made available under a Secondary License, and (ii) the initial
Contributor attached to the Source Code the notice described in
Exhibit A of this Agreement, then the Program may be made available
under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of
the Program.

3.3 Contributors may not remove or alter any copyright, patent,
trademark, attribution notices, disclaimers of warranty, or limitations
of liability ("notices") contained within the Program from any copy of
the Program which they Distribute, provided that Contributors may add
their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities
with respect to end users, business partners and the like. While this
license is intended to facilitate the commercial use of the Program,
the Contributor who includes the Program in a commercial product
offering should do so in a manner which does not create potential
liability for other Contributors. Therefore, if a Contributor includes
the Program in a commercial product offering, such Contributor
("Commercial Contributor") hereby agrees to defend and indemnify every
other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits
and other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such
Commercial Contributor in connection with its distribution of the Program
in a commercial product offering. The obligations in this section do not
apply to any claims or Losses relating to any actual or alleged
intellectual property infringement. In order to qualify, an Indemnified
Contributor must: a) promptly notify the Commercial Contributor in
writing of such claim, and b) allow the Commercial Contributor to control,
and cooperate with the Commercial Contributor in, the defense and any
related settlement negotiations. The Indemnified Contributor may
participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity
(including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.
If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

---

## The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all
its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs
If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.
This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

---

## CLASSPATH EXCEPTION

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

# Notices for Eclipse GlassFish

This content is produced and maintained by the Eclipse GlassFish project.

* Project home: https://projects.eclipse.org/projects/ee4j.glassfish

## Trademarks

Eclipse GlassFish, and GlassFish are trademarks of the Eclipse Foundation.

## Copyright

All content is the property of the respective authors or their employers. For more information regarding authorship of content, please consult the listed source code repository logs.

## Declared Project Licenses
This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at http://www.eclipse.org/legal/epl-2.0. This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License v. 2.0 are satisfied: GNU General Public License, version 2 with the GNU Classpath Exception which is available at https://www.gnu.org/software/classpath/license.html.

SPDX-License-Identifier: EPL-2.0 OR GPL-2.0 WITH Classpath-exception-2.0

## Source Code

The project maintains the following source code repositories:

* https://github.com/eclipse-ee4j/glassfish-ha-api
* https://github.com/eclipse-ee4j/glassfish-logging-annotation-processor
* https://github.com/eclipse-ee4j/glassfish-shoal
* https://github.com/eclipse-ee4j/glassfish-cdi-porting-tck
* https://github.com/eclipse-ee4j/glassfish-jsftemplating
* https://github.com/eclipse-ee4j/glassfish-hk2-extra
* https://github.com/eclipse-ee4j/glassfish-hk2
* https://github.com/eclipse-ee4j/glassfish-fighterfish

## Third-party Content

This project leverages the following third party content.

None

## Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country’s laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

### 1.3010 lodash.isplainobject 4.0.6

#### 1.3010.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.
1.3011.1 Available under license:

Copyright (c) 2013 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3012 zookeeper-jute 3.5.7

1.3012.1 Available under license:

Apache ZooKeeper - Jute
Copyright 2008-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3013 jnr-x86asm 1.0.2
1.3013.1 Available under license:

Copyright (C) 2010 Wayne Meissner
Copyright (c) 2008-2009, Petr Kobalicek <kobalicek.petr@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3014 dicer 0.2.5

1.3014.1 Available under license:

Copyright Brian White. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
1.3015 jetty-security 9.4.17.v20190418

1.3015.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE (“AGREEMENT”). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
iii) states that any provisions which differ from this Agreement are
offered by that Contributor alone and not by any other party; and
iv) states that source code for the Program is available from such
Contributor, and informs licensees how to obtain it in a reasonable
manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained
within the Program.

Each Contributor must identify itself as the originator of its Contribution,
if
any, in a manner that reasonably allows subsequent Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with
respect to end users, business partners and the like. While this license is
intended to facilitate the commercial use of the Program, the Contributor who
includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore,
if a Contributor includes the Program in a commercial product offering, such
Contributor ("Commercial Contributor") hereby agrees to defend and indemnify
every other Contributor ("Indemnified Contributor") against any losses,
damages and costs (collectively "Losses") arising from claims, lawsuits and
other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such Commercial
Contributor in connection with its distribution of the Program in a commercial
product offering. The obligations in this section do not apply to any claims
or Losses relating to any actual or alleged intellectual property
infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the
Commercial Contributor in, the defense and any related settlement
negotiations. The Indemnified Contributor may participate in any such claim at
its own expense.

For example, a Contributor might include the Program in a commercial product
offering, Product X. That Contributor is then a Commercial Contributor. If
that Commercial Contributor then makes performance claims, or offers
warranties related to Product X, those performance claims and warranties are
such Commercial Contributor's responsibility alone. Under this section, the
Commercial Contributor would have to defend claims against the other
Contributors related to those performance claims and warranties, and if a
court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue.
and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
http://www.apache.org/licenses/LICENSE-2.0.html

and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------
Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html
------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jasp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html
The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.3016 msgpack 0.6.6

1.3016.1 Available under license:

No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License);
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
// http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/MessagePackable.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/AnyTemplate.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/packer/AbstractPacker.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/builder/TemplateBuilderChain.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/type/ValueFactory.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/util/json/JSONPacker.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/MessagePackableTemplate.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/DoubleArrayTemplate.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/packer/Unconverter.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/annotation/Message.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/IntegerArrayTemplate.java
* /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-
Open Source Used In DNAC Platform Antelope 13025
jar/org/msgpack/MessageTypeException.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/util/TemplatePrecompiler.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/packer/BufferPacker.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/TemplateReference.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/unpacker/ValueAccept.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/AbstractTemplate.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/annotation/Index.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/builder/TemplateBuilder.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/FieldList.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/builder/AbstractTemplateBuilder.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/FieldOption.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/io/ByteBufferOutput.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/builder/ReflectionTemplateBuilder.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/Templates.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/GenericCollectionTemplate.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/packer/Packer.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/builder/JavassistTemplateBuilder.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/Template.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/unpacker/LongAccept.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/Templates.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/template/Template.java
  * /opt/cola/permits/1138552721_1614291037.54/0/msgpack-0-6-6-sources-1-jar/org/msgpack/packer/MessagePackPacker.java
1.3017.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Jed Watson <jed.watson@me.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3018 buffer-shims 1.0.0 1.0.0
1.3018.1 Available under license:

# Copyright (c) 2016 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.**
1.3019 typedarray 0.0.6 0.0.6

1.3019.1 Available under license:

/*
 * Copyright (c) 2010, Linden Research, Inc.
 * Copyright (c) 2012, Joshua Bell

 Permission is hereby granted, free of charge, to any person obtaining a copy
 of this software and associated documentation files (the "Software"), to deal
 in the Software without restriction, including without limitation the rights
 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 copies of the Software, and to permit persons to whom the Software is
 furnished to do so, subject to the following conditions:

 The above copyright notice and this permission notice shall be included in
 all copies or substantial portions of the Software.

 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
 THE SOFTWARE.
 */

$/LicenseInfo$

// Original can be found at:
//   https://bitbucket.org/lindenlab/llsd
// Modifications by Joshua Bell inexorabletash@gmail.com
//   https://github.com/inexorabletash/polyfill

// ES3/ES5 implementation of the Krhonos Typed Array Specification
// Ref: http://www.khronos.org/registry/typedarray/specs/latest/
// Date: 2011-02-01
//
// Variations:
// * Allows typed_array.get/set() as alias for subscripts (typed_array[])
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3021 paranamer 2.3
1.3021.1 Available under license:

* Copyright (c) 2007 Paul Hammant
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
1.3022 java-jwt 3.8.3

1.3022.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3023 jetty 9.4.17.v20190418

1.3023.1 Available under license:

This program and the accompanying materials are made available under the terms of the Eclipse Public License 2.0 which is available at http://www.eclipse.org/legal/epl-2.0, or the Apache Software License 2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:
a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution.

No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or
otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: 

a) promptly notify the Commercial Contributor in writing of such claim, and

b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under
applicable law, it shall not affect the validity or enforceability of the
remainder of the terms of this Agreement, and without further action by the
parties hereto, such provision shall be reformed to the minimum extent
necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Program itself
(excluding combinations of the Program with other software or hardware)
infringes such Recipient's patent(s), then such Recipient's rights granted
under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to
comply with any of the material terms or conditions of this Agreement and does
not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate,
Recipient agrees to cease use and distribution of the Program as soon as
reasonably practicable. However, Recipient's obligations under this Agreement
and any licenses granted by Recipient relating to the Program shall continue
and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
order to avoid inconsistency the Agreement is copyrighted and may only be
modified in the following manner. The Agreement Steward reserves the right to
publish new versions (including revisions) of this Agreement from time to
time. No one other than the Agreement Steward has the right to modify this
Agreement. The Eclipse Foundation is the initial Agreement Steward. The
Eclipse Foundation may assign the responsibility to serve as the Agreement
Steward to a suitable separate entity. Each new version of the Agreement will
be given a distinguishing version number. The Program (including
Contributions) may always be distributed subject to the version of the
Agreement under which it was received. In addition, after a new version of the
Agreement is published, Contributor may elect to distribute the Program
(including its Contributions) under the new version. Except as expressly
stated in Sections 2(a) and 2(b) above, Recipient receives no rights or
licenses to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in the
Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in
any resulting litigation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==============================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
==============================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd
unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from
following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

-----
Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

-----
Assorted

The UnixCrypt.java code implements the one way cryptography used by
Unix systems for simple password protection. Copyright 1996 Aki Yoshida,
modified April 2001 by Iris Van den Broeke, Daniel Deville.
Permission to use, copy, modify and distribute UnixCrypt
for non-commercial or commercial purposes and without fee is
granted provided that the copyright notice appears in all copies.

1.3024 grpc-protobuf-lite 1.27.1
1.3024.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such
license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
1.3025 source-map 0.5.7 0.5.7

1.3025.1 Available under license:

Copyright (c) 2009-2011, Mozilla Foundation and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the names of the Mozilla Foundation nor the names of project
  contributors may be used to endorse or promote products derived from this
  software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3026 detect-file 1.0.0
1.3026.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016-2017, Brian Woodward.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3027 jsreport-jsrender 2.1.1

1.3027.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. &lt;http://fsf.org/&gt;
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.
An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
1.3028 fast-deep-equal 1.1.0 1.1.0

1.3028.1 Available under license :

MIT License

Copyright (c) 2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3029 pixman 0.33.6 1

1.3029.1 Available under license :

The following is the MIT license, agreed upon by most contributors. Copyright holders of new code should use this license statement where possible. They may also add themselves to the list below.

/*
 * Copyright 1987, 1988, 1989, 1998 The Open Group
 * Copyright 1987, 1988, 1989 Digital Equipment Corporation
 * Copyright 1999, 2004, 2008 Keith Packard
 * Copyright 2000 SuSE, Inc.
 * Copyright 2000 Keith Packard, member of The XFree86 Project, Inc.
 * Copyright 2004 Nicholas Miell
 * Copyright 2005 Lars Knoll & Zack Rusin, Trolltech
 * Copyright 2005 Trolltech AS
 * Copyright 2007 Luca Barbato
 * Copyright 2008 Aaron Plattner, NVIDIA Corporation
 * Copyright 2008 Rodrigo Kumpera
*/
1.3030 follow-redirects 1.5.10

1.3030.1 Available under license:

Copyright 2014-present Olivier Lalonde <olalonde@gmail.com>, James Talmage <james@talmage.io>, Ruben Verborgh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR
1.3031 jwa 1.1.6 1.1.6
1.3031.1 Available under license:
Copyright (c) 2013 Brian J. Brennan

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the
Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTicular
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE
FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT
OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

1.3032 esutils 2.0.2 2.0.2
1.3032.1 Available under license:
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3033 agent-base 4.3.0
1.3033.1 Available under license:
No license file was found, but licenses were detected in source scan.

(The MIT License)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Found in path(s):
* /opt/cola/permits/1112087593_1607035045.04/0/agent-base-4-3-0-5-tgz/package/README.md

1.3034 unc-path-regex 0.1.2
1.3034.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3035 json5 0.5.1

1.3035.1 Available under license:

MIT License

Copyright (c) 2012-2016 Aseem Kishore, and [others](https://github.com/aseemk/json5/contributors).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3036 jsreport-scripts 2.6.0

1.3036.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.
"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.


The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the
Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license
document.


You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license
document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this
License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of
the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer
system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.3037 client-sessions 0.8.0
1.3037.1 Available under license :
Mozilla Public License Version 2.0
==================================

1. Definitions
--------------

1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.
1.8. "License" means this document.

1.9. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. "Modifications" means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License" means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form" means the form of the work preferred for making modifications.

1.14. "You" (or "Your") means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

----------------------------------------
2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to
distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

-------------------

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.
3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.
5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

********************************************************************
*                                                                 *
*  6. Disclaimer of Warranty                                       *
*  -------------------------                                     *
*                                                                 *
*  Covered Software is provided under this License on an "as is"  *
*  basis, without warranty of any kind, either expressed, implied, or *
*  statutory, including, without limitation, warranties that the    *
*  Covered Software is free of defects, merchantable, fit for a     *
*  particular purpose or non-infringing. The entire risk as to the *
*  quality and performance of the Covered Software is with You.     *
*  Should any Covered Software prove defective in any respect, You *
*  (not any Contributor) assume the cost of any necessary servicing, *
*  repair, or correction. This disclaimer of warranty constitutes an *
*  essential part of this License. No use of any Covered Software is *
*  authorized under this License except under this disclaimer.      *
*                                                                 *
********************************************************************
7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions
Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.
1.3038 babel-plugin-transform-es2015-block-scoping 6.26.0

1.3038.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The << Software identified by reference to the Ecma Standard* ("Software")>> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3039 axios 0.21.4 0.21.4
1.3039.1 Available under license :
Copyright (c) 2014-present Matt Zabriskie

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3040 jackson-jaxrs 2.10.4
1.3040.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.salaranta@iki.fi), and has
Open Source Used In DNAC Platform Antelope

Antelope has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

### 1.3041 indent-string 4.0.0

#### 1.3041.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.3042 home-or-tmp 2.0.0

#### 1.3042.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3043 babel-plugin-transform-property-literals 7.12.13
1.3043.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3044 js-yaml 3.13.1

1.3044.1 Available under license:
(The MIT License)

Copyright (C) 2011-2015 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3045 yargs-parser 13.1.2

1.3045.1 Available under license:
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.3046 lodash.isboolean 3.0.3

1.3046.1 Available under license:

Copyright 2012-2016 The Dojo Foundation <http://dojofoundation.org/>
Based on Underscore.js, copyright 2009-2016 Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3047 hive-spark-client 3.1.2

1.3047.1 Available under license:

Hive Spark Remote Client
Copyright 2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.3048 statuses 1.5.0
1.3048.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2016 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3049 jackson-dataformat-hocon 1.1.1
1.3049.1 Available under license:

Apache-2.0

1.3050 base64id 1.0.0
1.3050.1 Available under license:

(The MIT License)

Copyright (c) 2012-2016 Kristian Faeldt <faeldt_kristian@cyberagent.co.jp>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3051 utf-8-validate 1.2.1 1.2.1

1.3051.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3052 jansi 1.8

1.3052.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such
license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing,
shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.3053 free-type 2.6.1 0.1ubuntu2.4

1.3053.1 Available under license : 

The FreeType Project LICENSE

-----------------------------

2006-Jan-27

Copyright 1996-2002, 2006 by
David Turner, Robert Wilhelm, and Werner Lemberg

Introduction

-------------

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

  o We don't promise that this software works. However, we will be
interested in any kind of bug reports. ('as is' distribution)

- You can use this software for whatever you want, in parts or full form, without having to pay us. ('royalty-free' usage)

- You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. ('credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```""
Portions of this software are copyright <year> The FreeType Project (www.freetype.org). All rights reserved.
```

Please replace `<year>` with the value from the FreeType version you actually use.

Legal Terms
============

0. Definitions
-------------

Throughout this license, the terms 'package', 'FreeType Project', and 'FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the 'FreeType Project', be they named as alpha, beta or final release.

'You' refers to the licensee, or person using the project, where 'using' is a generic term including compiling the project's source code as well as linking it to form a 'program' or 'executable'. This program is referred to as 'a program using the FreeType engine'.

This license applies to all files distributed in the original
FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

2. Redistribution

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- Redistribution of source code must retain this license file (‘FTL.TXT’) unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.

- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.
3. Advertising

--------------

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: 'FreeType Project', 'FreeType Engine', 'FreeType library', or 'FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts

--------

There are two mailing lists related to FreeType:

- freetype@nongnu.org
  
  Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- freetype-devel@nongnu.org
  
  Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

https://www.freetype.org

--- end of FTL.TXT ---

# Files that don't get a copyright, or which are taken from elsewhere.
#
# All lines in this file are patterns, including the comment lines; this # means that e.g. 'FTL.TXT' matches all files that have this string in
# the file name (including the path relative to the current directory, 
# always starting with `.').

#
# Don't put empty lines into this file!

#
.gitignore
#
builds/unix/pkg.m4
#
docs/FTL.TXT
docs/GPLv2.TXT
#
include/freetype/internal/fthash.h
#
src/base/fthash.c
src/base/md5.c
src/base/md5.h
#
src/bdf/bdf.c
src/bdf/bdf.h
src/bdf/bdfdrivr.c
src/bdf/bdfdrivr.h
src/bdf/bdferror.h
src/bdf/bdflib.c
src/bdf/module.mk
src/bdf/README
src/bdf/rules.mk
#
src/pcf/module.mk
src/pcf/pcf.c
src/pcf/pcf.h
src/pcf/pcfdrivr.c
src/pcf/pcfdrivr.h
src/pcf/pcferror.h
src/pcf/pcfread.c
src/pcf/pcfread.h
src/pcf/pcfutil.c
src/pcf/pcfutil.h
src/pcf/README
src/pcf/rules.mk
#
src/gzip/adler32.c
src/gzip/infblock.c
src/gzip/infblock.h
src/gzip/infcodes.c
src/gzip/infcodes.h
src/gzip/inffixed.h
src/gzip/inflate.c
The FreeType 2 font engine is copyrighted work and cannot be used legally without a software license. In order to make this project usable to a vast majority of developers, we distribute it under two mutually exclusive open-source licenses.

This means that *you* must choose *one* of the two licenses described below, then obey all its terms and conditions when using FreeType 2 in any of your projects or products.

- The FreeType License, found in the file `FTL.TXT', which is similar to the original BSD license *with* an advertising clause that forces you to explicitly cite the FreeType project in your product's documentation. All details are in the license file. This license is suited to products which don't use the GNU General Public License.

  Note that this license is compatible to the GNU General Public License version 3, but not version 2.

- The GNU General Public License version 2, found in `GPLv2.TXT' (any later version can be used also), for programs which already use the GPL. Note that the FTL is incompatible with GPLv2 due to its advertisement clause.

The contributed BDF and PCF drivers come with a license similar to that of the X Window System. It is compatible to the above two licenses (see file src/bdf/README and src/pcf/README). The same holds for the files `fthash.c' and `fthash.h'; their code was part of the BDF driver in earlier FreeType versions.

The gzip module uses the zlib license (see src/gzip/zlib.h) which too is compatible to the above two licenses.

The MD5 checksum support (only used for debugging in development builds)
is in the public domain.

--- end of LICENSE.TXT ---

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

    Yoyodyne, Inc., hereby disclaims all copyright interest in the program
    `Gnomovision' (which makes passes at compilers) written by James Hacker.
This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.3054 babel-plugin-syntax-decorators 6.13.0
1.3054.1 Available under license:

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

BSD License

For "regenerator" software

Copyright (c) 2014, Facebook, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2015 Jed Watson
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3055 file-entry-cache 2.0.0
1.3055.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Roy Riojas

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Please read the different LICENSE files present in the licenses directory of
this distribution.

Camel :: Jackson
Copyright 2007-2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

Open Source Used In DNAC Platform Antelope 13106
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3057 react-dom 16.4.1 16.4.1
1.3057.1 Available under license:
MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3058 jffi 1.0.1

1.3058.1 Available under license:

libffi - Copyright (c) 1996-2009 Anthony Green, Red Hat, Inc and others.
See source files for details.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This code is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3 only, as published by the Free Software Foundation.

This code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License version 3 for more details.

You should have received a copy of the GNU Lesser General Public License version 3 along with this work. If not, see <http://www.gnu.org/licenses/>.
BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. Redistributions in binary form must reproduce
the above copyright notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse
or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT
SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY
WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.


You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)


You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.
If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.
For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.
To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but
which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source.
Regarding what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some
trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.
Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".
A contributor’s "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.
END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.
The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

1.3059 commons-beanutils 1.8.0
1.3059.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons BeanUtils
Copyright 2000-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.3060 babel-plugin-transform-runtime 6.23.0
1.3060.1 Available under license :

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

MIT License

Copyright (c) 2014-2017 Sebastian McKenzie <sebmck@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, 
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND 
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE 
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION 
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION 
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2015 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining 
a copy of this software and associated documentation files (the 
"Software"), to deal in the Software without restriction, including 
without limitation the rights to use, copy, modify, merge, publish, 
distribute, sublicense, and/or sell copies of the Software, and to 
permit persons to whom the Software is furnished to do so, subject to 
the following conditions:

The above copyright notice and this permission notice shall be 
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND 
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE 
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION 
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION 
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy 
of this software and associated documentation files (the "Software"), to deal 
in the Software without restriction, including without limitation the rights 
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom the Software is 
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3061 ws 1.1.1

1.3061.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*! 
 * ws: a node.js websocket client 
 * Copyright(c) 2011 Einar Otto Stangvik <einaros@gmail.com> 
 * MIT Licensed 
 */

Found in path(s):
/*!
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/Validation.fallback.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/WebSocket.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/BufferPool.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/index.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/Receiver.hixie.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/BufferUtil.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/Validation.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/Receiver.hixie.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/Sender.hixie.js
 * /opt/cola/permits/1110639661_1607050588.21/0/ws-1-1-1-1-tgz/package/lib/ErrorCodes.js
 No license file was found, but licenses were detected in source scan.

(The MIT License)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
1.3062 babel-plugin-transform-es2015-typeof-symbol 6.23.0

1.3062.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2014-2015 Jason Quense <jason@quense.me>

Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Software identified by reference to the Ecma Standard* ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXpress OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3063 babel-helper-create-regexp-features-plugin 7.12.17

1.3063.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3064 filter-obj 1.1.0 1.1.0

1.3064.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3065 supports-color 6.0.0

1.3065.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3066 uri-js 4.4.1 4.4.1

1.3066.1 Available under license:
Copyright 2011 Gary Court. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY GARY COURT "AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL GARY COURT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be
interpreted as representing official policies, either expressed or implied, of Gary Court.

1.3067 mongo-java-driver 3.12.4

1.3067.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright 2012 The Netty Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License);
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/connection/netty/ReadTimeoutHandler.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2005 Brian Goetz and Tim Peierls
 * Released under the Creative Commons Attribution License
 * (http://creativecommons.org/licenses/by/2.5)
 * Official home: http://www.jcip.net
 * Any republication or derived work distributed in source code form
 * must include this copyright and license notice.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/annotations/ThreadSafe.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/annotations/Immutable.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/annotations/NotThreadSafe.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *    http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/client/internal/package-info.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/client/internal/FallbackMongoIterableFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/client/internal/ClientSessionImpl.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/client/internal/Java8FindIterableImpl.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/lang/NonNullApi.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/internal/MongoIterables.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/operation/ServerVersionHelper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/CreateIndexOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/session/package-info.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/operation/BatchCursor.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/geojson/codecs/LineStringCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/ServerAddress.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/ServerAddress.java
Open Source Used In DNAC Platform Antelope 13149

jar/org/bson/codecs/pojo/ClassModelBuilder.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/MongoConfigurationException.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/model/DBCollectionUpdateOptions.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/model/GraphLookupOptions.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/MapCodecProvider.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/BsonReader.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/operation/CreateViewOperation.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/ChangeStreamIterable.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/ChangeListener.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/OperationHelper.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/model/Variable.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/model/DBCollectionUpdateOptions.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/model/DBCollectionUpdateOptions.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/model/GraphLookupOptions.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/MapCodecProvider.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/operation/CreateViewOperation.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/operation/CreateViewOperation.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/operation/UpdateUserOperation.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/operation/UpdateUserOperation.java

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/operation/UpdateUserOperation.java
Open Source Used In DNAC Platform Antelope 13151

jar/com/mongodb/internal/connection/NoOpSessionContext.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/event/ConnectionListener.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/client/model/vault/DataKeyOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/client/gridfs/codecs/GridFSFileCodecProvider.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/event/ConnectionPoolListener.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/client/model/changestream/OperationType.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/connection/ServerSettings.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/DBCollectionObjectFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/org/bson/BsonWriterSettings.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/org/bson/codec/configuration/CodecRegistry.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/operation/ListDatabasesOperation.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/operation/ChangeStreamOperation.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/internal/connection/SchemaHelper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/client/model/geojson/codecs/GeoJsonCodecProvider.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/client/model/geojson/MultiLineString.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/client/model/geojson/KeyRetriever.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/DBObjectFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/internal/connection/CommandEventSender.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/org/bson/json/JsonStreamBuffer.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/client/model/geojson/codecs/GeometryDecoderHelper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/client/model/KeyRetriever.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/com/mongodb/internal/connection/InsertMessage.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
Open Source Used In DNAC Platform Antelope 13159

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/event/ServerOpeningEvent.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/org/bson/codecs/pojo/DiscriminatorLookup.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/management/MBeanServerFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/internal/connection/AbstractByteBufBsonDocument.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/MapReduceCommand.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/internal/connection/SocketStreamHelper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/org/bson/ByteBuf.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/MongoInternalException.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/ErrorCategory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/bulk/WriteRequest.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/org/bson/json/JsonReader.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/org/bson/codecs/AtomicIntegerCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/operation/CommandResultDocumentCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/org/bson/json/JsOnJavaScriptConverter.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/internal/client/model/CountOptionsHelper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/binding/AsyncSingleServerBinding.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/internal/connection/LoggingCommandEventSender.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/internal/connection/OpCode.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/operation/DocumentHelper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/event/ServerListener.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/com/mongodb/client/MongoClientFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/org/bson/codecs/configuration/MapOfCodecsProvider.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/org/bson/codecs/StringCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources.jar/org/bson/BsonReaderMark.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
jar/org/bson/io/BasicOutputBuffer.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/connection/SocketStreamFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/pojo/PropertyModelSerializationImpl.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/MongoServerException.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/bulk/DeleteRequest.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/binding/SingleServerBinding.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/DefaultClusterableServerFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/json/ExtendedJsonBinaryConverter.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/CommandResult.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/MongoURI.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/Tag.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/pojo/annotations/BsonId.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/json/ExtendedJsonInt64Converter.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/MongoCompressor.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/connection/TlsChannelStreamFactoryFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/Document.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/JsonStringBuffer.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/selector/ServerSelector.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/MessageSettings.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/operation/package-info.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/gridfs/model/GridFSDownloadByNameOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/geojson/CoordinateReferenceSystem.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/IndexMap.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/event/ServerMonitorEventMulticaster.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
Open Source Used In DNAC Platform Antelope 13166

jar/org/bson/BasicBSONDecoder.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/AutoEncryptionSettings.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/json/JsonBooleanConverter.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/DoubleCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/FloatCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/LegacyProtocol.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/Facet.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/jndi/MongoClientFactory.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/json/ExtendedJsonInt32Converter.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/management/ConnectionPoolStatistics.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/MongoClientException.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/dns/DnsResolver.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/BucketGranularity.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/UpdateRequest.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/LazyBSONDecoder.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/gridfs/GridFSInputFile.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/BSON.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/CollationCaseFirst.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/operation/BsonArrayWrapper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/BsonBoolean.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/IndexOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/IndexOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/ObjectId.java
Open Source Used In DNAC Platform Antelope 13169
Open Source Used In DNAC Platform Antelope 13173

/jar/com/mongodb/internal/connection/ClusterableServer.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/org/bson/BSOneException.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/internal/connection/QueryMessage.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/org/bson/BsonSerializationException.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/operation/AggregateToCollectionOperation.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/org/bson/codes/BsonTypeClassMap.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/org/bson/internal/CycleDetectingCodecRegistry.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/internal/connection/AbstractMultiServerCluster.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/connection/StreamFactory.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/client/result/DeleteResult.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/MongoExecutionTimeoutException.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/ParallelScanOptions.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/org/bson/codes/pojo/Convention.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/org/bson/json/JsonTokenType.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/DBObjectCodecProvider.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/DBObjectCodecProvider.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/org/bson/codes/configuration/CodecConfigurationException.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/client/model/Filters.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/bo/Annotations/BsonIgnore.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/client/model/Accumulators.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/client/model/Accumulators.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/client/model/Accumulators.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
*/jar/com/mongodb/client/model/Accumulators.java
*/opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
Open Source Used In DNAC Platform Antelope 13175

jar/org/bson/codecs/AtomicBooleanCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/internal/FindIterableImpl.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/Block.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/DBCreateViewOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/CommandProtocolImpl.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/internal/Optional.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/diagnostics/logging/package-info.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/GetMoreProtocol.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/event/ServerMonitorListenerAdapter.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/connection/ByteBufferBsonOutput.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/event/ConnectionPoolClosedEvent.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/Java8SniSslHelper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/ParallelCollectionScanOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/io/BsonInput.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/binding/SingleConnectionReadBinding.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/internal/CryptBinding.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/MongoClient.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/MergeOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/internal/SimpleMongoClient.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/DBCallback.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/DescriptionHelper.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/ObjectIdGenerator.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/connection/Server.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/MongoOptions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
Open Source Used in DNAC Platform: Antelope 13179

- jar/com/mongodb/Function.java
- jar/com/mongodb/client/model/geojson/MultiPoint.java
- jar/com/mongodb/client/model/geojson/codecs/PolygonCodec.java
- jar/com/mongodb/LazyDBList.java
- jar/com/mongodb/client/model/PushOptions.java
- jar/com/mongodb/client/model/geojson/OverridableUuidRepresentationCodecRegistry.java
- jar/com/mongodb/client/model/geojson/MultiPoint.java
- jar/com/mongodb/connection/SplittablePayload.java
- jar/com/mongodb/unacknowledgedBulkWriteResult.java
- jar/com/mongodb/client/model/geojson/MultiPoint.java
- jar/com/mongodb/connection/netty/NettyStreamFactoryFactory.java
- jar/com/mongodb/connection/netty/MongoSocketOpenException.java
- jar/com/mongodb/connection/netty/DnsSrvRecordInitializer.java
- jar/com/mongodb/connection/netty/MongoSocketOpenException.java
- jar/com/mongodb/connection/netty/NettyStreamFactoryFactory.java
- jar/com/mongodb/connection/netty/BaseFindAndModifyOperation.java
- jar/com/mongodb/connection/netty/ResponseCallback.java
- jar/com/mongodb/event/ConnectionMessagesSentEvent.java
- jar/com/mongodb/event/ConnectionMessagesSentEvent.java
- jar/com/mongodb/operation/RenameCollectionOperation.java
- jar/com/mongodb/operation/RenameCollectionOperation.java
- jar/com/mongodb/operation/FsyncUnlockOperation.java
- jar/com/mongodb/operation/FsyncUnlockOperation.java
- jar/com/mongodb/event/ServerListenerMulticaster.java
- jar/com/mongodb/event/ServerListenerMulticaster.java
Open Source Used In DNAC Platform Antelope 13182
Open Source Used In DNAC Platform Antelope 13187
jar/com/mongodb/client/model/ReplaceOneModel.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/operation/DropCollectionOperation.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codescs/Decimal128Codec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/operation/CommandWriteOperation.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codescs/configuration/CodecProvider.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/json/ShellObjectldConverter.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/DistinctIterable.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/internal/LazyCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/client/model/Collation.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/ConnectionFactory.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright (c) 2008-2014 Atlassian Pty Ltd
 * *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * *
 * http://www.apache.org/licenses/LICENSE-2.0
 * *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/assertions/Assertions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/util/AbstractCopyOnWriteMap.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/assertions/Assertions.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/org/bson/codecs/pojo/IdPropertyModelHolder.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/org/bson/codecs/pojo/IdGenerator.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/org/bson/codecs/pojo/ConventionObjectIdGeneratorsImpl.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/org/bson/codecs/pojo/IdGenerators.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/com/mongodb/client/model/AggregationLevel.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/org/bson/codecs/BsonCodecProvider.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-
  jar/org/bson/codecs/BsonCodec.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright 2018 Cezary Bartosiak
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/jsr310/InstantCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/jsr310/Jsr310CodecProvider.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/jsr310/LocalDateTimeCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/jsr310/LocalDateCodec.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/codecs/jsr310/LocalTimeCodec.java
No license file was found, but licenses were detected in source scan.

/*
  * Copyright 2008-present MongoDB, Inc.
  *
  * Licensed under the Apache License, Version 2.0 (the "License");
  * you may not use this file except in compliance with the License.
  * You may obtain a copy of the License at
  *
  * http://www.apache.org/licenses/LICENSE-2.0
  *
  * Unless required by applicable law or agreed to in writing, software
  * distributed under the License is distributed on an "AS IS" BASIS,
  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  * See the License for the specific language governing permissions and
  * limitations under the License.
  */

/**
 * Returns an array containing all of the elements in this deque, in
 * proper sequence (from first to last element).
 *
 * <p>The returned array will be "safe" in that no references to it are
 * maintained by this deque. (In other words, this method must allocate
 * a new array). The caller is thus free to modify the returned array.
 *
 * <p>This method acts as bridge between array-based and collection-based
 * APIs.
* @return an array containing all of the elements in this deque
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/ConcurrentLinkedDeque.java

No license file was found, but licenses were detected in source scan.

不允许使用此文件，除非符合许可证。
* 您可以取得许可证副本。
* http://www.apache.org/licenses/LICENSE-2.0
* 原始工作：MIT 许可证，版权所有 (c) [2015-2018] 所有贡献者
* https://github.com/marianobarrios/tls-channel
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/TlsChannelBuilder.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/async/AsynchronousTlsChannelGroup.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/async/AsynchronousTlsChannel.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/util/Util.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/impl/BufferHolder.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/impl/ByteBufferSet.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/impl/PortNumberSet.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/impl/ExtendedAsynchronousByteChannel.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/impl/AsyncChannelGroup.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/tlschannel/TlsClientSession.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/tlschannel/TlsSession.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/tlschannel/Util.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/tlschannel/unsafe/UnsafeBufferHolder.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright 2017 Tom Bentley
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):

/*
 * Copyright 2008-present MongoDB, Inc.
 * Copyright 2010 The Guava Authors
 * Copyright 2011 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 */
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):

* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/org/bson/internal/UnsignedLongs.java
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/annotations/Beta.java

No license file was found, but licenses were detected in source scan.

Copyright 2008-present MongoDB, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Original Work: MIT License, Copyright (c) [2015-2018] all contributors
https://github.com/marianobarrios/tls-channel

Writes a sequence of bytesProduced to this channel from a subsequence of the
given buffers.

See { @link GatheringByteChannel#write(ByteBuffer[], int, int) } for more
details of the meaning of this signature.

This method behaves slightly different than the interface specification,
with respect to non-blocking responses, see { @link write(ByteBuffer) }
@param srcs  The buffers from which bytesProduced are to be retrieved
* @param offset The offset within the buffer array of the first buffer from
*    which bytesProduced are to be retrieved; must be non-negative and no
*    larger than <tt>srcs.length</tt>
* @param length The maximum number of buffers to be accessed; must be
*    non-negative and no larger than <tt>srcs.length</tt>
* &nbsp;-&nbsp;<tt>offset</tt>
* @return The number of bytesProduced written, contrary to the behavior specified
* in [ @link ByteChannel], this method never returns 0, but throws
* [ @link WouldBlockException]
* @throws IndexOutOfBoundsException If the preconditions on the <tt>offset</tt> and
*    <tt>length</tt> parameters do not hold
* @throws WouldBlockException  if the channel is in non-blocking mode and the IO operation
*    cannot be completed immediately
* @throws NeedsTaskException if the channel is not configured to run tasks automatically
*    and a task needs to be executed to complete the operation
* @throws SSLException  if the [ @link SSLEngine] throws a SSLException
* @throws IOException       if the underlying channel throws an IOException
*/
/**
* Reads a sequence of bytesProduced from this channel into a subsequence of the
* given buffers.
* *
* <p>
* See [ @link ScatteringByteChannel#read(ByteBuffer[], int, int)] for more
* details of the meaning of this signature.
* *
* <p>
* This method behaves slightly different than the interface specification,
* with respect to non-blocking responses, see [ @link #read(ByteBuffer)] for
* more details.
* *
* @param dsts  The buffers into which bytesProduced are to be transferred
* @param offset The offset within the buffer array of the first buffer into
*    which bytesProduced are to be transferred; must be non-negative and no
*    larger than <tt>dsts.length</tt>
* @param length The maximum number of buffers to be accessed; must be
*    non-negative and no larger than <tt>dsts.length</tt>
* &nbsp;-&nbsp;<tt>offset</tt>
* @return The number of bytesProduced read, or <tt>-1</tt> if the channel has
* reached end-of-stream; contrary to the behavior specified in
* in [ @link ByteChannel], this method never returns 0, but throws
* [ @link WouldBlockException]
* @throws IndexOutOfBoundsException If the preconditions on the <tt>offset</tt> and
*    <tt>length</tt> parameters do not hold
* @throws WouldBlockException  if the channel is in non-blocking mode and the IO operation
*    cannot be completed immediately
*/
* @throws NeedsTaskException if the channel is not configured to run tasks automatically
* and a task needs to be executed to complete the operation
* @throws SSLEException if the \{@link SSLEngine\} throws a SSLEException
* @throws IOException if the underlying channel throws an IOException
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1065171190_1593486559.87/0/mongo-java-driver-3-12-4-sources-jar/com/mongodb/internal/connection/tlschannel/TlsChannel.java

### 1.3068 is-absolute 1.0.0

#### 1.3068.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.
Copyright (c) 2009-2014, TJ Holowaychuk

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 1.3069 xmlgraphics-commons 2.4

#### 1.3069.1 Available under license:

Apache XML Graphics Commons
Copyright 2006-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions.
to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.3070 har-validator 5.0.3 5.0.3
1.3070.1 Available under license :
Copyright (c) 2015, Ahmad Nassri <ahmad@ahmadnassri.com>

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.3071 jetty-continuation 9.4.41.v20210516
1.3071.1 Available under license :
This program and the accompanying materials are made available under the
terms of the Eclipse Public License 2.0 which is available at
http://www.eclipse.org/legal/epl-2.0, or the Apache Software License
2.0 which is available at https://www.apache.org/licenses/LICENSE-2.0.

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution” means:
a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:
   i) changes to the Program, and
   ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution ‘originates’ from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor” means any person or entity that distributes the Program.

"Licensed Patents” mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program” means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS
a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or
otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and
b) its license agreement:
   i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
   ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
   iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
   iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program. Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner
which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must:

a) promptly notify the Commercial Contributor in writing of such claim, and
b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE
7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

any resulting litigation.
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

==================================================================
Jetty Web Container
Copyright 1995-2018 Mort Bay Consulting Pty Ltd.
==================================================================

The Jetty Web Container is Copyright Mort Bay Consulting Pty Ltd unless otherwise noted.

Jetty is dual licensed under both

* The Apache 2.0 License
  http://www.apache.org/licenses/LICENSE-2.0.html

  and

* The Eclipse Public 1.0 License

Jetty may be distributed under either license.

------
Eclipse

The following artifacts are EPL.
* org.eclipse.jetty.orbit:org.eclipse.jdt.core

The following artifacts are EPL and ASL2.
* org.eclipse.jetty.orbit:javax.security.auth.message

The following artifacts are EPL and CDDL 1.0.
* org.eclipse.jetty.orbit:javax.mail.glassfish

------

Open Source Used In DNAC Platform Antelope 13209
Oracle

The following artifacts are CDDL + GPLv2 with classpath exception.
https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

* javax.servlet:javax.servlet-api
* javax.annotation:javax.annotation-api
* javax.transaction:javax.transaction-api
* javax.websocket:javax.websocket-api

------

Oracle OpenJDK

If ALPN is used to negotiate HTTP/2 connections, then the following artifacts may be included in the distribution or downloaded when ALPN module is selected.

* java.sun.security.ssl

These artifacts replace/modify OpenJDK classes. The modifications are hosted at github and both modified and original are under GPL v2 with classpath exceptions.
http://openjdk.java.net/legal/gplv2+ce.html

------

OW2

The following artifacts are licensed by the OW2 Foundation according to the terms of http://asm.ow2.org/license.html

org.ow2.asm:asm-commons
org.ow2.asm:asm

------

Apache

The following artifacts are ASL2 licensed.

org.apache.taglibs:taglibs-standard-spec
org.apache.taglibs:taglibs-standard-impl

------

MortBay

The following artifacts are ASL2 licensed. Based on selected classes from
Open Source Used In DNAC Platform Antelope

following Apache Tomcat jars, all ASL2 licensed.

org.mortbay.jasper:apache-jsp
org.apache.tomcat:tomcat-jasper
org.apache.tomcat:tomcat-juli
org.apache.tomcat:tomcat-jsp-api
org.apache.tomcat:tomcat-el-api
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-api
org.apache.tomcat:tomcat-util-scan
org.apache.tomcat:tomcat-util

org.mortbay.jasper:apache-el
org.apache.tomcat:tomcat-jasper-el
org.apache.tomcat:tomcat-el-api

------

Mortbay

The following artifacts are CDDL + GPLv2 with classpath exception.

https://glassfish.dev.java.net/nonav/public/CDDL+GPL.html

org.eclipse.jetty.toolchain:jetty-schemas

------

Assorted

The UnixCrypt.java code implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville. Permission to use, copy, modify and distribute UnixCrypt for non-commercial or commercial purposes and without fee is granted provided that the copyright notice appears in all copies.

1.3072 resolve-dir 1.0.1

1.3072.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2016, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3073 json-schema 0.2.3 0.2.3

1.3073.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3074 commons-lang3 3.10
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Lang
Copyright 2001-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

1.3075 ansi-regex 5.0.0
1.3075.1 Available under license :
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3076 tar-stream 1.6.2
1.3076.1 Available under license :
Copyright (c) Isaac Z. Schlueter ("Author")
All rights reserved.

The BSD License
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2013, Rod Vagg (the "Original Author")
All rights reserved.

MIT +no-false-attrs License
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Distributions of all or part of the Software intended to be used by the recipients as they would use the unmodified Software, containing modifications that substantially alter, remove, or disable functionality of the Software, outside of the documented configuration mechanisms provided by the Software, shall be modified such that the Original Author's bug reporting email addresses and urls are either replaced with the contact information of the parties responsible for the changes, or removed entirely.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except where noted, this license applies to any and all software programs and associated documentation files created by the Original Author, when distributed with the Software.

1.3077 pstree-remy 1.1.8
1.3077.1 Available under license :

The MIT License (MIT)

Copyright 2019 Remy Sharp, https://remysharp.com <remy@remysharp.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3078 shelljs 0.7.8
1.3078.1 Available under license:
Copyright (c) 2012, Artur Adib <arturadib@gmail.com>
All rights reserved.

You may use this project under the terms of the New BSD license as follows:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Artur Adib nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ARTUR ADIB BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3079 argparse 1.0.10
1.3079.1 Available under license:
(The MIT License)

Copyright (C) 2012 by Vitaly Puzrin
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3080 winston-transport 3.0.1

1.3080.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015 Charlie Robbins & the contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3081 jackson-jaxrs 2.12.1

1.3081.1 Available under license:

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

1.3082 netty 4.1.59.Final

1.3082.1 Available under license:

Copyright (c) 2003-2008 Yuta Mori All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The MIT License

Copyright (c) 2009 William Kinney

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each Contribut
provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Copyright 2009-2010 Ning, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License. You may obtain a copy of
the License at https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,WITHOUT
/*
 * Copyright (c) 2004-2006, 2008, 2009, 2011 Apple Inc. All rights reserved.
 *
 * @APPLE_LICENSE_HEADER_START@
 *
 * This file contains Original Code and/or Modifications of Original Code
 * as defined in and that are subject to the Apple Public Source License
 * Version 2.0 (the 'License'). You may not use this file except in
 * compliance with the License. Please obtain a copy of the License at
 * https://www.opensource.apple.com/apsl/ and read it before using this
 * file.
 *
 * The Original Code and all software distributed under the License are
 * distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER
 * EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES,
 * INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT.
 * Please see the License for the specific language governing rights and
 * limitations under the License.
 *
 * @APPLE_LICENSE_HEADER_END@
 */

The MIT License (MIT)

Copyright (c) 2014 Cory Benfield

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright 2011, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2010-2011 Matthew J. Francis and Contributors of the jZip2 Project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of
the country from which the work is published, or (b) hereby dedicates whatever
copyright the dedicators holds in the work of authorship identified below (the
"Work") to the public domain. A certifier, moreover, dedicates any copyright
interest he may have in the associated work, and for these purposes, is
described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this
work. Certifier recognizes that his good faith efforts may not shield him from
liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to
the detriment of the Dedicator's heirs and successors. Dedicator intends this
dedication to be an overt act of relinquishment in perpetuate of all present
and future rights under copyright law, whether vested or contingent, in the
Work. Dedicator understands that such relinquishment of all rights includes
the relinquishment of all rights to enforce (by lawsuit or otherwise) those
copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be
freely reproduced, distributed, transmitted, used, modified, built upon, or
otherwise exploited by anyone for any purpose, commercial or non-commercial,
and in any way, including by methods that have not yet been invented or
conceived.

The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa
Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Netty Project
Please visit the Netty web site for more information:

* https://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
  * HOMEPAGE:
This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * https://www.slf4j.org/

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* NOTICE:
  * license/NOTICE.harmony.txt
* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * https://archive.apache.org/dist/harmony/

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://github.com/y-256/libdivsufsort

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:

* LICENSE:
  * license/LICENSE.jctools.txt (ASL2 License)
* HOMEPAGE:
  * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:
This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/protobuf

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:
This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/snappy

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jboss-remoting/jboss-marshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/caliper

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * https://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:
This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Cory Benfield. It can be obtained at:

* LICENSE:
  * license/LICENSE.hyper-hpack.txt (MIT License)
* HOMEPAGE:
  * https://github.com/python-hyper/hpack/

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Tatsuhiro Tsujikawa. It can be obtained at:

* LICENSE:
  * license/LICENSE.nghttp2-hpack.txt (MIT License)
* HOMEPAGE:
  * https://github.com/nghttp2/nghttp2/

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user has everything necessary to run the Maven build.

* LICENSE:
  * license/LICENSE.mvn-wraper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/takari/maven-wrapper

This product contains the dnsinfo.h header file, that provides a way to retrieve the system DNS configuration on
MacOS.
This private header is also used by Apple's open source
mDNSResponder (https://opensource.apple.com/tarballs/mDNSResponder/).

* LICENSE:
  * license/LICENSE.dnsinfo.txt (Apple Public Source License 2.0)
* HOMEPAGE:
  Apache License
  Version 2.0, January 2004
  https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

(BSD License: https://www.opensource.org/licenses/bsd-license)

Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Webbit nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc. (https://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, 
OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF 
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING 
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, 
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/* 
* Copyright (c) 2004-2007 QOS.ch 
* All rights reserved. 
* 
* Permission is hereby granted, free of charge, to any person obtaining 
* a copy of this software and associated documentation files (the 
* "Software"), to deal in the Software without restriction, including 
* without limitation the rights to use, copy, modify, merge, publish, 
* distribute, sublicense, and/or sell copies of the Software, and to 
* permit persons to whom the Software is furnished to do so, subject to 
* the following conditions: 
* 
* The above copyright notice and this permission notice shall be 
* included in all copies or substantial portions of the Software. 
* 
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, 
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF 
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND 
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE 
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION 
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION 
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. 
*/ 

Protocol Buffers - Google's data interchange format
Copyright 2013 Google Inc. All rights reserved.
https://developers.google.com/protocol-buffers/

Redistribution and use in source and binary forms, with or without 
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright 
  notice, this list of conditions and the following disclaimer. 
* Redistributions in binary form must reproduce the above 
  copyright notice, this list of conditions and the following disclaimer 
  in the documentation and/or other materials provided with the 
  distribution. 
* Neither the name of Google Inc. nor the names of its 
  contributors may be used to endorse or promote products derived from 
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

This copy of Aalto XML processor is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

https://www.apache.org/licenses/

A copy is also included with both the the downloadable source code package and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/asl"; and in jar "META-INF/"

Apache Harmony

Copyright 2006, 2010 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

1.3083 codemodell 2.6

1.3084 hoist-non-react-statics 2.5.5 2.5.5

1.3084.1 Available under license:

Software License Agreement (BSD License)

=====================================================================

Copyright (c) 2015, Yahoo! Inc. All rights reserved.

=====================================================================
Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Yahoo! Inc. nor the names of YUI's contributors may be used to endorse or promote products derived from this software without specific prior written permission of Yahoo! Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3085 os-homedir 1.0.2
1.3085.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
1.3086 babylon 6.18.0 6.18.0
1.3086.1 Available under license :
Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3087 babel-preset-env 1.7.0
1.3087.1 Available under license :
MIT License

Copyright (c) 2016-2017 Babel

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3088 tiny-inflate 1.0.3
1.3088.1 Available under license:
MIT License

Copyright (c) 2015-present Devon Govett

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3089 os-tmpdir 1.0.2
1.3089.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3090 commons-configuration 1.6

1.3090.1 Available under license:

Apache Commons Configuration
Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.
END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3091 tsconfig-paths 3.9.0

1.3091.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016 Jonas Kello

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3092 highlight-words-core 1.2.2 1.2.2
1.3092.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Treasure Data

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3093 number-is-nan 1.0.1 1.0.1
1.3093.1 Available under license :
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3094 @babel/plugin-transform-typeof-symbol 7.12.13

1.3094.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3095 cookie 0.3.1 0.3.1

1.3095.1 Available under license :
(The MIT License)

Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3096 curator-client 5.1.0
1.3096.1 Available under license :

Curator Client
Copyright 2011-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.3097 psl 1.8.0

1.3097.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017 Lupo Montero lupomontero@gmail.com
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3098 base64-arraybuffer 0.1.5

1.3098.1 Available under license:

Copyright (c) 2012 Niklas von Hertzen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3099 content-type 1.0.4 1.0.4
1.3099.1 Available under license:
(The MIT License)

Copyright (c) 2015 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3100 lodash 4.17.20
1.3100.1 Available under license:
Software License Agreement (BSD License)

Copyright (c) 2007, Parakey Inc.
All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Parakey Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2010-2016 Jeremy Ashkenas, DocumentCloud

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright OpenJS Foundation and other contributors <https://openjsf.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash
The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

=====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

Copyright (c) 2009-2016 Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3101 kafka-metadata 2.8.0

1.3101.1 Available under license:

Apache Kafka
Copyright 2021 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

This distribution has a binary dependency on jersey, which is available under the CDDL License. The source code of jersey can be found at https://github.com/jersey/jersey/.

The streams-scala (streams/streams-scala) module was donated by Lightbend and the original code was copyrighted by them:

Copyright (C) 2018 Lightbend Inc. <https://www.lightbend.com>
Copyright (C) 2017-2018 Alexis Seigneurin.

This project contains the following code copied from Apache Hadoop:
clients/src/main/java/org/apache/kafka/common/utils/PureJavaCrc32C.java
Some portions of this file Copyright (c) 2004-2006 Intel Corporation and licensed under the BSD license.

This project contains the following code copied from Apache Hive:
streams/src/main/java/org/apache/kafka/streams/state/internals/Murmur3.java

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the
2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, 
unless required by applicable law (such as deliberate and grossly 
negligent acts) or agreed to in writing, shall any Contributor be 
liable to You for damages, including any direct, indirect, special, 
incidental, or consequential damages of any character arising as a 
result of this License or out of the use or inability to use the 
Work (including but not limited to damages for loss of goodwill, 
work stoppage, computer failure or malfunction, or any and all 
other commercial damages or losses), even if such Contributor 
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing 
the Work or Derivative Works thereof, You may choose to offer, 
and charge a fee for, acceptance of support, warranty, indemnity, 
or other liability obligations and/or rights consistent with this 
License. However, in accepting such obligations, You may act only 
on Your own behalf and on Your sole responsibility, not on behalf 
of any other Contributor, and only if You agree to indemnify, 
defend, and hold each Contributor harmless for any liability 
incurred by, or claims asserted against, such Contributor by reason 
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following 
boilerplate notice, with the fields enclosed by brackets "[]" 
replaced with your own identifying information. (Don't include 
the brackets!) The text should be enclosed in the appropriate 
comment syntax for the file format. We also recommend that a 
file or class name and description of purpose be included on the 
same "printed page" as the copyright notice for easier 
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); 
you may not use this file except in compliance with the License. 
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software 
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.
1.3102 icu 44

1.3102.1 Available under license:

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories
http://www.unicode.org/Public/, http://www.unicode.org/reports/, and
http://www.unicode.org/cldr/data/. Unicode Software includes any source code
published in the Unicode Standard or under the directories
http://www.unicode.org/Public/, http://www.unicode.org/reports/, and
http://www.unicode.org/cldr/data/.

NOTICE TO USER: Carefully read the following legal agreement. BY
DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES
("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND
AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU
DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES
OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright 1991-2009 Unicode, Inc. All rights reserved. Distributed under
the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy
of the Unicode data files and any associated documentation (the "Data Files") or
Unicode software and any associated documentation (the "Software") to deal in
the Data Files or Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, and/or sell copies of
the Data Files or Software, and to permit persons to whom the Data Files or
Software are furnished to do so, provided that (a) the above copyright notice(s)
and this permission notice appear with all copies of the Data Files or Software,
(b) both the above copyright notice(s) and this permission notice appear in
associated documentation, and (c) there is clear notice in each modified Data
File or in the Software as well as in the documentation associated with the Data
File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD
PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR
SOFTWARE.
Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

## -*-makefile-*-

## BSD-specific setup (FreeBSD, OpenBSD, NetBSD, *BSD)
## Copyright (c) 1999-2009, International Business Machines Corporation and ## others. All Rights Reserved.

## Commands to generate dependency files
GEN_DEPS.c=$(CC) -E -MM $(DEFS) $(CPPFLAGS)
GEN_DEPS.cc=$(CXX) -E -MM $(DEFS) $(CPPFLAGS)

## Flags for position independent code
SHAREDLIBCFLAGS = -fPIC
SHAREDLIBCXXFLAGS = -fPIC
SHAREDLIBCPPFLAGS = -DPIC

## Additional flags when building libraries and with threads
THREADSCPPFLAGS = -D_REENTRANT
LIBCPPFLAGS =

## Compiler switch to embed a runtime search path
LD_RPATH=
LD_RPATH_PRE=-Wl,-rpath,

## Compiler switch to embed a library name
LD_SONAME = -Wl,-soname -Wl,$(notdir $(MIDDLE_SO_TARGET))

## Shared library options
LD_SOOPTIONS= -Wl,-Bsymbolic

## Shared object suffix
SO = so
## Non-shared intermediate object suffix
STATIC_O = ao

## Compilation rules
%.$(STATIC_O): $(srcdir)/%.c $(STATICCPPFLAGS) $(STATICCFLAGS) -o $@ $<
%.o: $(srcdir)/%.c $(DYNAMICCPPFLAGS) $(DYNAMICCFLAGS) -o $@ $<

%.$(STATIC_O): $(srcdir)/%.cpp $(STATICCPPFLAGS) $(STATICCXXFLAGS) -o $@ $<
%.o: $(srcdir)/%.cpp $(DYNAMICCPPFLAGS) $(DYNAMICCXXFLAGS) -o $@ $<
Dependency rules

%.d: $(srcdir)/%.c
@echo "generating dependency information for $<"
@(SHELL) -ec '$(GEN_DEPS.c) $< \
| sed "\$(\$\*)\.o\[ :\]*/\$@ : /g" > $@;\n[ -s $@ ] || rm -f $@

%.d: $(srcdir)/%.cpp
@echo "generating dependency information for $<"
@(SHELL) -ec '$(GEN_DEPS.cc) $< \
| sed "\$(\$\*)\.o\[ :\]*/\$@ : /g" > $@;\n[ -s $@ ] || rm -f $@

Versioned libraries rules

%.$(SO).$(SO_TARGET_VERSION_MAJOR): %.$(SO).$(SO_TARGET_VERSION)
$(RM) $@ && ln -s ${<F} $@
%.$(SO): %.$(SO).$(SO_TARGET_VERSION_MAJOR)
$(RM) $@ && ln -s ${*F}.$(SO).$(SO).$(SO_TARGET_VERSION_MAJOR) $@

End FreeBSD-specific setup

1.3103 http-signature 1.2.0 1.2.0

1.3103.1 Available under license:

Copyright Joyent, Inc. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
1.3104 commons-logging 1.2
1.3104.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2001-2006 The Apache Software Foundation.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
  jar/org/apache/commons/logging/LogFactory.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2001-2004,2006 The Apache Software Foundation.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
  jar/org/apache/commons/logging/impl/AvalonLogger.java
No license file was found, but licenses were detected in source scan.
/*
 * Copyright 2004 The Apache Software Foundation.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

2004 The Apache Software Foundation.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
* /opt/cola/permits/1135840457_1613613080.13/0/commons-logging-1-1-sources-9-
  jar/org/apache/commons/logging/impl/WeakHashtable.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2001-2004 The Apache Software Foundation.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 */
1.3105 libproxy 0.4.11-5ubuntu1

1.3105.1 Available under license:

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of
this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.
For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)
"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library’s complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.
(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.3106 getpass 0.1.7

1.3106.1 Available under license:
Copyright Joyent, Inc. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3107 batik-svg-dom 1.13

1.3107.1 Available under license:

No license file was found, but licenses were detected in source scan.

<!-- ................................................................. -->
<!-- SVG 1.1 Core Attribute Module ................................. -->
<!-- file: svg-core-attrib.mod -->

This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-core-attrib.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Core Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-core-attrib.mod"

................................................................. -->

<!-- Core Attribute
id, xml:base, xml:lang, xml:space

This module defines the core set of attributes that can be present on any element. -->

<!--ENTITY % SVG.id.attrib
"id ID #IMPLIED"
>

<!--ENTITY % SVG.base.attrib
"xml:base %URI.datatype; #IMPLIED"
>

<!--ENTITY % SVG.lang.attrib
"xml:lang %LanguageCode.datatype; #IMPLIED"
>
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-profile.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Color Profile//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-profile.mod"

This module declares markup to provide support for color profile.
This is SVG Tiny, a proper subset of SVG.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg11-tiny-attribs.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Tiny Attribute Collection//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-tiny-attribs.mod"

This module defines the set of common attributes that can be present
on many SVG elements.

<!ENTITY % ExtensionList.datatype "CDATA" >
<!ENTITY % FeatureList.datatype "CDATA" >

<!ENTITY % SVG.Conditional.extra.attrib "" >
<!ENTITY % SVG.Conditional.attrib
"requiredFeatures %FeatureList.datatype; #IMPLIED
requiredExtensions %ExtensionList.datatype; #IMPLIED
systemLanguage %LanguageCodes.datatype; #IMPLIED
%SVG.Conditional.extra.attrib;" >

<!ENTITY % ClassList.datatype "CDATA" >
<!ENTITY % StyleSheet.datatype "CDATA" >

<!ENTITY % SVG.Style.extra.attrib "" >
<!ENTITY % SVG.Style.attrib
"style %StyleSheet.datatype; #IMPLIED
class %ClassList.datatype; #IMPLIED
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-cursor.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

```xml
PUBLIC "-//W3C//ELEMENTS SVG 1.1 Cursor//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-cursor.mod"
```

This module declares markup to provide support for cursor.

```xml
<cursor/>
```

This module defines the `cursor` property/attribute value (e.g., 'crosshair', `<uri>`).
SVG 1.1 Tiny DTD

This is SVG Tiny, a proper subset of SVG.

The Scalable Vector Graphics (SVG)
(Massachusetts Institute of Technology, Institut National de
Recherche en Informatique et en Automatique, Keio University).
All Rights Reserved.

Permission to use, copy, modify and distribute the SVG DTD and its
accompanying documentation for any purpose and without fee is hereby
granted in perpetuity, provided that the above copyright notice and
this paragraph appear in all copies. The copyright holders make no
representation about the suitability of the DTD for any purpose.

It is provided "as is" without expressed or implied warranty.

Author: Jun Fujisawa <fujisawa.jun@canon.co.jp>
Revision: $Id: svg11

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
  jar/org/apache/batik/dom/svg/resources/svg11-tiny.dtd
No license file was found, but licenses were detected in source scan.
PUBLIC "-//W3C//ENTITIES SVG 1.1 Basic Graphics Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-basic-graphics-attrib.mod"

....................................................................... -->

<!-- Basic Graphics Attribute

display, visibility

This module defines the Graphics attribute set.
-->

<!ENTITY % SVG.display.attrib
"display ( inline | block | list-item | run-in | compact | marker |
        table | inline-table | table-row-group | table-header-group |
        table-footer-group | table-row | table-column-group |
        table-column | table-cell | table-caption | none | inherit ) #IMPLIED"
>

<!ENTITY % SVG.visibility.attrib
"visibility ( visible | hidden | inherit ) #IMPLIED"
>

<!ENTITY % SVG.Graphics.extra.attrib "">

<!ENTITY % SVG.Graphics.attrib
"%SVG.display.attrib;
%SVG.visibility.attrib;
%SVG.Graphics.extra.attrib;"
>

<!-- end of svg-basic-graphics-attrib.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
  jar/org/apache/batik/dom/svg/resources/svg-basic-graphics-attrib.mod

No license file was found, but licenses were detected in source scan.

<!-- ................................................................. -->
<!-- SVG 1.1 Modular Framework Module ............................. -->
<!-- file: svg-framework.mod

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-framework.mod 1733416 2016-03-03 07:07:13Z gadams $
This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Modular Framework//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-framework.mod"

....................................................................... -->

<!-- Modular Framework
This module instantiates the modules needed to support the SVG modularization model, including:

+ Datatypes
+ Qualified Name
+ Document Model
+ Attribute Collection
-->

<!ENTITY % svg-datatypes.module "INCLUDE" >
<![%svg-datatypes.module;]
<!ENTITY % svg-datatypes.mod
PUBLIC "-//W3C//ENTITIES SVG 1.1 Datatypes//EN"
"svg-datatypes.mod" >
%svg-datatypes.mod;]]>

<!ENTITY % svg-qname.module "INCLUDE" >
<![%svg-qname.module;]
<!ENTITY % svg-qname.mod
PUBLIC "-//W3C//ENTITIES SVG 1.1 Qualified Name//EN"
"svg-qname.mod" >
%svg-qname.mod;]]>

<!ENTITY % svg-model.module "INCLUDE" >
<![%svg-model.module;]
<!-- instantiate the Document Model declared in the DTD driver -->
%svg-model.mod;]]>

<!ENTITY % svg-attribs.module "INCLUDE" >
<![%svg-attribs.module;]
<!-- instantiate the Attribute Collection declared in the DTD driver -->
%svg-attribs.mod;]]>

<!-- end of svg-framework.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/resources/svg-framework.mod
No license file was found, but licenses were detected in source scan.

<!-- ................................................................. -->
<!-- SVG 1.1 Structure Module ...................................... -->
<!-- file: svg-structure.mod
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-structure.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-/W3C/ELEMENTS SVG 1.1 Structure//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-structure.mod"

................................................................. -->

<!-- Structure
svg, g, defs, desc, title, metadata, symbol, use

This module declares the major structural elements and their attributes.
-->

<!-- Qualified Names (Default) ......................... -->
<!-- Attribute Collections (Default) .................... -->

<!ENTITY % SVG.svg.qname "svg" >
<!ENTITY % SVG.g.qname "g" >
<!ENTITY % SVG.defs.qname "defs" >
<!ENTITY % SVG.desc.qname "desc" >
<!ENTITY % SVG.title.qname "title" >
<!ENTITY % SVG.metadata.qname "metadata" >
<!ENTITY % SVG.symbol.qname "symbol" >
<!ENTITY % SVG.use.qname "use" >
<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Container.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Marker.attrib "" >
<!ENTITY % SVG.Profile.attrib "" >
<!ENTITY % SVG.Gradient.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.FilterColor.attrib "" >
<!ENTITY % SVG.DocumentEvents.attrib "" >
<!ENTITY % SVG.GraphicalEvents.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.XLinkEmbed.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Description.class ..................................... -->
<!ENTITY % SVG.Description.extra.class "" >
<!ENTITY % SVG.Description.class
"%SVG.desc.qname; | %SVG.title.qname; | %SVG.metadata.qname;
 %SVG.Description.extra.class;" >

<!-- SVG.Use.class ..................................... -->
<!ENTITY % SVG.Use.extra.class "" >
<!ENTITY % SVG.Use.class
"| %SVG.use.qname; %SVG.Use.extra.class;" >

<!-- SVG.Structure.class ..................................... -->
<!ENTITY % SVG.Structure.extra.class "" >
<!ENTITY % SVG.Structure.class
"| %SVG.svg.qname; | %SVG.g.qname; | %SVG.defs.qname; | %SVG.symbol.qname;
 %SVG.Use.class; %SVG.Structure.extra.class;" >

<!-- SVG.Presentation.attrib ..................................... -->
<!ENTITY % SVG.Presentation.extra.attrib "" >
<!ENTITY % SVG.Presentation.attrib
"%SVG.Container.attrib;
 %SVG.Viewport.attrib;
 %SVG.Text.attrib;" >
<ENTITY % SVG.svg.extra.content "" >
<!ENTITY % SVG.svg.element "INCLUDE" >
<!ENTITY % SVG.svg.content "( %SVG.Description.class; | %SVG.Animation.class; %SVG.Structure.class; %SVG.Conditional.class; %SVG.Image.class; %SVG.Style.class; %SVG.Shape.class; %SVG.Text.class; %SVG.Marker.class; %SVG.Profile.class; %SVG.Gradient.class; %SVG.Pattern.class; %SVG.Clip.class; %SVG.Mask.class; %SVG.Filter.class; %SVG.Cursor.class; %SVG.Hyperlink.class; %SVG.View.class; %SVG.Script.class; %SVG.Font.class; %SVG.svg.extra.content; )*" >
<!ELEMENT %SVG.svg.qname; %SVG.svg.content; >
<!!-- end of SVG.svg.element -->>
<!ENTITY % SVG.svg.attlist "INCLUDE" >
<!ENTITY %SVG.svg.qname; %SVG.svg.content; >
<!ENTITY %SVG.svg.attlist;[
%SVGxmlns.attrib;
%SVG.Core.attrib;
%SVG.Conditional.attrib;
%SVG.Style.attrib;
%SVG.Presentation.attrib;
%SVG.DocumentEvents.attrib;
%SVG.GraphicalEvents.attrib;
]>

Open Source Used In DNAC Platform Antelope 13304
dimensional graphics in XML.

The Scalable Vector Graphics (SVG)
(Massachusetts Institute of Technology, Institut National de
Recherche en Informatique et en Automatique, Keio University).
All Rights Reserved.

Permission to use, copy, modify and distribute the SVG DTD and its accompanying documentation for any purpose and without fee is hereby granted in perpetuity, provided that the above copyright notice and this paragraph appear in all copies. The copyright holders make no representation about the suitability of the DTD for any purpose.

It is provided "as is" without expressed or implied warranty.

Author: Jun Fujisawa <fujisawa.jun@canon.co.jp>
Revision: $Id: svg11

dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg11

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/resources/svg11-flat.dtd
No license file was found, but licenses were detected in source scan.

<!-- ............................................................... -->
<!!-- SVG 1.1 Basic Structure Module .......................... -->
<!!-- file: svg-basic-structure.mod

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-basic-structure.mod 1733416 2016-03-03 07:07:13Z gadams $
This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Basic Structure//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-basic-structure.mod"

<!-- Basic Structure

svg, g, defs, desc, title, metadata, use

This module declares the major structural elements and their attributes.

-->

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.svg.qname "svg" >
<!ENTITY % SVG.g.qname "g" >
<!ENTITY % SVG.defs.qname "defs" >
<!ENTITY % SVG.desc.qname "desc" >
<!ENTITY % SVG.title.qname "title" >
<!ENTITY % SVG.metadata.qname "metadata" >
<!ENTITY % SVG.use.qname "use" >

<!-- Attribute Collections (Default) ................... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Container.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Marker.attrib "" >
<!ENTITY % SVG.Profile.attrib "" >
<!ENTITY % SVG.Gradient.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.FilterColor.attrib "" >
<!ENTITY % SVG.DocumentEvents.attrib "" >
<!ENTITY % SVG.GraphicalEvents.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-shape.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-/W3C//ELEMENTS SVG 1.1 Shape//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-shape.mod"

...... .......... ....... . -->

Shape

path, rect, circle, line, ellipse, polyline, polygon

This module declares markup to provide support for graphical shapes.

-->

<Link to the end of the file>
<!ENTITY % SVG.circle.qname "circle" >
<!ENTITY % SVG.line.qname "line" >
<!ENTITY % SVG.ellipse.qname "ellipse" >
<!ENTITY % SVG.polyline.qname "polyline" >
<!ENTITY % SVG.polygon.qname "polygon" >

<!-- Attribute Collections (Default) ................. -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Marker.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.GraphicalEvents.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Shape.class ..................................

<!ENTITY % SVG.Shape.extra.class "" >

<!ENTITY % SVG.Shape.class
"| %SVG.path.qname; | %SVG.rect.qname; | %SVG.circle.qname;
| %SVG.line.qname; | %SVG.ellipse.qname; | %SVG.polyline.qname;
| %SVG.polygon.qname; %SVG.Shape.extra.class;"
>

<!-- path: Path Element ................................

<!ENTITY % SVG.path.extra.content "" >

<!ENTITY % SVG.path.element "INCLUDE" >
<![%SVG.path.element:]
<!ENTITY % SVG.path.content
"(( %SVG.Description.class; )*; ( %SVG.Animation.class;
   %SVG.path.extra.content; )*)"
>
<!ELEMENT %SVG.path.qname; %SVG.path.content; >
<!!-- end of SVG.path.element -->]>

<!ENTITY % SVG.path.attlist "INCLUDE" >
<![%SVG.path.attlist:]
%SVG.Cursor.attrib;
%SVG.External.attrib;
x %Coordinate.datatype; #IMPLIED
y %Coordinate.datatype; #IMPLIED
width %Length.datatype; #REQUIRED
height %Length.datatype; #REQUIRED
rx %Length.datatype; #IMPLIED
ry %Length.datatype; #IMPLIED
transform %TransformList.datatype; #IMPLIED
>
<![-- end of SVG.rect.attlist --]>>

<!-- circle: Circle Element ................................ -->

<!ENTITY % SVG.circle.extra.content "" >

<!ENTITY % SVG.circle.element "INCLUDE" >
<![%SVG.circle.element;[
<!ENTITY % SVG.circle.content
"(( %SVG.Description.class; )*, ( %SVG.Animation.class;
 %SVG.circle.extra.content; )*)"
>
<![-- end of SVG.circle.qname; %SVG.circle.content; -->
<![-- end of SVG.circle.element --]]>}

<!ENTITY % SVG.circle.attlist "INCLUDE" >
<![%SVG.circle.attlist;[
<!ATTLIST %SVG.circle.qname; %SVG.Core.attrib;
%SVG.Conditional.attrib;
%SVG.Style.attrib;
%SVG.Paint.attrib;
%SVG.Color.attrib;
%SVG.Opacity.attrib;
%SVG.Graphics.attrib;
%SVG.Clip.attrib;
%SVG.Mask.attrib;
%SVG.Filter.attrib;
%SVG.GraphicalEvents.attrib;
%SVG.Cursor.attrib;
%SVG.External.attrib;
cx %Coordinate.datatype; #IMPLIED
cy %Coordinate.datatype; #IMPLIED
r %Length.datatype; #REQUIRED
transform %TransformList.datatype; #IMPLIED
>
<![-- end of SVG.circle.attlist --]]>
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-gradient.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Gradient//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-gradient.mod"

....................................................................... -->

<!-- Gradient

linearGradient, radialGradient, stop

This module declares markup to provide support for gradient fill. -->

<!-- a <number> or a <percentage> -->
<!ENTITY % NumberOrPercentage.datatype "CDATA" >

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.linearGradient.qname "linearGradient" >
<!ENTITY % SVG.radialGradient.qname "radialGradient" >
<!ENTITY % SVG.stop.qname "stop" >

<!-- Attribute Collections (Default) ............... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.XLink.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Gradient.class ......................... -->
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-marker.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Marker//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-marker.mod"

This module declares markup to provide support for marker.

marker

This module declares markup to provide support for marker.
<!-- Attribute Collections (Default) ....................... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Container.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Profiles.attrib "" >
<!ENTITY % SVG.Gradient.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.FilterColor.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Marker.class ........................................ -->

<!ENTITY % SVG.Marker.extra.class "" >

<!ENTITY % SVG.Marker.class
" | %SVG.marker.qname; %SVG.Marker.extra.class;"
>

<!-- SVG.Marker.attrib ....................................... -->

<!ENTITY % SVG.Marker.extra.attrib "" >

<!ENTITY % SVG.Marker.attrib
"marker-start %MarkerValue.datatype; #IMPLIED
marker-mid %MarkerValue.datatype; #IMPLIED
marker-end %MarkerValue.datatype; #IMPLIED
%SVG.Marker.extra.attrib;"
>

<!-- SVG.Presentation.attrib ................................ -->

<!ENTITY % SVG.Presentation.extra.attrib "" >

<!ENTITY % SVG.Presentation.attrib
"%SVG.Container.attrib;
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-extensibility.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Extensibility//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-extensibility.mod"

................................................................. -->

<!--- Extensibility

foreignObject

This module declares markup to provide support for extensibility. -->

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.foreignObject.qname "foreignObject" >

<!-- Attribute Collections (Default) ................. -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
flood-color %SVGColor.datatype; #IMPLIED
flood-opacity %OpacityValue.datatype; #IMPLIED
lighting-color %SVGColor.datatype; #IMPLIED
 %SVG.Presentation.extra.attrib;"
>

<!-- foreignObject: Foreign Object Element .......... -->

<!ENTITY % SVG.foreignObject.extra.content "" >

<!ENTITY % SVG.foreignObject.element "INCLUDE" >
<![%SVG.foreignObject.element;]
<!ENTITY % SVG.foreignObject.content
 "(#PCDATA %SVG.foreignObject.extra.content: )*" 
> 
<!ELEMENT %SVG.foreignObject.qname; %SVG.foreignObject.content; >
<!-- end of SVG.foreignObject.element -->]

<!ENTITY % SVG.foreignObject.attlist "INCLUDE" >
<![%SVG.foreignObject.attlist;]
<!ATTLIST %SVG.foreignObject.qname;
 %SVG.Core.attrib;
 %SVG.Conditional.attrib;
 %SVG.Style.attrib;
 %SVG.Presentation.attrib;
 %SVG.GraphicalEvents.attrib;
 %SVG.External.attrib;
x %Coordinate.datatype; #IMPLIED
y %Coordinate.datatype; #IMPLIED
width %Length.datatype; #REQUIRED
height %Length.datatype; #REQUIRED
transform %TransformList.datatype; #IMPLIED
>
<!-- end of SVG.foreignObject.attlist -->]

<!-- end of svg-extensibility.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1- jar/org/apache/batik/dom/svg/resources/svg-extensibility.mod
No license file was found, but licenses were detected in source scan.

<!-- ................................................................. -->
<!-- SVG 1.1 Mask Module .......................................... -->
<!-- file: svg-mask.mod

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Mask//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-mask.mod"

<!-- Mask

This module declares markup to provide support for masking. -->

<!-- 'mask' property/attribute value (e.g., 'none', <uri>) -->
<!ENTITY % MaskValue.datatype "CDATA" >

<!-- Qualified Names (Default) ......................... -->
<!ENTITY % SVG.mask.qname "mask" >

<!-- Attribute Collections (Default) ................. -->
<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Container.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Marker.attrib "" >
<!ENTITY % SVG.Profile.attrib "" >
<!ENTITY % SVG.Gradient.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.FilterColor.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Mask.class .................................... -->
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg11-model.mod 1733416 2016-03-03 07:07:13Z gadams $ 

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Document Model//EN"

Open Source Used In DNAC Platform Antelope 13334
<!---- SVG 1.1 Document Model

This module describes the groupings of elements that make up common content models for SVG elements.

-->
<!-- module: svg-shape.mod ............................. -->

<!ENTITY % SVG.Shape.extra.class "" >
<!ENTITY % SVG.Shape.class
  "| %SVG.path.qname; | %SVG.rect.qname; | %SVG.circle.qname;
  | %SVG.line.qname; | %SVG.ellipse.qname; | %SVG.polyline.qname;
  | %SVG.polygon.qname; %SVG.Shape.extra.class;"
>

<!-- module: svg-text.mod .............................. -->

<!ENTITY % SVG.Text.extra.class "" >
<!ENTITY % SVG.Text.class
  "| %SVG.text.qname; | %SVG.altGlyphDef.qname; %SVG.Text.extra.class;"
>

<!ENTITY % SVG.TextContent.extra.class "" >
<!ENTITY % SVG.TextContent.class
  "| %SVG.tspan.qname; | %SVG.tref.qname; | %SVG.textPath.qname;
  | %SVG.altGlyph.qname; %SVG.TextContent.extra.class;"
>

<!-- module: svg-marker.mod ............................ -->

<!ENTITY % SVG.Marker.extra.class "" >
<!ENTITY % SVG.Marker.class
  "| %SVG.marker.qname; %SVG.Marker.extra.class;"
>

<!-- module: svg-profile.mod ........................... -->

<!ENTITY % SVG.Profile.extra.class "" >
<!ENTITY % SVG.Profile.class
  "| %SVG.color-profile.qname; %SVG.Profile.extra.class;"
>

<!-- module: svg-gradient.mod .......................... -->

<!ENTITY % SVG.Gradient.extra.class "" >
<!ENTITY % SVG.Gradient.class
  "| %SVG.linearGradient.qname; | %SVG.radialGradient.qname;
  | %SVG.Gradient.extra.class;"
>

<!-- module: svg-pattern.mod ........................... -->
<!!-- ... ............................................... -->
<!-- SVG 1.1 View Module ............................... -->
<!-- file: svg-view.mod

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-view.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "//W3C//ELEMENTS SVG 1.1 View//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-view.mod"

................................................................. -->

<!-- View

view

This module declares markup to provide support for view.

-->

<!-- Qualified Names (Default) ...................... -->

<!ENTITY % SVG.view.qname "view" >

<!-- Attribute Collections (Default) ............... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.View.class .................................... -->

<!ENTITY % SVG.View.extra.class "" >

<!ENTITY % SVG.View.class
"| %SVG.view.qname; %SVG.View.extra.class;"
>

<!-- view: View Element ............................. -->

<!ENTITY % SVG.view.extra.content "" >

<!ENTITY % SVG.view.element "INCLUDE" >
<![%SVG.view.element;]
<!ENTITY % SVG.view.content
"(( %SVG.Description.class; %SVG.view.extra.content; )*" 
>

Open Source Used In DNAC Platform Antelope 13339
<!ELEMENT %SVG.view.qname; %SVG.view.content; >
<!-- end of SVG.view.element -->

<!ENTITY % SVG.view.attlist "INCLUDE" >
<![%SVG.view.attlist;[
<!ATTLIST %SVG.view.qname;
 %SVG.Core.attrib;
 %SVG.External.attrib;
 viewBox %ViewBoxSpec.datatype; #IMPLIED
 preserveAspectRatio %PreserveAspectRatioSpec.datatype; 'xMidYMid meet'
 zoomAndPan ( disable | magnify ) 'magnify'
 viewTarget CDATA #IMPLIED
 >
<!-- end of SVG.view.attlist -->]]>

<!-- end of svg-view.mod -->

Found in path(s):
 * /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
  jar/org/apache/batik/dom/svg/resources/svg-view.mod
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2005 World Wide Web Consortium,
 *
* (Massachusetts Institute of Technology, European Research Consortium for
* Informatics and Mathematics, Keio University). All Rights Reserved. This
* work is distributed under the W3C(r) Software License [1] in the hope that
* it will be useful, but WITHOUT ANY WARRANTY; without even the implied
* warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
* *
* *
* Modifications:
* *
* September 18, 2005
* Translated IDL to Java.
* Placed interface in org.apache.batik.dom.svg12 for the time being.
* Added javadocs.
*/

Found in path(s):
 * /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
  jar/org/apache/batik/dom/svg12/SVGGlobal.java
 * /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
  jar/org/apache/batik/dom/svg12/Global.java
No license file was found, but licenses were detected in source scan.
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-animevents-attrib.mod 1733416 2016-03-03 07:07:13Z gados $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Animation Events Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-animevents-attrib.mod"

...... Animation Events Attribute

onbegin, onend, onrepeat, onload

This module defines the AnimationEvents attribute set.

......

<!ENTITY % SVG.onbegin.attrib "onbegin %Script.datatype; #IMPLIED">

<!ENTITY % SVG.onend.attrib "onend %Script.datatype; #IMPLIED">

<!ENTITY % SVG.onrepeat.attrib "onrepeat %Script.datatype; #IMPLIED">

<!ENTITY % SVG.onload.attrib "onload %Script.datatype; #IMPLIED">

<!ENTITY % SVG.AnimationEvents.extra.attrib "">

<!ENTITY % SVG.AnimationEvents.attrib "%SVG.onbegin.attrib;
%SVG.onend.attrib;
%SVG.onrepeat.attrib;
%SVG.onload.attrib;
%SVG.AnimationEvents.extra.attrib;" >
<!!-- end of svg-animevents-attrib.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
  jar/org/apache/batik/dom/svg/resources/svg-animevents-attrib.mod
No license file was found, but licenses were detected in source scan.

<!-- ................................................................. -->
<!-- SVG 1.1 Basic Document Model Module ...................... -->
<!-- file: svg11-basic-model.mod

This is SVG Basic, a proper subset of SVG.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg11-basic-model.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC ".//W3C//ENTITIES SVG 1.1 Basic Document Model//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-basic-model.mod"

................................................................. -->

<!-- SVG 1.1 Basic Document Model

This module describes the groupings of elements that make up
common content models for SVG elements.
-->

<!-- Content Models (Default) ......................... -->

<!ENTITY % SVG.Marker.class "" >

<!-- module: svg-structure.mod ......................... -->

<!ENTITY % SVG.Description.extra.class "" >
<!ENTITY % SVG.Description.class
"%SVG.desc.qname; | %SVG.title.qname; | %SVG.metadata.qname;
%SVG.Description.extra.class;"
>

<!ENTITY % SVG.Use.extra.class "" >
<!ENTITY % SVG.Use.class
"| %SVG.use.qname; %SVG.Use.extra.class;"
>

<!ENTITY % SVG.Structure.extra.class "" >
<!ENTITY % SVG.Structure.class
"| %SVG.svg.qname; | %SVG.g.qname; | %SVG.defs.qname; | %SVG.symbol.qname;"
%SVG.Use.class; %SVG.Structure.extra.class;

<!-- module: svg-conditional.mod ....................... -->

<!ENTITY % SVG.Conditional.extra.class "" >
<!ENTITY % SVG.Conditional.class
"| %SVG.switch.qname; %SVG.Conditional.extra.class;"
>

<!-- module: svg-image.mod .............................. -->

<!ENTITY % SVG.Image.extra.class "" >
<!ENTITY % SVG.Image.class
"| %SVG.image.qname; %SVG.Image.extra.class;"
>

<!-- module: svg-style.mod ............................. -->

<!ENTITY % SVG.Style.extra.class "" >
<!ENTITY % SVG.Style.class
"| %SVG.style.qname; %SVG.Style.extra.class;"
>

<!-- module: svg-shape.mod ............................. -->

<!ENTITY % SVG.Shape.extra.class "" >
<!ENTITY % SVG.Shape.class
"| %SVG.path.qname; | %SVG.rect.qname; | %SVG.circle.qname;
| %SVG.line.qname; | %SVG.ellipse.qname; | %SVG.polyline.qname;
| %SVG.polygon.qname; %SVG.Shape.extra.class;"
>

<!-- module: svg-text.mod .............................. -->

<!ENTITY % SVG.Text.extra.class "" >
<!ENTITY % SVG.Text.class
"| %SVG.text.qname; %SVG.Text.extra.class;"
>

<!ENTITY % SVG.TextContent.extra.class "" >
<!ENTITY % SVG.TextContent.class
"| %SVG.tspan.qname; | %SVG.tref.qname; | %SVG.textPath.qname;
%SVG.TextContent.extra.class;"
>

<!-- module: svg-profile.mod ........................... -->
Copyright (c) 2000 W3C (MIT, INRIA, Keio), All Rights Reserved.

For SVG 1.0:

Namespace:
http://www.w3.org/2000/svg

Public identifier:
PUBLIC "

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-hyperlink.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Hyperlinking//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-hyperlink.mod"

Hyperlinking

a
This module declares markup to provide support for hyper linking.

<!-- link to this target -->
<!-- Qualified Names (Default) ...................... -->

<!-- Attribute Collections (Default) ............... -->

<!-- SVG.Hyperlink.class .............................. -->

<!-- SVG.Presentation.attrib .......................... -->

<!ENTITY % SVG.Hyperlink.extra.class "" >
<!ENTITY % SVG.Hyperlink.class "| %SVG.a.qname; %SVG.Hyperlink.extra.class;" >
<!ENTITY % SVG.Presentation.attrib "" >
<!ENTITY % SVG.Presentation.extra.attrib "" >
<!ENTITY % SVG.Container.attrib;
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-container-attrib.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-/W3C//ENTITIES SVG 1.1 Container Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-container-attrib.mod"

................................................................. -->

<!-- Container Attribute

   enable-background

   This module defines the Container attribute set. -->

<ENTITY % EnableBackgroundValue.datatype "CDATA" >

<ENTITY % SVG.enable-background.attrib
   "enable-background %EnableBackgroundValue.datatype; #IMPLIED"
>

<ENTITY % SVG.Container.extra.attrib ">

<ENTITY % SVG.Container.attrib
   "%SVG.enable-background.attrib;
   %SVG.Container.extra.attrib;"
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-clip.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS SVG 1.1 Clip//EN"
    SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-clip.mod"

This module declares markup to provide support for clipping.

<!-- 'clip-path' property/attribute value (e.g., 'none', <uri>) -->
<!ENTITY % ClipPathValue.datatype "CDATA" >

<!-- Qualified Names (Default) ......................... -->
<!ENTITY % SVG.clipPath.qname "clipPath" >

<!-- Attribute Collections (Default) ............... -->
<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/resources/svg-clip.mod
No license file was found, but licenses were detected in source scan.

/*

Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/SVGAnimatedPathDataSupport.java
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/SVGItem.java
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/TraitAccess.java
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/SVGOMTransform.java
No license file was found, but licenses were detected in source scan.

<!-- ................................. -->
<!-- SVG 1.1 Qualified Name Module ................................. -->
<!-- file: svg-qname.mod

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-qname.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "/W3C//ENTITIES SVG 1.1 Qualified Name//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-qname.mod"

.......................................................... -->

<!-- Qualified Name
This module is contained in two parts, labeled Section 'A' and 'B':

Section A declares parameter entities to support namespace-qualified names, namespace declarations, and name prefixing for SVG and extensions.

Section B declares parameter entities used to provide namespace-qualified names for all SVG element types:

--> 

<!-- Section A: SVG XML Namespace Framework __________________ -->

<!-- 1. Declare a %SVG.prefixed; conditional section keyword, used to activate namespace prefixing. The default value should inherit '%NS.prefixed;' from the DTD driver, so that unless overridden, the default behaviour follows the overall DTD prefixing scheme. -->

<!ENTITY % NS.prefixed "IGNORE" >
<!ENTITY % SVG.prefixed "%NS.prefixed;" >

<!-- 2. Declare a parameter entity (eg., %SVG.xmlns;) containing the URI reference used to identify the SVG namespace: -->

<!ENTITY % SVG.xmlns "http://www.w3.org/2000/svg" >
<!ENTITY % XLINK.xmlns "http://www.w3.org/1999/xlink" >

<!-- 3. Declare parameter entities (eg., %SVG.prefix;) containing the default namespace prefix string(s) to use when prefixing is enabled. This may be overridden in the DTD driver or the internal subset of an document instance. If no default prefix is desired, this may be declared as an empty string. -->

<!ENTITY % SVG.prefix "" >
<!ENTITY % XLINK.prefix "xlink" >

<!-- 4. Declare parameter entities (eg., %SVG.pfx;) containing the colonized prefix(es) (eg., '%SVG.prefix;:') used when prefixing is active, an empty string when it is not. -->

<![%SVG.prefixed;[
<!ENTITY % SVG.pfx "%SVG.prefix;:" >
]]>
<!ENTITY % SVG.pfx "" >
<!ENTITY % XLINK.pfx "%XLINK.prefix;:" >

<!-- 5. The parameter entity %SVG.xmlns.extra.attrib; may be
redeclared to contain any non-SVG namespace declaration attributes for namespaces embedded in SVG. The default is an empty string.

-->
<!ENTITY % SVG.xmlns.extra.attrib "" >

<!-- Declare a parameter entity XLINK.xmlns.attrib containing the XML Namespace declarations for XLink. -->
<!ENTITY % XLINK.xmlns.attrib "xmlns:%XLINK.prefix; %URI.datatype; #FIXED '%XLINK.xmlns;'' >

<!-- Declare a parameter entity %NS.decl.attrib; containing all XML Namespace declarations used in the DTD, plus the xmlns declaration for SVG, its form dependent on whether prefixing is active. -->
<!ENTITY % NS.decl.attrib "xmlns:%NS.decl.attrib; %XLINK.xmlns.attrib; %SVG.xmlns.extra.attrib;" >

<!-- Declare a parameter entity %NS.decl.attrib; containing all XML namespace declaration attributes used by SVG, including a default xmlns attribute when prefixing is inactive. -->
<!ENTITY % NS.decl.attrib "xmlns %URI.datatype; #FIXED '%NS.decl.attrib;' >

<!-- Section B: SVG Qualified Names :::::::::::::::::::::::::::::: -->

<!-- 6. This section declares parameter entities used to provide

<!-- Section B: SVG Qualified Names ::::::::::::::::::::::::::::::: -->

<%!-- 6. This section declares parameter entities used to provide
namespace-qualified names for all SVG element types.

<!-- module: svg-structure.mod ............................ -->

<!ENTITY % SVG.svg.qname "%SVG.pfx;svg" >
<!ENTITY % SVG.g.qname "%SVG.pfx:g" >
<!ENTITY % SVG.defs.qname "%SVG.pfx;defs" >
<!ENTITY % SVG.desc.qname "%SVG.pfx;desc" >
<!ENTITY % SVG.title.qname "%SVG.pfx:title" >
<!ENTITY % SVG.metadata.qname "%SVG.pfx:metadata" >
<!ENTITY % SVG.symbol.qname "%SVG.pfx;symbol" >
<!ENTITY % SVG.use.qname "%SVG.pfx;use" >

<!-- module: svg-conditional.mod ......................... -->

<!ENTITY % SVG.switch.qname "%SVG.pfx;switch" >

<!-- module: svg-image.mod ............................. -->

<!ENTITY % SVG.image.qname "%SVG.pfx:image" >

<!-- module: svg-style.mod ............................. -->

<!ENTITY % SVG.style.qname "%SVG.pfx;style" >

<!-- module: svg-shape.mod ............................. -->

<!ENTITY % SVG.path.qname "%SVG.pfx:path" >
<!ENTITY % SVG.rect.qname "%SVG.pfx:rect" >
<!ENTITY % SVG.circle.qname "%SVG.pfx:circle" >
<!ENTITY % SVG.line.qname "%SVG.pfx;line" >
<!ENTITY % SVG.ellipse.qname "%SVG.pfx:ellipse" >
<!ENTITY % SVG.polyline.qname "%SVG.pfx;polyline" >
<!ENTITY % SVG.polygon.qname "%SVG.pfx;polygon" >

<!-- module: svg-text.mod ................................ -->

<!ENTITY % SVG.text.qname "%SVG.pfx;text" >
<!ENTITY % SVG.tspan.qname "%SVG.pfx:tspan" >
<!ENTITY % SVG.tref.qname "%SVG.pfx;tref" >
<!ENTITY % SVG.textPath.qname "%SVG.pfx;textPath" >
<!ENTITY % SVG.altGlyph.qname "%SVG.pfx;altGlyph" >
<!ENTITY % SVG.altGlyphDef.qname "%SVG.pfx;altGlyphDef" >
<!ENTITY % SVG.altGlyphItem.qname "%SVG.pfx;altGlyphItem" >
<!ENTITY % SVG.glyphRef.qname "%SVG.pfx;glyphRef" >

<!-- module: svg-marker.mod ............................. -->
<!ENTITY % SVG.feFuncR.qname "%SVG.pfx;feFuncR" >
<!ENTITY % SVG.feFuncG.qname "%SVG.pfx;feFuncG" >
<!ENTITY % SVG.feFuncB.qname "%SVG.pfx;feFuncB" >
<!ENTITY % SVG.feFuncA.qname "%SVG.pfx;feFuncA" >

<!-- module: svg-cursor.mod ......................... -->

<!ENTITY % SVG.cursor.qname "%SVG.pfx;cursor" >

<!-- module: svg-hyperlink.mod ....................... -->

<!ENTITY % SVG.a.qname "%SVG.pfx;a" >

<!-- module: svg-view.mod ......................... -->

<!ENTITY % SVG.view.qname "%SVG.pfx;view" >

<!-- module: svg-script.mod ........................ -->

<!ENTITY % SVG.script.qname "%SVG.pfx;script" >

<!-- module: svg-animation.mod ..................... -->

<!ENTITY % SVG.animateMotion.qname "%SVG.pfx;animateMotion" >
<!ENTITY % SVG.animateColor.qname "%SVG.pfx;animateColor" >
<!ENTITY % SVG.animateTransform.qname "%SVG.pfx;animateTransform" >

<!-- module: svg-font.mod .......................... -->

<!ENTITY % SVG.font.qname "%SVG.pfx;font" >
<!ENTITY % SVG.font-face.qname "%SVG.pfx;font-face" >
<!ENTITY % SVG.glyph.qname "%SVG.pfx;glyph" >
<!ENTITY % SVG.missing-glyph.qname "%SVG.pfx;missing-glyph" >
<!ENTITY % SVG.hkern.qname "%SVG.pfx;hkern" >
<!ENTITY % SVG.vkern.qname "%SVG.pfx;vkern" >
<!ENTITY % SVG.font-face-src.qname "%SVG.pfx;font-face-src" >
<!ENTITY % SVG.font-face-uri.qname "%SVG.pfx;font-face-uri" >
<!ENTITY % SVG.font-face-format.qname "%SVG.pfx;font-face-format" >
<!ENTITY % SVG.font-face-name.qname "%SVG.pfx;font-face-name" >
<!ENTITY % SVG.definition-src.qname "%SVG.pfx;definition-src" >

<!-- module: svg-extensibility.mod ................... -->

<!ENTITY % SVG.foreignObject.qname "%SVG.pfx;foreignObject" >
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-opacity-attrib.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:
PUBLIC "-/W3C//ENTITIES SVG 1.1 Paint Opacity Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-opacity-attrib.mod"

-opacity, fill-opacity, stroke-opacity

This module defines the Opacity attribute set.
-->

<!ENTITY % SVG.opacity.attrib
"opacity %OpacityValue.datatype; #IMPLIED"
>

<!ENTITY % SVG.fill-opacity.attrib
"fill-opacity %OpacityValue.datatype; #IMPLIED"
>

<!ENTITY % SVG.stroke-opacity.attrib
"stroke-opacity %OpacityValue.datatype; #IMPLIED"
>

<!ENTITY % SVG.opacity attrib "" >

<!ENTITY % SVG.opacity attrib
"%SVG.opacity attrib;
%SVG.fill-opacity attrib;
%SVG.stroke-opacity attrib;
%SVG.Opacity.extra attrib;"
This is SVG Basic, a proper subset of SVG.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg11-basic-attribs.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Basic Attribute Collection//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-basic-attribs.mod"

....................................................................... -->

<!-- SVG 1.1 Basic Attribute Collection

This module defines the set of common attributes that can be present
on many SVG elements.
-->
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-basic-clip.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Basic Clip//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-basic-clip.mod"

<!ENTITY % ClipPathValue.datatype "CDATA" >

<!ENTITY % SVG.clipPath.qname "clipPath" >

<!ENTITY % SVG.Clip.class ................. -->
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-docevents-attrib.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Document Events Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-docevents-attrib.mod"

This module defines the DocumentEvents attribute set.

onunload, onabort, onerror, onresize, onscroll, onzoom

onunload %Script.datatype; #IMPLIED

onabort %Script.datatype; #IMPLIED

onerror %Script.datatype; #IMPLIED

onresize %Script.datatype; #IMPLIED

onscroll %Script.datatype; #IMPLIED

<!ENTITY % SVG.onunload.attrib
"onunload %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onabort.attrib
"onabort %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onerror.attrib
"onerror %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onresize.attrib
"onresize %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onscroll.attrib
"onscroll %Script.datatype; #IMPLIED"
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-datatypes.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Datatypes//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-datatypes.mod"

Datatypes

This module declares common data types for properties and attributes.

<ENTITY % Boolean.datatype "( false | true )" >
<!-- 'clip-rule' or 'fill-rule' property/attribute value -->
<!ENTITY % ClipFillRule.datatype "( nonzero | evenodd | inherit )" >

<!-- media type, as per [RFC2045] -->
<!ENTITY % ContentType.datatype "CDATA" >

<!-- a <coordinate> -->
<!ENTITY % Coordinate.datatype "CDATA" >

<!-- a list of <coordinate>s -->
<!ENTITY % Coordinates.datatype "CDATA" >

<!-- a <color> value -->
<!ENTITY % Color.datatype "CDATA" >

<!-- a <integer> -->
<!ENTITY % Integer.datatype "CDATA" >

<!-- a language code, as per [RFC3066] -->
<!ENTITY % LanguageCode.datatype "NMTOKEN" >

<!-- comma-separated list of language codes, as per [RFC3066] -->
<!ENTITY % LanguageCodes.datatype "CDATA" >

<!-- a <length> -->
<!ENTITY % Length.datatype "CDATA" >

<!-- a list of <length>s -->
<!ENTITY % Lengths.datatype "CDATA" >

<!-- a <number> -->
<!ENTITY % Number.datatype "CDATA" >

<!-- a list of <number>s -->
<!ENTITY % Numbers.datatype "CDATA" >

<!-- opacity value (e.g., <number>) -->
<!ENTITY % OpacityValue.datatype "CDATA" >

<!-- a path data specification -->
<!ENTITY % PathData.datatype "CDATA" >

<!-- 'preserveAspectRatio' attribute specification -->
<!ENTITY % PreserveAspectRatioSpec.datatype "CDATA" >

<!-- script expression -->
<!ENTITY % Script.datatype "CDATA" >
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-graphevents-attrib.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-/W3C//ENTITIES SVG 1.1 Graphical Element Events Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-graphevents-attrib.mod"

This module defines the GraphicalEvents attribute set.
<!ENTITY % SVG.onfocusout.attrib
  "onfocusout %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onactivate.attrib
  "onactivate %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onclick.attrib
  "onclick %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onmousedown.attrib
  "onmousedown %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onmouseup.attrib
  "onmouseup %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onmouseover.attrib
  "onmouseover %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onmousemove.attrib
  "onmousemove %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onmouseout.attrib
  "onmouseout %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.onload.attrib
  "onload %Script.datatype; #IMPLIED"
>

<!ENTITY % SVG.GraphicalEvents.extra.attrib "" >

<!ENTITY % SVG.GraphicalEvents.attrib
  "%SVG.onfocusin.attrib;
  %SVG.onfocusout.attrib;
  %SVG.onactivate.attrib;
  %SVG.onclick.attrib;
  %SVG.onmousedown.attrib;
  %SVG.onmouseup.attrib;
  %SVG.onmouseover.attrib;
  %SVG.onmousemoveattrib;
  %SVG.onmouseout.attrib;
  %SVG.onload.attrib"
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-pattern.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS SVG 1.1 Pattern//EN"
    SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-pattern.mod"

....................................................................... -->

<!!-- Pattern

pattern

This module declares markup to provide support for pattern fill.
-->

<!!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.pattern.qname "pattern" >

<!!-- Attribute Collections (Default) ................... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG_CONTAINER.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
No license file was found, but licenses were detected in source scan.

<!-- ................................................................. -->
<!-- SVG 1.1 Attribute Collection Module ...................... -->
<!-- file: svg11-attribs.mod

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg11-attribs.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-/W3C/ENTITIES SVG 1.1 Attribute Collection//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-attribs.mod"

................................................................. -->

<!-- SVG 1.1 Attribute Collection

This module defines the set of common attributes that can be present
on many SVG elements. -->

<!-- module: svg-conditional.mod ....................... -->

<!ENTITY % ExtensionList.datatype "CDATA" >
<!ENTITY % FeatureList.datatype "CDATA" >

<!ENTITY % SVG.Conditional.extra.attrib "" >
<!ENTITY % SVG.Conditional.attrib
  "requiredFeatures %FeatureList.datatype; #IMPLIED
   requiredExtensions %ExtensionList.datatype; #IMPLIED
   systemLanguage %LanguageCodes.datatype; #IMPLIED
   %SVG.Conditional.extra.attrib;"
>

<!-- module: svg-style.mod ................. -->

<!ENTITY % ClassList.datatype "CDATA" >
<!ENTITY % StyleSheet.datatype "CDATA" >

<!ENTITY % SVG.Style.extra.attrib "" >
<!ENTITY % SVG.Style.attrib
  "style %StyleSheet.datatype; #IMPLIED
   class %ClassList.datatype; #IMPLIED
   %SVG.Style.extra.attrib;"
>
font-stretch ( normal | wider | narrower | ultra-condensed |
extra-condensed | condensed | semi-condensed |
semi-expanded | expanded | extra-expanded |
ultra-expanded | inherit ) #IMPLIED
font-style ( normal | italic | oblique | inherit ) #IMPLIED
font-variant ( normal | small-caps | inherit ) #IMPLIED
font-weight ( normal | bold | bolder | lighter | 100 | 200 | 300 | 400 |
500 | 600 | 700 | 800 | 900 | inherit ) #IMPLIED
%SVG.Font.extra.attrib;" >

<!-- module: svg-marker.mod ......................... -->

<!ENTITY % MarkerValue.datatype "CDATA" >

<!ENTITY % SVG.Marker.extra.attrib "" >
<!ENTITY % SVG.Marker.attrib
"marker-start %MarkerValue.datatype; #IMPLIED
marker-mid %MarkerValue.datatype; #IMPLIED
marker-end %MarkerValue.datatype; #IMPLIED
%SVG.Marker.extra.attrib;" >

<!-- module: svg-profile.mod ....................... -->

<!ENTITY % SVG.Profile.extra.attrib "" >
<!ENTITY % SVG.Profile.attrib
"color-profile CDATA #IMPLIED
%SVG.Profile.extra.attrib;" >

<!-- module: svg-gradient.mod ...................... -->

<!ENTITY % NumberOrPercentage.datatype "CDATA" >

<!ENTITY % SVG.Gradient.extra.attrib "" >
<!ENTITY % SVG.Gradient.attrib
"stop-color %SVGColor.datatype; #IMPLIED
stop-opacity %OpacityValue.datatype; #IMPLIED
%SVG.Gradient.extra.attrib;" >

<!-- module: svg-clip.mod ......................... -->

<!ENTITY % ClipPathValue.datatype "CDATA" >

<!ENTITY % SVG.Clip.extra.attrib "" >
<!ENTITY % SVG.Clip.attrib
"clip-path %ClipPathValue.datatype; #IMPLIED
clip-rule %ClipFillRule.datatype; #IMPLIED
%SVG.Clip.extra.attrib;"
>
<!-- module: svg-mask.mod ......................... -->

<!ENTITY % MaskValue.datatype "CDATA" >

<!ENTITY % SVG.Mask.extra.attrib "" >
<!ENTITY % SVG.Mask.attrib
"mask %MaskValue.datatype; #IMPLIED
%SVG.Mask.extra.attrib;"
>
<!-- module: svg-filter.mod ....................... -->

<!ENTITY % FilterValue.datatype "CDATA" >
<!ENTITY % NumberOptionalNumber.datatype "CDATA" >

<!ENTITY % SVG.Filter.extra.attrib "" >
<!ENTITY % SVG.Filter.attrib
"filter %FilterValue.datatype; #IMPLIED
%SVG.Filter.extra.attrib;"
>
<!ENTITY % SVG.FilterColor.extra.attrib "" >
<!ENTITY % SVG.FilterColor.attrib
"color-interpolation-filters ( auto | sRGB | linearRGB | inherit )
#IMPLIED
%SVG.FilterColor.extra.attrib;"
>
<!-- module: svg-cursor.mod ...................... -->

<!ENTITY % CursorValue.datatype "CDATA" >

<!ENTITY % SVG.Cursor.extra.attrib "" >
<!ENTITY % SVG.Cursor.attrib
"cursor %CursorValue.datatype; #IMPLIED
%SVG.Cursor.extra.attrib;"
>
<!-- end of svg11-attribs.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1.jar/org/apache/batik/dom/svg/resources/svg11-attribs.mod
No license file was found, but licenses were detected in source scan.

<!-- ............................................................. -->
<!-- SVG 1.1 Filter Module ....................................... -->
<!-- file: svg-filter.mod

This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-filter.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Filter//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-filter.mod"

....................................................................... -->

<!-- Filter


This module declares markup to provide support for filter effect.
-->

<!-- 'filter' property/attribute value (e.g., 'none', <uri>) -->
<!ENTITY % FilterValue.datatype "CDATA" >

<!-- list of <number>s, but at least one and at most two -->
<!ENTITY % NumberOptionalNumber.datatype "CDATA" >

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.filter.qname "filter" >
<!ENTITY % SVG.feBlend.qname "feBlend" >
<!ENTITY % SVG.feColorMatrix.qname "feColorMatrix" >
<!ENTITY % SVG.feComponentTransfer.qname "feComponentTransfer" >
<!ENTITY % SVG.feComposite.qname "feComposite" >
<!ENTITY % SVG.feConvolveMatrix.qname "feConvolveMatrix" >
<!ENTITY % SVG.feDiffuseLighting.qname "feDiffuseLighting" >
<!ENTITY % SVG.feDisplacementMap.qname "feDisplacementMap" >
<!ENTITY % SVG.feFlood.qname "feFlood" >
<!ENTITY % SVG.feGaussianBlur.qname "feGaussianBlur" >
<!ENTITY % SVG.feImage.qname "feImage" >
<!ENTITY % SVG.feMerge.qname "feMerge" >
<!ENTITY % SVG.feMergeNode.qname "feMergeNode" >
<!ENTITY % SVG.feMorphology.qname "feMorphology" >
<!ENTITY % SVG.feOffset.qname "feOffset" >
<!ENTITY % SVG.feSpecularLighting.qname "feSpecularLighting" >
<!ENTITY % SVG.feTile.qname "feTile" >
<!ENTITY % SVG.feTurbulence.qname "feTurbulence" >
<!ENTITY % SVG.feDistantLight.qname "feDistantLight" >
<!ENTITY % SVG.fePointLight.qname "fePointLight" >
<!ENTITY % SVG.feSpotLight.qname "feSpotLight" >
<!ENTITY % SVG.feFuncR.qname "feFuncR" >
<!ENTITY % SVG.feFuncG.qname "feFuncG" >
<!ENTITY % SVG.feFuncB.qname "feFuncB" >
<!ENTITY % SVG.feFuncA.qname "feFuncA" >
Open Source Used In DNAC Platform Antelope  13385
<ATTLIST %SVG.feMerge.qname;
  %SVG.Core.attrib;
  %SVG.FilterColor.attrib;
  %SVG.FilterPrimitive.attrib;
>
<!-- end of SVG.feMerge.attlist -->]]>

<!-- feMergeNode: Filter Effect Merge Node Element .... -->

ENTITY % SVG.feMergeNode.extra.content "" >

ENTITY % SVG.feMergeNode.element "INCLUDE" >
<![%SVG.feMergeNode.element;[ ENTITY % SVG.feMergeNode.content
"(%SVG.animate.qname; | %SVG.set.qname; %SVG.feMergeNode.extra.content; )*"
>
ELEMENT %SVG.feMergeNode.qname; %SVG.feMergeNode.content; >
<!-- end of SVG.feMergeNode.element -->]]>

ENTITY % SVG.feMergeNode.attlist "INCLUDE" >
<![%SVG.feMergeNode.attlist;[ ATTLIST %SVG.feMergeNode.qname;
  %SVG.Core.attrib;
  in CDATA #IMPLIED
>
<!-- end of SVG.feMergeNode.attlist -->]]>

<!-- feMorphology: Filter Effect Morphology Element .... -->

ENTITY % SVG.feMorphology.extra.content "" >

ENTITY % SVG.feMorphology.element "INCLUDE" >
<![%SVG.feMorphology.element;[ ENTITY % SVG.feMorphology.content
"(%SVG.animate.qname; | %SVG.set.qname;
  %SVG.feMorphology.extra.content; )*"
>
ELEMENT %SVG.feMorphology.qname; %SVG.feMorphology.content; >
<!-- end of SVG.feMorphology.element -->]]>

ENTITY % SVG.feMorphology.attlist "INCLUDE" >
<![%SVG.feMorphology.attlist;[ ATTLIST %SVG.feMorphology.qname;
  %SVG.Core.attrib;
  %SVG.FilterColor.attrib;
  %SVG.FilterPrimitiveWithIn.attrib;
  operator ( erode | dilate ) 'erode'
  radius %NumberOptionalNumber.datatype; #IMPLIED
  >
baseFrequency %NumberOptionalNumber.datatype; #IMPLIED
numOctaves %Integer.datatype; #IMPLIED
seed %Number.datatype; #IMPLIED
stitchTiles ( stitch | noStitch ) 'noStitch'
type ( fractalNoise | turbulence ) 'turbulence'
>
</!-- end of SVG.feTurbulence.attlist -->]]>

</!-- feDistantLight: Filter Effect Distant Light Element -->

<!ENTITY % SVG.feDistantLight.extra.content "" >

<!ENTITY % SVG.feDistantLight.element "INCLUDE" >
</[%SVG.feDistantLight.element;[
<!ENTITY % SVG.feDistantLight.content
"( %SVG.animate.qname; | %SVG.set.qname;
 %SVG.feDistantLight.extra.content; )*"
>
</!-- end of SVG.feDistantLight.qname; %SVG.feDistantLight.content; >
</!-- end of SVG.feDistantLight.element --]]>

<!ENTITY % SVG.feDistantLight.attlist "INCLUDE" >
</[%SVG.feDistantLight.attlist;[
<!ATTLIST %SVG.feDistantLight.qname; %SVG.Core.attrib;
 azimuth %Number.datatype; #IMPLIED
elevation %Number.datatype; #IMPLIED
>
</!-- end of SVG.feDistantLight.attlist -->]]>

</!-- fePointLight: Filter Effect Point Light Element ... -->

<!ENTITY % SVG.fePointLight.extra.content "" >

<!ENTITY % SVG.fePointLight.element "INCLUDE" >
</[%SVG.fePointLight.element;[
<!ENTITY % SVG.fePointLight.content
"( %SVG.animate.qname; | %SVG.set.qname;
 %SVG.fePointLight.extra.content; )*"
>
</!-- end of SVG.fePointLight.qname; %SVG.fePointLight.content; >
</!-- end of SVG.fePointLight.element --]]>

<!ENTITY % SVG.fePointLight.attlist "INCLUDE" >
</[%SVG.fePointLight.attlist;[
<!ATTLIST %SVG.fePointLight.qname; %SVG.Core.attrib;
x %Number.datatype; #IMPLIED
y %Number.datatype; #IMPLIED
z %Number.datatype; #IMPLIED
>
<!-- end of SVG.fePointLight.attlist -->]]>

<!-- feSpotLight: Filter Effect Spot Light Element ..... -->

<!--ENTITY % SVG.feSpotLight.extra.content "" >

<!--ENTITY % SVG.feSpotLight.element "INCLUDE" >
<![SVG.feSpotLight.element;[
<!--ENTITY % SVG.feSpotLight.content
"( %SVG.animate.qname; | %SVG.set.qname; %SVG.feSpotLight.extra.content; )*"
>
<!--ELEMENT %SVG.feSpotLight.qname; %SVG.feSpotLight.content; >
<!-- end of SVG.feSpotLight.element -->]]>

<!--ENTITY % SVG.feSpotLight.attlist "INCLUDE" >
<![SVG.feSpotLight.attlist;[
<!--ATTLIST %SVG.feSpotLight.qname;
%SVG.Core.attrib;
x %Number.datatype; #IMPLIED
y %Number.datatype; #IMPLIED
z %Number.datatype; #IMPLIED
pointsAtX %Number.datatype; #IMPLIED
pointsAtY %Number.datatype; #IMPLIED
pointsAtZ %Number.datatype; #IMPLIED
specularExponent %Number.datatype; #IMPLIED
limitingConeAngle %Number.datatype; #IMPLIED
>
<!-- end of SVG.feSpotLight.attlist -->]]>

<!-- feFuncR: Filter Effect Function Red Element ....... -->

<!--ENTITY % SVG.feFuncR.extra.content "" >

<!--ENTITY % SVG.feFuncR.element "INCLUDE" >
<![SVG.feFuncR.element;[
<!--ENTITY % SVG.feFuncR.content
"( %SVG.animate.qname; | %SVG.set.qname; %SVG.feFuncR.extra.content; )*"
>
<!--ELEMENT %SVG.feFuncR.qname; %SVG.feFuncR.content; >
<!-- end of SVG.feFuncR.element -->]]>

<!--ENTITY % SVG.feFuncR.attlist "INCLUDE" >
<![SVG.feFuncR.attlist;[
<!--ATTLIST %SVG.feFuncR.qname;
%SVG.Core.attrib;
The Scalable Vector Graphics (SVG)
(Massachusetts Institute of Technology, Institut National de
Recherche en Informatique et en Automatique, Keio University).
All Rights Reserved.

Permission to use, copy, modify and distribute the SVG DTD and its
accompanying documentation for any purpose and without fee is hereby
granted in perpetuity, provided that the above copyright notice and
this paragraph appear in all copies. The copyright holders make no
representation about the suitability of the DTD for any purpose.

It is provided "as is" without expressed or implied warranty.

Author: Jun Fujisawa <fujisawa.jun@canon.co.jp>
Revision: $Id: svg12

dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg12

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-basic-filter.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "/W3C//ELEMENTS SVG 1.1 Basic Filter//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-basic-filter.mod"

................................................................. -->

<!- Basic Filter

  filter, feBlend, feColorMatrix, feComponentTransfer, feComposite,
  feFlood, feGaussianBlur, feImage, feMerge, feMergeNode, feOffset,
  feTile, feFuncR, feFuncG, feFuncB, feFuncA

This module declares markup to provide support for filter effect.
<!-- 'filter' property/attribute value (e.g., 'none', <uri>) -->
<!ENTITY % FilterValue.datatype "CDATA">

<!-- list of <number>s, but at least one and at most two -->
<!ENTITY % NumberOptionalNumber.datatype "CDATA">

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.filter.qname "filter">
<!ENTITY % SVG.feBlend.qname "feBlend">
<!ENTITY % SVG.feColorMatrix.qname "feColorMatrix">
<!ENTITY % SVG.feComponentTransfer.qname "feComponentTransfer">
<!ENTITY % SVG.feComposite.qname "feComposite">
<!ENTITY % SVG.feFlood.qname "feFlood">
<!ENTITY % SVG.feGaussianBlur.qname "feGaussianBlur">
<!ENTITY % SVG.feImage.qname "feImage">
<!ENTITY % SVG.feMerge.qname "feMerge">
<!ENTITY % SVG.feMergeNode.qname "feMergeNode">
<!ENTITY % SVG.feOffset.qname "feOffset">
<!ENTITY % SVG.feTile.qname "feTile">
<!ENTITY % SVG.feFuncR.qname "feFuncR">
<!ENTITY % SVG.feFuncG.qname "feFuncG">
<!ENTITY % SVG.feFuncB.qname "feFuncB">
<!ENTITY % SVG.feFuncA.qname "feFuncA">

<!-- Attribute Collections (Default) ................... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Container.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Marker.attrib "" >
<!ENTITY % SVG.Profile.attrib "" >
<!ENTITY % SVG.Gradient.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.XLink.attrib "" >
<!ENTITY % SVG.XLinkEmbed.attrib "" >
<!ENTITY % SVG.filter.element "INCLUDE" >
<![%SVG.filter.element;]
<!ENTITY % SVG.filter.content
  "(( %SVG.Description.class; )* , ( %SVG.animate.qname; | %SVG.set.qname;
   %SVG.FilterPrimitive.class; %SVG.filter.extra.content; )* )")
>
<!ELEMENT %SVG.filter.qname; %SVG.filter.content; >
<!-- end of SVG.filter.element -->]]>

<!ENTITY % SVG.filter.attlist "INCLUDE" >
<![%SVG.filter.attlist;]
<!ATTLIST %SVG.filter.qname;
  %SVG.Core.attrib;
  %SVG.Style.attrib;
  %SVG.Presentation.attrib;
  %SVG.XLink.attrib;
  %SVG.External.attrib;
  x %Coordinate.datatype; #IMPLIED
  y %Coordinate.datatype; #IMPLIED
  width %Length.datatype; #IMPLIED
  height %Length.datatype; #IMPLIED
  filterRes %NumberOptionalNumber.datatype; #IMPLIED
  filterUnits ( userSpaceOnUse | objectBoundingBox ) #IMPLIED
  primitiveUnits ( userSpaceOnUse | objectBoundingBox ) #IMPLIED
>
<!-- end of SVG.filter.attlist -->]]>

<!-- feBlend: Filter Effect Blend Element .............. -->
<!ENTITY % SVG.feBlend.extra.content "" >

<!ENTITY % SVG.feBlend.element "INCLUDE" >
<![%SVG.feBlend.element;]
<!ENTITY % SVG.feBlend.content
  "( %SVG.animate.qname; | %SVG.set.qname; %SVG.feBlend.extra.content; )*" 
>
<!ELEMENT %SVG.feBlend.qname; %SVG.feBlend.content; >
<!-- end of SVG.feBlend.element -->]]>

<!ENTITY % SVG.feBlend.attlist "INCLUDE" >
<![%SVG.feBlend.attlist;]
<!ATTLIST %SVG.feBlend.qname;
  %SVG.Core.attrib;
  %SVG.FilterColor.attrib;
  %SVG.FilterPrimitiveWithIn.attrib;
  in2 CDATA #REQUIRED
  mode ( normal | multiply | screen | darken | lighten ) 'normal'
>
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-paint-attrib.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Paint Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-paint-attrib.mod"

................................................................. -->

<!- file: svg-paint-attrib.mod
<!-- Paint Attribute

   fill, fill-rule, stroke, stroke-dasharray, stroke-dashoffset,
   stroke-linecap, stroke-linejoin, stroke-miterlimit, stroke-width, color,
   color-interpolation, color-rendering

This module defines the Paint and Color attribute sets.
-->

<!-- a 'fill' or 'stroke' property/attribute value: <paint> -->
<!ENTITY % Paint.datatype "CDATA" >

<!-- 'stroke-dasharray' property/attribute value (e.g., 'none', list of <number>s) -->
<!ENTITY % StrokeDashArrayValue.datatype "CDATA" >

<!-- 'stroke-dashoffset' property/attribute value (e.g., 'none', <length>) -->
<!ENTITY % StrokeDashOffsetValue.datatype "CDATA" >

<!-- 'stroke-miterlimit' property/attribute value (e.g., <number>) -->
<!ENTITY % StrokeMiterLimitValue.datatype "CDATA" >

<!-- 'stroke-width' property/attribute value (e.g., <length>) -->
<!ENTITY % StrokeWidthValue.datatype "CDATA" >

<!ENTITY % SVG.fill.attrib
   "fill %Paint.datatype; #IMPLIED"
>

<!ENTITY % SVG.fill-rule.attrib
   "fill-rule %ClipFillRule.datatype; #IMPLIED"
>

<!ENTITY % SVG.stroke.attrib
   "stroke %Paint.datatype; #IMPLIED"
>

<!ENTITY % SVG.stroke-dasharray.attrib
   "stroke-dasharray %StrokeDashArrayValue.datatype; #IMPLIED"
>

<!ENTITY % SVG.stroke-dashoffset.attrib
   "stroke-dashoffset %StrokeDashOffsetValue.datatype; #IMPLIED"
>

<!ENTITY % SVG.stroke-linecap.attrib
   "stroke-linecap ( butt | round | square | inherit ) #IMPLIED"
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-extresources-attrib.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 External Resources Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-extresources-attrib.mod"

................................................................. -->

<!-- External Resources Attribute -->

externalResourcesRequired

This module defines the External attribute set.

-->

<!ENTITY % SVG.externalResourcesRequired.attrib
"externalResourcesRequired %Boolean.datatype; #IMPLIED"
>

<!ENTITY % SVG.External.extra.attrib "" >

<!ENTITY % SVG.External.attrib
"%SVG.externalResourcesRequired.attrib;
%SVG.External.extra.attrib;"
>

<!-- end of svg-extresources-attrib.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1.jar/org/apache/batik/dom/svg/resources/svg-extresources-attrib.mod
No license file was found, but licenses were detected in source scan.

SVG 1.1 Basic DTD

This is SVG Basic, a proper subset of SVG.

The Scalable Vector Graphics (SVG)
(Massachusetts Institute of Technology, Institut National de
Recherche en Informatique et en Automatique, Keio University).
All Rights Reserved.

Permission to use, copy, modify and distribute the SVG DTD and its
accompanying documentation for any purpose and without fee is hereby
granted in perpetuity, provided that the above copyright notice and
this paragraph appear in all copies. The copyright holders make no
representation about the suitability of the DTD for any purpose.

It is provided "as is" without expressed or implied warranty.

Author: Jun Fujisawa <fujisawa.jun@canon.co.jp>
Revision: $Id: svg11

dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg11

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
jar/org/apache/batik/dom/svg/resources/svg11-basic-flat.dtd

No license file was found, but licenses were detected in source scan.

<!-- ....................................................................... -->
<!-- SVG 1.1 Text Module ................................................... -->
<!-- file: svg-text.mod

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-text.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS SVG 1.1 Text//EN"
    SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-text.mod"

................................................................. -->

<!-- Text
This module declares markup to provide support for alternate glyph.

<!-- 'baseline-shift' property/attribute value (e.g., 'baseline', 'sub', etc.) -->
<!ENTITY % BaselineShiftValue.datatype "CDATA" >

<!-- 'font-family' property/attribute value (i.e., list of fonts) -->
<!ENTITY % FontFamilyValue.datatype "CDATA" >

<!-- 'font-size' property/attribute value -->
<!ENTITY % FontSizeValue.datatype "CDATA" >

<!-- 'font-size-adjust' property/attribute value -->
<!ENTITY % FontSizeAdjustValue.datatype "CDATA" >

<!-- 'glyph-orientation-horizontal' property/attribute value (e.g., <angle>) -->
<!ENTITY % GlyphOrientationHorizontalValue.datatype "CDATA" >

<!-- 'glyph-orientation-vertical' property/attribute value (e.g., 'auto', <angle>) -->
<!ENTITY % GlyphOrientationVerticalValue.datatype "CDATA" >

<!-- 'kerning' property/attribute value (e.g., 'auto', <length>) -->
<!ENTITY % KerningValue.datatype "CDATA" >

<!-- 'letter-spacing' or 'word-spacing' property/attribute value (e.g., 'normal', <length>) -->
<!ENTITY % SpacingValue.datatype "CDATA" >

<!-- 'text-decoration' property/attribute value (e.g., 'none', 'underline') -->
<!ENTITY % TextDecorationValue.datatype "CDATA" >

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.text.qname "text" >
<!ENTITY % SVG.tspan.qname "tspan" >
<!ENTITY % SVG.tref.qname "tref" >
<!ENTITY % SVG.textPath.qname "textPath" >
<!ENTITY % SVG.altGlyph.qname "altGlyph" >
<!ENTITY % SVG.altGlyphDef.qname "altGlyphDef" >
<!ENTITY % SVG.altGlyphItem.qname "altGlyphItem" >
<!ENTITY % SVG.glyphRef.qname "glyphRef" >

<!-- Attribute Collections (Default) ..................... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<ENTITY % SVG.Style.attrib "" >
<ENTITY % SVG.Paint.attrib "" >
<ENTITY % SVG.Color.attrib "" >
<ENTITY % SVG.Opacity.attrib "" >
<ENTITY % SVG.Graphics.attrib "" >
<ENTITY % SVG.Clip.attrib "" >
<ENTITY % SVG.Mask.attrib "" >
<ENTITY % SVG.Filter.attrib "" >
<ENTITY % SVG.GraphicalEvents.attrib "" >
<ENTITY % SVG.Cursor.attrib "" >
<ENTITY % SVG.XLink.attrib "" >
<ENTITY % SVG.XLinkRequired.attrib "" >
<ENTITY % SVG.External.attrib "" >

<!-- SVG.Text.class .................................... -->

<!ENTITY % SVG.Text.class | %SVG.tspan.qname; | %SVG.tref.qname; | %SVG.textPath.qname; | %SVG.altGlyph.qname; %SVG.Text.extra.class;>

<!-- SVG.TextContent.class ............................. -->

<!ENTITY % SVG.TextContent.class | %SVG.tspan.qname; | %SVG.tref.qname; | %SVG.textPath.qname; | %SVG.altGlyph.qname; %SVG.TextContent.extra.class;>

<!-- SVG.Text.attrib ................................... -->

<!ENTITY % SVG.Text.attrib "writing-mode ( lr-tb | rl-tb | tb-rl | lr | rl | tb | inherit ) #IMPLIED %SVG.Text.extra.attrib;" >

<!-- SVG.TextContent.attrib ............................ -->

<!ENTITY % SVG.TextContent.attrib "alignment-baseline ( auto | baseline | before-edge | text-before-edge | middle | central | after-edge | text-after-edge | ideographic | alphabetic | hanging | mathematical |
inherit ) #IMPLIED
baseline-shift %BaselineShiftValue.datatype; #IMPLIED
direction ( ltr | rtl | inherit ) #IMPLIED
dominant-baseline ( auto | use-script | no-change | reset-size |
    ideographic | alphabetic | hanging | mathematical |
    central | middle | text-after-edge | text-before-edge |
    inherit ) #IMPLIED
glyph-orientation-horizontal %GlyphOrientationHorizontalValue.datatype; #IMPLIED
glyph-orientation-vertical %GlyphOrientationVerticalValue.datatype; #IMPLIED
kerning %KerningValue.datatype; #IMPLIED
letter-spacing %SpacingValue.datatype; #IMPLIED
text-anchor ( start | middle | end | inherit ) #IMPLIED
text-decoration %TextDecorationValue.datatype; #IMPLIED
unicode-bidi ( normal | embed | bidi-override | inherit ) #IMPLIED
word-spacing %SpacingValue.datatype; #IMPLIED
%SVG.TextContent.extra.attrib;"
>
<!-- SVG.Font.attrib ................................. -->

<!ENTITY % SVG.Font.extra.attrib "" >

<!ENTITY % SVG.Font.attrib
"font-family %FontFamilyValue.datatype; #IMPLIED
font-size %FontSizeValue.datatype; #IMPLIED
font-size-adjust %FontSizeAdjustValue.datatype; #IMPLIED
font-stretch ( normal | wider | narrower | ultra-condensed |
    extra-condensed | condensed | semi-condensed |
    semi-expanded | expanded | extra-expanded |
    ultra-expanded | inherit ) #IMPLIED
font-style ( normal | italic | oblique | inherit ) #IMPLIED
font-variant ( normal | small-caps | inherit ) #IMPLIED
font-weight ( normal | bold | bolder | lighter | 100 | 200 | 300 | 400 |
    500 | 600 | 700 | 800 | 900 | inherit ) #IMPLIED
%SVG.Font.extra.attrib;"
>
<!-- text: Text Element .............................. -->

<!ENTITY % SVG.text.extra.content "" >

<!ENTITY % SVG.text.element "INCLUDE" >
<![%SVG.text.element;]
<!ENTITY % SVG.text.content
"(#PCDATA | %SVG.Description.class; | %SVG.Animation.class;
    %SVG.TextContent.class; %SVG.Hyperlink.class;
%SVG.External.attrib;
x %Coordinates.datatype; #IMPLIED
y %Coordinates.datatype; #IMPLIED
dx %Lengths.datatype; #IMPLIED
dy %Lengths.datatype; #IMPLIED
glyphRef CDATA #IMPLIED
format CDATA #IMPLIED
rotate %Numbers.datatype; #IMPLIED
>
<!-- end of SVG.altGlyph.attlist -->]]>

<!-- altGlyphDef: Alternate Glyph Definition Element ... -->

<!ENTITY % SVG.altGlyphDef.extra.content "" >

<!ENTITY % SVG.altGlyphDef.element "INCLUDE" >
<![%SVG.altGlyphDef.element;]
<!ENTITY % SVG.altGlyphDef.content
"(( %SVG.glyphRef.qname;+ | %SVG.altGlyphItem.qname;+ )
 %SVG.altGlyphDef.extra.content; )"
>
<!ELEMENT %SVG.altGlyphDef.qname; %SVG.altGlyphDef.content; >
<!!-- end of SVG.altGlyphDef.element -->]]>

<!ENTITY % SVG.altGlyphDef.attlist "INCLUDE" >
<![%SVG.altGlyphDef.attlist;]
<!ATTLIST %SVG.altGlyphDef.qname;
 %SVG.Core.attrib;
>
<!!-- end of SVG.altGlyphDef.attlist -->]]>

<!-- altGlyphItem: Alternate Glyph Item Element ....... -->

<!ENTITY % SVG.altGlyphItem.extra.content "" >

<!ENTITY % SVG.altGlyphItem.element "INCLUDE" >
<![%SVG.altGlyphItem.element;]
<!ENTITY % SVG.altGlyphItem.content
"(( %SVG.glyphRef.qname;+ %SVG.altGlyphItem.extra.content; )"
>
<!ELEMENT %SVG.altGlyphItem.qname; %SVG.altGlyphItem.content; >
<!!-- end of SVG.altGlyphItem.element -->]]>

<!ENTITY % SVG.altGlyphItem.attlist "INCLUDE" >
<![%SVG.altGlyphItem.attlist;]
<!ATTLIST %SVG.altGlyphItem.qname;
 %SVG.Core.attrib;
>
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-style.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Style//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-style.mod"

......................................................... -->

<!-- Style

style

This module declares markup to provide support for stylesheet.
-->

<!-- list of classes -->
<!ENTITY % ClassList.datatype "CDATA" >

<!-- comma-separated list of media descriptors. -->
<!ENTITY % MediaDesc.datatype "CDATA" >

<!-- style sheet data -->
<!ENTITY % StyleSheet.datatype "CDATA" >

<!-- Qualified Names (Default) ..................... -->
<!ENTITY % SVG.style.qname "style" >

<!-- Attribute Collections (Default) .............. -->
<!ENTITY % SVG.Core.attrib "" >

<!-- SVG.Style.class .................................. -->
<!ENTITY % SVG.Style.extra.class "" >

<!ENTITY % SVG.Style.class
"| %SVG.style.qname; %SVG.Style.extra.class;"
>

<!-- SVG.Style.attrib ................................

<!ENTITY % SVG.Style.extra.attrib "" >

<!ENTITY % SVG.Style.attrib
"style %StyleSheet.datatype; #IMPLIED
class %ClassList.datatype; #IMPLIED
%SVG.Style.extra.attrib;"
>
<!-- style: Style Element ......................... -->

<!ENTITY % SVG.style.extra.content "" >

<!ENTITY % SVG.style.element "INCLUDE" >
<![%SVG.style.element;[
<!ENTITY % SVG.style.content
"(#PCDATA %SVG.style.extra.content;)*"
>
<!ELEMENT %SVG.style.qname; %SVG.style.content; >
<!!-- end of SVG.style.element --]>

<!ENTITY % SVG.style.attlist "INCLUDE" >
<![%SVG.style.attlist;[
<!ATTLIST %SVG.style.qname;
   xml:space ( preserve ) #FIXED 'preserve'
   %SVG.Core.attrib;
   type %ContentType.datatype; #REQUIRED
   media %MediaDesc.datatype; #IMPLIED
   title %Text.datatype; #IMPLIED
>
<!!-- end of SVG.style.attlist --]>

<!-- end of svg-style.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
jar/org/apache/batik/dom/svg/resources/svg-style.mod
No license file was found, but licenses were detected in source scan.

<!-- ............................................................... -->

<!-- SVG 1.1 Basic Font Module ........................................ -->
<!-- file: svg-basic-font.mod -->

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-basic-font.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-/W3C//ELEMENTS SVG 1.1 Basic Font//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-basic-font.mod"
-- Basic Font
/font, font-face, glyph, missing-glyph, hkern, vkern, font-face-src,
font-face-uri, font-face-name

This module declares markup to provide support for template.
-->

-- Qualified Names (Default) ...................... -->

<!ENTITY % SVG.font.qname "font">
<!ENTITY % SVG.font-face.qname "font-face">
<!ENTITY % SVG.glyph.qname "glyph">
<!ENTITY % SVG.missing-glyph.qname "missing-glyph">
<!ENTITY % SVG.hkern.qname "hkern">
<!ENTITY % SVG.vkern.qname "vkern">
<!ENTITY % SVG.font-face-src.qname "font-face-src">
<!ENTITY % SVG.font-face-uri.qname "font-face-uri">
<!ENTITY % SVG.font-face-name.qname "font-face-name">

-- Attribute Collections (Default) ............. -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Container.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Marker.attrib "" >
<!ENTITY % SVG.Profile.attrib "" >
<!ENTITY % SVG.Gradient.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.FilterColor.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.XLinkRequired.attrib "" >
<!ENTITY % SVG.External.attrib "" >

-- SVG.Font.class .................................. -->
Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/resources/svg-basic-font.mod

No license file was found, but licenses were detected in source scan.

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-conditional.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C/ELEMENTS SVG 1.1 Conditional Processing//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-conditional.mod"

................................................................. -->
<!-- Conditional Processing

switch

This module declares markup to provide support for conditional processing.
-->

<!-- extension list specification -->
<!ENTITY % ExtensionList.datatype "CDATA" >

<!-- feature list specification -->
<!ENTITY % FeatureList.datatype "CDATA" >

<!-- Qualified Names (Default) ......................... -->
<!ENTITY % SVG.switch.qname "switch" >

<!-- Attribute Collections (Default) .................... -->
<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Container.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Marker.attrib "" >
<!ENTITY % SVG.Profile.attrib "" >
<!ENTITY % SVG.Gradient.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.FilterColor.attrib "" >
<!ENTITY % SVG.GraphicalEvents.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Conditional.class ............................. -->
<!ENTITY % SVG.Conditional.extra.class "" >
<!ENTITY % SVG.Conditional.class "| %SVG.switch.qname; %SVG.Conditional.extra.class;" >
<!-- SVG.Conditional.attrib ................................ -->

<!ENTITY % SVG.Conditional.extra.attrib "" >

<!ENTITY % SVG.Conditional.attrib
"requiredFeatures %FeatureList.datatype; #IMPLIED
requiredExtensions %ExtensionList.datatype; #IMPLIED
systemLanguage %LanguageCodes.datatype; #IMPLIED
%SVG.Conditional.extra.attrib;"
>

<!-- SVG.Presentation.attrib ........................... -->

<!ENTITY % SVG.Presentation.extra.attrib "" >

<!ENTITY % SVG.Presentation.attrib
"%SVG.Container.attrib;
%SVG.Viewport.attrib;
%SVG.Text.attrib;
%SVG.TextContent.attrib;
%SVG.Font.attrib;
%SVG.Paint.attrib;
%SVG.Color.attrib;
%SVG.Opacity.attrib;
%SVG/Graphics.attrib;
%SVG.Marker.attrib;
%SVG.Profile.attrib;
%SVG.Gradient.attrib;
%SVG.Clip.attrib;
%SVG.Mask.attrib;
%SVG.Filter.attrib;
%SVG.FilterColor.attrib;
%SVG.Cursor.attrib;
flood-color %SVGColor.datatype; #IMPLIED
flood-opacity %OpacityValue.datatype; #IMPLIED
lighting-color %SVGColor.datatype; #IMPLIED
%SVG.Presentation.extra.attrib;"
>

<!-- switch: Switch Element ............................ -->

<!ENTITY % SVG.switch.extra.content "" >

<!ENTITY % SVG.switch.element "INCLUDE" >
<![%SVG.switch.element;]
<!ENTITY % SVG.switch.content
"((%SVG.Description.class;)*, (%SVG.svg.qname; | %SVG.g.qname;
Open Source Used In DNAC Platform Antelope 13433

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
jar/org/apache/batik/dom/svg/resources/svg-conditional.mod

No license file was found, but licenses were detected in source scan.

SVG 1.1 Basic DTD

This is SVG Basic, a proper subset of SVG.

The Scalable Vector Graphics (SVG)
(Massachusetts Institute of Technology, Institut National de
Recherche en Informatique et en Automatique, Keio University).
All Rights Reserved.

Permission to use, copy, modify and distribute the SVG DTD and its
accompanying documentation for any purpose and without fee is hereby
granted in perpetuity, provided that the above copyright notice and
this paragraph appear in all copies. The copyright holders make no
representation about the suitability of the DTD for any purpose.

It is provided "as is" without expressed or implied warranty.

Author: Jun Fujisawa <fujisawa.jun@canon.co.jp>
Revision: $Id: svg11
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-graphics-attrib.mod 1733416 2016-03-03 07:07:13Z gadams$

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Graphics Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-graphics-attrib.mod"

<!-- Graphics Attribute
    display, image-rendering, pointer-events, shape-rendering,
    text-rendering, visibility

    This module defines the Graphics attribute set. -->

<!ENTITY % SVG.display.attrib "display ( inline | block | list-item | run-in | compact | marker |
    table | inline-table | table-row-group | table-header-group |
    table-footer-group | table-row | table-column-group |
    table-column | table-cell | table-caption | none | inherit )
#IMPLIED"
>

<!ENTITY % SVG.image-rendering.attrib "image-rendering ( auto | optimizeSpeed | optimizeQuality | inherit )
#IMPLIED"
>

<!ENTITY % SVG.pointer-events.attrib "pointer-events ( visiblePainted | visibleFill | visibleStroke | visible |
    painted | fill | stroke | all | none | inherit )
#IMPLIED"
>
<!ENTITY % SVG.shape-rendering.attrib
    "shape-rendering ( auto | optimizeSpeed | crispEdges | geometricPrecision |
      inherit ) #IMPLIED”>

<!ENTITY % SVG.text-rendering.attrib
    "text-rendering ( auto | optimizeSpeed | optimizeLegibility |
      geometricPrecision | inherit ) #IMPLIED”>

<!ENTITY % SVG.visibility.attrib
    "visibility ( visible | hidden | inherit ) #IMPLIED”>

<!ENTITY % SVG.Graphics.extra.attrib "" >

<!ENTITY % SVG.Graphics.attrib
    "%SVG.display.attrib;
    %SVG.image-rendering.attrib;
    %SVG.pointer-events.attrib;
    %SVG.shape-rendering.attrib;
    %SVG.text-rendering.attrib;
    %SVG.visibility.attrib;
    %SVG.Graphics.extra.attrib;”
>

<!-- end of svg-graphics-attrib.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
  jar/org/apache/batik/dom/svg/resources/svg-graphics-attrib.mod
No license file was found, but licenses were detected in source scan.

<!-- ....................................................................... -->
<!-- SVG 1.1 XLink Attribute Module ........................................ -->
<!-- file: svg-xlink-attrib.mod
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-xlink-attrib.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

    PUBLIC "-//W3C//ENTITIES SVG 1.1 XLink Attribute//EN"
    SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-xlink-attrib.mod"

....................................................................... -->
<!-- XLink Attribute

    type, href, role, arcrole, title, show, actuate

This module defines the XLink, XLinkRequired, XLinkEmbed, and XLinkReplace attribute set.
-->
<ENTITY % SVG.XLinkReplace.extra.attrib "" >

<ENTITY % SVG.XLinkReplace.attrib
 "%XLINK.xmlns.attrib;
 %XLINK.pfx;type ( simple ) #FIXED 'simple'
 %XLINK.pfx:href %URI.datatype; #REQUIRED
 %XLINK.pfx;role %URI.datatype; #IMPLIED
 %XLINK.pfx;arcrole %URI.datatype; #IMPLIED
 %XLINK.pfx:title CDATA #IMPLIED
 %XLINK.pfx;show ( new | replace ) 'replace'
 %XLINK.pfx;actuate ( onRequest ) #FIXED 'onRequest'
 %SVG.XLinkReplace.extra.attrib;" >

<!-- end of svg-xlink-attrib.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-
jar/org/apache/batik/dom/svg/resources/svg-xlink-attrib.mod
No license file was found, but licenses were detected in source scan.

<!-- .......................................................... -->
<!-- SVG 1.1 Scripting Module ...................................... -->
<!-- file: svg-script.mod

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-script.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Scripting//EN"
SYSTEM "http://www.w3.org/Specifications/1.1/DTD/svg-script.mod"

.......................................................... -->

<!-- Scripting

script

This module declares markup to provide support for scripting.
-->

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.script.qname "script" >
This is SVG, a language for describing two-dimensional graphics in XML. Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved. Revision: $Id: svg-font.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Font//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-font.mod"

............................................................... -->

<!-- Font
font, font-face, glyph, missing-glyph, hkern, vkern, font-face-src,
font-face-uri, font-face-format, font-face-name, definition-src

This module declares markup to provide support for template.
-->

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.font.qname "font" >
<!ENTITY % SVG.font-face.qname "font-face" >
<!ENTITY % SVG.glyph.qname "glyph" >
<!ENTITY % SVG.missing-glyph.qname "missing-glyph" >
<!ENTITY % SVG.hkern.qname "hkern" >
<!ENTITY % SVG.vkern.qname "vkern" >
<!ENTITY % SVG.font-face-src.qname "font-face-src" >
<!ENTITY % SVG.font-face-uri.qname "font-face-uri" >
<!ENTITY % SVG.font-face-format.qname "font-face-format" >
<!ENTITY % SVG.font-face-name.qname "font-face-name" >
<!ENTITY % SVG.definition-src.qname "definition-src" >

</!-- Attribute Collections (Default) ................... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Container.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Text.attrib "" >
<!ENTITY % SVG.TextContent.attrib "" >
<!ENTITY % SVG.Font.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Marker.attrib "" >
<!ENTITY % SVG.Profile.attrib "" >
<!ENTITY % SVG.Gradient.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.FilterColor.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.XLinkRequired.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Font.class .................................... -->
<!ENTITY % SVG.Font.extra.class "" >
<!ENTITY % SVG.Font.class "| %SVG.font.qname; | %SVG.font-face.qname; %SVG.Font.extra.class;" >

<!-- SVG.Presentation.attrib ........................... -->
<!ENTITY % SVG.Presentation.extra.attrib "" >
<!ENTITY % SVG.Presentation.attrib "| %SVG.Container.attrib; %SVG.Viewport.attrib; %SVG.Text.attrib; %SVG.TextContent.attrib; %SVG.Font.attrib; %SVG.Paint.attrib; %SVG.Color.attrib; %SVG.Opacity.attrib; %SVG/Graphics.attrib; %SVG.Marker.attrib; %SVG.Profile.attrib; %SVG.Gradient.attrib; %SVG.Clip.attrib; %SVG.Mask.attrib; %SVG.Filter.attrib; %SVG.FilterColor.attrib; %SVG.Cursor.attrib; flood-color %SVGColor.datatype; #IMPLIED flood-opacity %OpacityValue.datatype; #IMPLIED lighting-color %SVGColor.datatype; #IMPLIED %SVG.Presentation.extra.attrib;" >

<!-- font: Font Element ................................ -->
<!ENTITY % SVG.font.extra.content "" >
Open Source Used In DNAC Platform Antelope 13441
%SVG.Core.attrib;
        string CDATA #IMPLIED
    >
    <!-- end of SVG.font-face-format.attlist -->]

<!-- font-face-name: Font Face Name Element ............ -->

<!ENTITY % SVG.font-face-name.element "INCLUDE">
<!ENTITY %SVG.font-face-name.element:

<!ENTITY % SVG.font-face-name.content "EMPTY">
<!ENTITY %SVG.font-face-name.qname; %SVG.font-face-name.content; >
<!-- end of SVG.font-face-name.element --]>

<!ENTITY % SVG.font-face-name.attlist "INCLUDE">
<!ENTITY %SVG.font-face-name.attlist:

<!ATTLIST %SVG.font-face-name.qname;
    %SVG.Core.attrib;
    name CDATA #IMPLIED
    >
    <!-- end of SVG.font-face-name.attlist -->]

<!-- definition-src: Definition Source Element ......... -->

<!ENTITY % SVG.definition-src.element "INCLUDE">
<!ENTITY %SVG.definition-src.element:

<!ENTITY % SVG.definition-src.content "EMPTY">
<!ENTITY %SVG.definition-src.qname; %SVG.definition-src.content; >
<!-- end of SVG.definition-src.element --]>

<!ENTITY % SVG.definition-src.attlist "INCLUDE">
<!ENTITY %SVG.definition-src.attlist:

<!ATTLIST %SVG.definition-src.qname;
    %SVG.Core.attrib;
    %SVG.XLinkRequired.attrib;
    >
    <!-- end of SVG.definition-src.attlist -->]

<!-- end of svg-font.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/resources/svg-font.mod

No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Found in path(s):
No license file was found, but licenses were detected in source scan.

SVG 1.1 Tiny DTD

This is SVG Tiny, a proper subset of SVG.

The Scalable Vector Graphics (SVG)
(Massachusetts Institute of Technology, Institut National de
Recherche en Informatique et en Automatique, Keio University).
All Rights Reserved.

Permission to use, copy, modify and distribute the SVG DTD and its
accompanying documentation for any purpose and without fee is hereby
granted in perpetuity, provided that the above copyright notice and
this paragraph appear in all copies. The copyright holders make no
representation about the suitability of the DTD for any purpose.

It is provided "as is" without expressed or implied warranty.

Author: Jun Fujisawa <fujisawa.jun@canon.co.jp>
Revision: $Id: svg11

dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg11

Found in path(s):
No license file was found, but licenses were detected in source scan.

<!-- ....................................................................... -->
<!-- SVG 1.1 Tiny Document Model Module .................................... -->
This is SVG Tiny, a proper subset of SVG.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg11-tiny-model.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Tiny Document Model//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-tiny-model.mod"

<!-- SVG 1.1 Tiny Document Model

This module describes the groupings of elements that make up
common content models for SVG elements.
-->

<!-- Content Models (Default) ......................... -->

<!ENTITY % SVG.Marker.class "" >
<!ENTITY % SVG.Profile.class "" >
<!ENTITY % SVG.Gradient.class "" >
<!ENTITY % SVG.Pattern.class "" >
<!ENTITY % SVG.Clip.class "" >
<!ENTITY % SVG.Mask.class "" >
<!ENTITY % SVG.Filter.class "" >
<!ENTITY % SVG.FilterPrimitive.class "" >
<!ENTITY % SVG.Cursor.class "" >

<!-- module: svg-basic-structure.mod ................. -->

<!ENTITY % SVG.Description.extra.class "" >
<!ENTITY % SVG.Description.class
 "%SVG.desc.qname; | %SVG.title.qname; | %SVG.metadata.qname;
 %SVG.Description.extra.class;"
 >

<!ENTITY % SVG.Use.extra.class "" >
<!ENTITY % SVG.Use.class
 "| %SVG.use.qname; %SVG.Use.extra.class;"
 >

<!ENTITY % SVG.Structure.extra.class "" >
<!ENTITY % SVG.Structure.class
 "| %SVG.svg.qname; | %SVG.g.qname; | %SVG.defs.qname; %SVG.Use.class;
 %SVG.Structure.extra.class;"
<!-- SVG 1.1 Animation Module ............................................... -->
<!-- file: svg-animation.mod -->

This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-animation.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

    PUBLIC "-//W3C//ELEMENTS SVG 1.1 Animation//EN"
    SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-animation.mod"

............................................................... -->

<!-- Animation -->

animate, set, animateMotion, animateColor, animateTransform, mpath

This module declares markup to provide support for animation.
-->

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.animate.qname "animate" >
<!ENTITY % SVG.set.qname "set" >
<!ENTITY % SVG.animateMotion.qname "animateMotion" >
<!ENTITY % SVG.animateColor.qname "animateColor" >
<!ENTITY % SVG.animateTransform.qname "animateTransform" >
<!ENTITY % SVG.mpath.qname "mpath" >

<!-- Attribute Collections (Default) ................... -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.AnimationEvents.attrib "" >
<!ENTITY % SVG.XLink.attrib "" >
<!ENTITY % SVG.XLinkRequired.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Animation.class ............................... -->

<!ENTITY % SVG.Animation.class "" >

<!ENTITY % SVG.Animation.class
"%SVG.animate.qname; | %SVG.set.qname; | %SVG.animateMotion.qname; |
%SVG.animateColor.qname; | %SVG.animateTransform.qname; |
%SVG.Animation.extra.class;" >
"( %SVG.Description.class; %SVG.set.extra.content; )*"
>
<ENTITY %SVG.set.attlist "INCLUDE" >
<![%SVG.set.attlist;[
<!ATTLIST %SVG.set.qname;
%SVG.Core.attrib;
%SVG.Conditional.attrib;
%SVG.AnimationEvents.attrib;
%SVG.External.attrib;
%SVG.Animation.attrib;
%SVG.AnimationAttribute.attrib;
%SVG.AnimationTiming.attrib;
to CDATA #IMPLIED
>
<![-- end of SVG.set.attlist --]]>

<!-- animateMotion: Animate Motion Element .......... -->

<ENTITY %SVG.animateMotion.attlist "INCLUDE" >
<![%SVG.animateMotion.attlist;[
<!ENTITY %SVG.animateMotion.content
"(( %SVG.Description.class; )*, %SVG.mpath.qname;?
%SVG.animateMotion.extra.content; )"
>
<!ELEMENT %SVG.animateMotion.qname; %SVG.animateMotion.content; >
<![-- end of SVG.animateMotion.content --]]>

<!ENTITY %SVG.animateMotion.attlist "INCLUDE" >
<![%SVG.animateMotion.attlist;[
<!ATTLIST %SVG.animateMotion.qname;
%SVG.Core.attrib;
%SVG.Conditiona.attrib;
%SVG.AnimationEvents.attrib;
%SVG.External.attrib;
%SVG.Animation.attrib;
%SVG.AnimationTiming.attrib;
%SVG.AnimationAttribute.attrib;
%SVG.AnimationAddtion.attrib;
calcMode ( discrete | linear | paced | spline ) 'paced'
values CDATA #IMPLIED
keyTimes CDATA #IMPLIED
keySplings CDATA #IMPLIED
from CDATA #IMPLIED
to CDATA #IMPLIED
<![%SVG.animateTransform.attlist;]
<!ATTLIST %SVG.animateTransform.qname;
 %SVG.Core.attrib;
 %SVG.Conditional.attrib;
 %SVG.AnimationEvents.attrib;
 %SVG.External.attrib;
 %SVG.Animation.attrib;
 %SVG.AnimationAttribute.attrib;
 %SVG.AnimationTiming.attrib;
 %SVG.AnimationValue.attrib;
 %SVG.AnimationAddition.attrib;
 type ( translate | scale | rotate | skewX | skewY ) 'translate'
 >
 <!-- end of SVG.animateTransform.attlist -->]]>

<!-- mpath: Motion Path Element ......................... -->

<!ENTITY % SVG.mpath.extra.content "" >

<!ENTITY % SVG.mpath.element "INCLUDE" >
<![%SVG.mpath.element;[
 <!ENTITY % SVG.mpath.content
 "( %SVG.Description.class; %SVG.mpath.extra.content; )*"
 >
 <!ELEMENT %SVG.mpath.qname; %SVG.mpath.content; >
 <!-- end of SVG.mpath.element -->]]>

<!ENTITY % SVG.mpath.attlist "INCLUDE" >
<![%SVG.mpath.attlist;[
 <!ATTLIST %SVG.mpath.qname;
 %SVG.Core.attrib;
 %SVG.XLinkRequired.attrib;
 %SVG.External.attrib;
 >
 <!-- end of SVG.mpath.attlist -->]]>

<!-- end of svg-animation.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1-jar/org/apache/batik/dom/svg/resources/svg-animation.mod
No license file was found, but licenses were detected in source scan.

<!-- ....................................................................... -->
<!-- SVG 1.1 Viewport Attribute Module ..................................... -->
<!-- file: svg-viewport-attrib.mod
This is SVG, a language for describing two-dimensional graphics in XML. -->
This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ENTITIES SVG 1.1 Viewport Attribute//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-viewport-attrib.mod"

This module defines the Viewport attribute set.

<!ENTITY % ClipValue.datatype "CDATA" >

<!ENTITY % SVG.clip.attrib 
"clip %ClipValue.datatype; #IMPLIED" >

<!ENTITY % SVG.overflow.attrib 
"overflow ( visible | hidden | scroll | auto | inherit ) #IMPLIED" >

<!ENTITY % SVG.Viewport.extra.attrib 
"" >

<!ENTITY % SVG.Viewport.attrib 
"%SVG.clip.attrib;
 %SVG.overflow.attrib;
 %SVG.Viewport.extra.attrib;" >

<!-- end of svg-viewport-attrib.mod -->

Found in path(s):
* /opt/cola/permits/1146878524_1616835142.61/0/batik-svg-dom-1-13-sources-1.jar/org/apache/batik/dom/svg/resources/svg-viewport-attrib.mod
No license file was found, but licenses were detected in source scan.

This is SVG, a language for describing two-dimensional graphics in XML.
This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "//W3C//ELEMENTS SVG 1.1 Basic Text//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-basic-text.mod"

<!-- Basic Text

text, altGlyph, altGlyphDef, glyphRef

This module declares markup to provide support for text. -->

<!ENTITY % FontFamilyValue.datatype "CDATA" >

<!ENTITY % FontSizeValue.datatype "CDATA" >

<!ENTITY % SVG.text.qname "text" >
<!ENTITY % SVG.altGlyph.qname "altGlyph" >
<!ENTITY % SVG.altGlyphDef.qname "altGlyphDef" >
<!ENTITY % SVG.glyphRef.qname "glyphRef" >

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Paint.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.GraphicalEvents.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.XLink.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!ENTITY % SVG.Text.class "" >
jar/org/apache/batik/dom/svg/resources/svg-basic-text.mod
No license file was found, but licenses were detected in source scan.

<!-- .......................................................... -->
<!-- SVG 1.1 Image Module ....................................... -->
<!-- file: svg-image.mod
This is SVG, a language for describing two-dimensional graphics in XML.
Copyright 2001, 2002 W3C (MIT, INRIA, Keio), All Rights Reserved.
Revision: $Id: svg-image.mod 1733416 2016-03-03 07:07:13Z gadams $

This DTD module is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//ELEMENTS SVG 1.1 Image//EN"
SYSTEM "http://www.w3.org/Graphics/SVG/1.1/DTD/svg-image.mod"

.......................................................... -->

<!-- Image

image

This module declares markup to provide support for image.
-->

<!-- Qualified Names (Default) ......................... -->

<!ENTITY % SVG.image.qname "image" >

<!-- Attribute Collections (Default) ................. -->

<!ENTITY % SVG.Core.attrib "" >
<!ENTITY % SVG.Conditional.attrib "" >
<!ENTITY % SVG.Style.attrib "" >
<!ENTITY % SVG.Viewport.attrib "" >
<!ENTITY % SVG.Color.attrib "" >
<!ENTITY % SVG.Opacity.attrib "" >
<!ENTITY % SVG.Graphics.attrib "" >
<!ENTITY % SVG.Profile.attrib "" >
<!ENTITY % SVG.Clip.attrib "" >
<!ENTITY % SVG.Mask.attrib "" >
<!ENTITY % SVG.Filter.attrib "" >
<!ENTITY % SVG.GraphicalEvents.attrib "" >
<!ENTITY % SVG.Cursor.attrib "" >
<!ENTITY % SVG.XLinkEmbed.attrib "" >
<!ENTITY % SVG.External.attrib "" >

<!-- SVG.Image.class ................................... -->
1.3108 drill-jdbc 1.19.0
1.3108.1 Available under license:

Drill : Exec : JDBC Driver using dependencies
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3109 qs 6.5.2

1.3109.1 Available under license:

Copyright (c) 2014 Nathan LaFreniere and other contributors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE
FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors

1.3110 worker-loader 1.1.1 1.1.1
1.3110.1 Available under license :

Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3111 h2 1.4.200
1.3111.1 Available under license :

Mozilla Public License
Version 2.0
1. Definitions
1.1. “Contributor”
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. “Contributor Version”
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.

1.3. “Contribution”
means Covered Software of a particular Contributor.

1.4. “Covered Software”
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. “Incompatible With Secondary Licenses”
means
that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or
that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

1.6. “Executable Form”
means any form of the work other than Source Code Form.

1.7. “Larger Work”
means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

1.8. “License”
means this document.

1.9. “Licensable”
means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

1.10. “Modifications”
means any of the following:
any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or
any new file in Source Code Form that contains any Covered Software.

1.11. “Patent Claims” of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. “Secondary License”
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.
1.13. “Source Code Form”
means the form of the work preferred for making modifications.

1.14. “You” (or “Your”)
means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any
entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control”
means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or
otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of
such entity.

2. License Grants and Conditions
2.1. Grants
Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce,
make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an
unmodified basis, with Modifications, or as part of a Larger Work; and

under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer
either its Contributions or its Contributor Version.

2.2. Effective Date
The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the
date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope
The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses
will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section
2.1(b) above, no patent license is granted by a Contributor:

for any code that a Contributor has removed from Covered Software; or

for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the
combination of its Contributions with other software (except as part of its Contributor Version); or

under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may
be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses
No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a
subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under
the terms of Section 3.3).

2.5. Representation
Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has
sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use
This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions
Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities
3.1. Distribution of Source Form
All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients’ rights in the Source Code Form.

3.2. Distribution of Executable Form
If You distribute Covered Software in Executable Form then:

such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients’ rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work
You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices
You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms
You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.
4. Inability to Comply Due to Statute or Regulation
If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination
5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

6. Disclaimer of Warranty
Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

7. Limitation of Liability
Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

8. Litigation
Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant
maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party’s ability to bring cross-claims or counter-claims.

9. Miscellaneous
This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License
10.1. New Versions
Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions
You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions
If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses
If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice
This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - “Incompatible With Secondary Licenses” Notice
This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.
H2 is dual licensed and available under the MPL 2.0 (Mozilla Public License Version 2.0) or under the EPL 1.0 (Eclipse Public License).

1.3112 async 1.0.0
1.3112.1 Available under license:
Copyright (c) 2010 StumbleUpon, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the StumbleUpon nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3113 velocity-engine-core 2.0
1.3113.1 Available under license:

Apache Velocity - Engine
Copyright 2000-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache Commons IO
Copyright 2002-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

1.3114 jsreport-licensing 2.2.3

1.3114.1 Available under license:

No license file was found, but licenses were detected in source scan.

```
license: 'enterprise',
license: body.license,
license: definition.options.license,
```

Found in path(s):
* /opt/cola/permits/1226014123_1636487531.07/0/jsreport-jsreport-licensing-2-2-3-0-g3215556-tar-gz/jsreport-jsreport-licensing-3215556/lib/licensing.js

1.3115 react-is 16.13.1

1.3115.1 Available under license:

MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3116 range-parser 1.0.3
1.3116.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3117 socket.io-parser 3.4.1
1.3117.1 Available under license:
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache Xerces Java
Copyright 1999-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- voluntary contributions made by Paul Eng on behalf of the
  Apache Software Foundation that were originally developed at iClick, Inc.,
  software copyright (c) 1999.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3119 snake-yaml 1.29

1.3119.1 Available under license :

No license file was found, but licenses were detected in source scan.

<name>Apache License, Version 2.0</name>
<url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>

Found in path(s):
* /opt/cola/permits/1175979988_1624558313.98/0/snakeyaml-1-29-jar/META-INF/maven/org.yaml/snakeyaml/pom.xml
No license file was found, but licenses were detected in source scan.

Manifest-Version: 1.0
Automatic-Module-Name: org.yaml.snakeyaml
Bnd-LastModified: 1623172599972
Build-Jdk: 11.0.11
Built-By: alex13
Bundle-Description: YAML 1.1 parser and emitter for Java
Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.txt
Bundle-ManifestVersion: 2
Bundle-Name: SnakeYAML
Bundle-SymbolicName: org.yaml.snakeyaml
Bundle-Version: 1.29.0
Created-By: Apache Maven Bundle Plugin
Export-Package: org.yaml.snakeyaml;version="1.29",org.yaml.snakeyaml.comments;version="1.29",org.yaml.snakeyaml.composer;version="1.29",org.yaml.snakeyaml.constructor;version="1.29",org.yaml.snakeyaml.emitter;version="1.29",org.yaml.snakeyaml.env;version="1.29",org.yaml.snakeyaml.error;version="1.29",org.yaml.snakeyaml.events;version="1.29",org.yaml.snakeyaml.extensions.compactnotation;version="1.29",org.yaml.snakeyaml.introspector;version="1.29",org.yaml.snakeyaml.nodes;version="1.29",org.yaml.snakeyaml.parser;version="1.29",org.yaml.snakeyaml.reader;version="1.29",org.yaml.snakeyaml.representer;version="1.29",org.yaml.snakeyaml.resolver;version="1.29",org.yaml.snakeyaml.scanner;version="1.29",org.yaml.snakeyaml.tokens;version="1.29",org.yaml.snakeyaml.util;version="1.29"
Import-Package: org.yaml.snakeyaml;version="[1.29,2)",org.yaml.snakeyaml.comments;version="[1.29,2)",org.yaml.snakeyaml.composer;version="[1.29,2)",org.yaml.snakeyaml.constructor;version="[1.29,2)",org.yaml.snakeyaml.error;version="[1.29,2)",org.yaml.snakeyaml.events;version="[1.29,2)",org.yaml.snakeyaml.introspector;version="[1.29,2)",org.yaml.snakeyaml.nodes;version="[1.29,2)",org.yaml.snakeyaml.parser;version="[1.29,2)",org.yaml.snakeyaml.reader;version="[1.29,2)",org.yaml.snakeyaml.resolver;version="[1.29,2)",org.yaml.snakeyaml.scanner;version="[1.29,2)",org.yaml.snakeyaml.tokens;version="[1.29,2)"
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.7))"
Tool: Bnd-3.5.0.201709291849

1.3120 cglib 2.2.2
1.3120.1 Available under license:

========================================================================
== NOTICE file corresponding to the section 4 d of                     ==
== the Apache License, Version 2.0,                                    ==
== in this case for the Apache Ant distribution.                     ==
========================================================================
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

This product includes also software developed by:
- the W3C consortium (http://www.w3c.org),
- the SAX project (http://www.saxproject.org)

Please read the different LICENSE files present in the root directory of
this distribution.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<title>Common Public License - v 1.0</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF" vlink="#800000">

<p align="center"><b>Common Public License - v 1.0</b></p>
<p><b></b><font size="3"></font></p>
<p><font size="3"></font><font size="2">THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.</font></p>
<p><font size="2"></font></p>
<p><font size="2"><b>1. DEFINITIONS</b></font></p>
<p><font size="2">"Contribution" means:</font></p>
<ul>
  <li>a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and</li>
  <li>b) in the case of each subsequent Contributor:</li>
</ul>
<ul>
  <li>i) changes to the Program, and</li>
  <li>ii) additions to the Program;</li>
</ul>

<ul>
  <li>where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. </li>
</ul>
<ul>
  <li>A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.</li>
</ul>
<ul>
  <li>Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.</li>
</ul>
"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
<UL><FONT SIZE="2">

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement. </FONT></UL>

<UL><FONT SIZE="2"><P><B>3. REQUIREMENTS</B></P><B>A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

<UL><FONT SIZE="2">a) it complies with the terms and conditions of this Agreement; and

<UL><FONT SIZE="2">b) its license agreement:

<UL><FONT SIZE="2">i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

<UL><FONT SIZE="2">ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

<UL><FONT SIZE="2">iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

<UL><FONT SIZE="2">iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

</UL></UL></UL></UL></UL></UL>

<UL><FONT SIZE="2" COLOR="#FF0000">When the Program is made available in source code form:

<UL><FONT SIZE="2">a) it must be made available under this Agreement; and

<UL><FONT SIZE="2">b) a copy of this Agreement must be included with each copy of the Program.

</UL></UL></UL></UL>
Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

Except as expressly set forth in this Agreement, the Program is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

Except as expressly set forth in this Agreement, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS).
SIZE="2"> HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.</FONT>

<P><FONT SIZE="2"></FONT><FONT SIZE="2">B7. GENERAL</FONT></P>

<P><FONT SIZE="2"></FONT><FONT SIZE="2">If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.</FONT></P>

<P><FONT SIZE="2"></FONT><FONT SIZE="2">If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.</FONT></P>

<P><FONT SIZE="2"></FONT><FONT SIZE="2">All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.</FONT></P>

<P><FONT SIZE="2"></FONT><FONT SIZE="2">Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.</FONT></P>

<P><FONT SIZE="2"></FONT><FONT SIZE="2">This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.</FONT></P>
ASM: a very small and fast Java bytecode manipulation framework
Copyright (c) 2000,2002,2003 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS”
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

/*
  Apache License
  Version 2.0, January 2004
  http://www.apache.org/licenses/
*/

* TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
 *
* 1. Definitions.
 *
* "License" shall mean the terms and conditions for use, reproduction,
* and distribution as defined by Sections 1 through 9 of this document.
 *
* "Licensor" shall mean the copyright owner or entity authorized by
* the copyright owner that is granting the License.
 *
* "Legal Entity" shall mean the union of the acting entity and all
* other entities that control, are controlled by, or are under common
* control with that entity. For the purposes of this definition,
* "control" means (i) the power, direct or indirect, to cause the
* direction or management of such entity, whether by contract or
* otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable
* copyright license to reproduce, prepare Derivative Works of,
* publicly display, publicly perform, sublicense, and distribute the
* Work and such Derivative Works in Source or Object form.
*
* 3. Grant of Patent License. Subject to the terms and conditions of
* this License, each Contributor hereby grants to You a perpetual,
* worldwide, non-exclusive, no-charge, royalty-free, irrevocable
* (except as stated in this section) patent license to make, have made,
* use, offer to sell, sell, import, and otherwise transfer the Work,
* where such license applies only to those patent claims licensable
* by such Contributor that are necessarily infringed by their
* Contribution(s) alone or by combination of their Contribution(s)
* with the Work to which such Contribution(s) was submitted. If You
* institute patent litigation against any entity (including a
* cross-claim or counterclaim in a lawsuit) alleging that the Work
* or a Contribution incorporated within the Work constitutes direct
* or contributory patent infringement, then any patent licenses
* granted to You under this License for that Work shall terminate
* as of the date such litigation is filed.
*
* 4. Redistribution. You may reproduce and distribute copies of the
* Work or Derivative Works thereof in any medium, with or without
* modifications, and in Source or Object form, provided that You
* meet the following conditions:
*
* (a) You must give any other recipients of the Work or
* Derivative Works a copy of this License; and
*
* (b) You must cause any modified files to carry prominent notices
* stating that You changed the files; and
*
* (c) You must retain, in the Source form of any Derivative Works
* that You distribute, all copyright, patent, trademark, and
* attribution notices from the Source form of the Work,
* excluding those notices that do not pertain to any part of
* the Derivative Works; and
*
* (d) If the Work includes a "NOTICE" text file as part of its
* distribution, then any Derivative Works that You distribute must
* include a readable copy of the attribution notices contained
* within such NOTICE file, excluding those notices that do not
* pertain to any part of the Derivative Works, in at least one
* of the following places: within a NOTICE text file distributed
* as part of the Derivative Works; within the Source form or
* documentation, if provided along with the Derivative Works; or,
* within a display generated by the Derivative Works, if and
* wherever such third-party notices normally appear. The contents
* of the NOTICE file are for informational purposes only and
* do not modify the License. You may add Your own attribution
* notices within Derivative Works that You distribute, alongside
* or as an addendum to the NOTICE text from the Work, provided
* that such additional attribution notices cannot be construed
* as modifying the License.
*
* You may add Your own copyright statement to Your modifications and
* may provide additional or different license terms and conditions
* for use, reproduction, or distribution of Your modifications, or
* for any such Derivative Works as a whole, provided Your use,
* reproduction, and distribution of the Work otherwise complies with
* the conditions stated in this License.
*
* 5. Submission of Contributions. Unless You explicitly state otherwise,
* any Contribution intentionally submitted for inclusion in the Work
* by You to the Licensor shall be under the terms and conditions of
* this License, without any additional terms or conditions.
* Notwithstanding the above, nothing herein shall supersede or modify
* the terms of any separate license agreement you may have executed
* with Licensor regarding such Contributions.
*
* 6. Trademarks. This License does not grant permission to use the trade
* names, trademarks, service marks, or product names of the Licensor,
* except as required for reasonable and customary use in describing the
* origin of the Work and reproducing the content of the NOTICE file.
*
* 7. Disclaimer of Warranty. Unless required by applicable law or
* agreed to in writing, Licensor provides the Work (and each
* Contributor provides its Contributions) on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
* implied, including, without limitation, any warranties or conditions
* of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
* PARTICULAR PURPOSE. You are solely responsible for determining the
* appropriateness of using or redistributing the Work and assume any
* risks associated with Your exercise of permissions under this License.
*
* 8. Limitation of Liability. In no event and under no legal theory,
* whether in tort (including negligence), contract, or otherwise,
* unless required by applicable law (such as deliberate and grossly
* negligent acts) or agreed to in writing, shall any Contributor be
* liable to You for damages, including any direct, indirect, special,
* incidental, or consequential damages of any character arising as a
* result of this License or out of the use or inability to use the
* Work (including but not limited to damages for loss of goodwill,
* work stoppage, computer failure or malfunction, or any and all
* other commercial damages or losses), even if such Contributor
* has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

---

Open Source Used In DNAC Platform Antelope 13500
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.3121 drill-common 1.19.0
1.3121.1 Available under license:

Drill : Common
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.
"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a ”NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.
To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3122 async-limiter 1.0.1
1.3122.1 Available under license:

The MIT License (MIT)
Copyright (c) 2017 Samuel Reed <samuel.trace.reed@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
## Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

### Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

### Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses. This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

## jetty-client 9.3.27.v20190418

## any-promise 1.3.0

Copyright (C) 2014-2016 Kevin Beaty

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3126 get-caller-file 2.0.5 2.0.5

1.3126.1 Available under license:
   ISC License (ISC)
   Copyright 2018 Stefan Penner

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.3127 drill-format-syslog 1.19.0

1.3127.1 Available under license:
   Drill : Contrib : Format : Syslog
   Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.3128 buffer-alloc 1.2.0
1.3128.1 Available under license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3129 httpcore5 5.0.2
1.3129.1 Available under license:

Apache HttpComponents Core HTTP/1.1
Copyright 2005-2020 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3130 babel-plugin-transform-regenerator
6.26.0

1.3130.1 Available under license:

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The << Software identified by reference to the Ecma Standard* ("Software") >> is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT http://www.ecma-international.org/memento/codeofconduct.htm FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS*.

Copyright (C) 2012-2013 Ecma International
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
   following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived
   from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTicular PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

* Ecma International Standards hereafter means Ecma International Standards as well as Ecma Technical Reports
Copyright (c) 2014-2015 Jason Quense <jason@quense.me>
Original work by respective rule authors; copywrites noted in files.

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without
Open Source Used In DNAC Platform Antelope 13521

modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License
Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)
Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3131 ws 7.3.1

1.3131.1 Available under license :
(The MIT License)

Copyright (c) 2017-2020 Kamil Mysliwiec <https://kamilmysliwiec.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3132 netty 4.1.42.Final

1.3132.1 Available under license :
Copyright (c) 2003-2008 Yuta Mori All Rights Reserved.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright (yyyy) [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

The Netty Project

Please visit the Netty web site for more information:

* https://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

* LICENSE:
  * license/LICENSE.jsr166y.txt (Public Domain)
* HOMEPAGE:
  * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
  * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

* LICENSE:
  * license/LICENSE.base64.txt (Public Domain)
* HOMEPAGE:
This product contains a modified portion of 'Webbit', an event based WebSocket and HTTP server, which can be obtained at:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product contains a modified portion of 'Apache Harmony', an open source Java SE, which can be obtained at:

* NOTICE:
  * license/NOTICE.harmony.txt
* LICENSE:
  * license/LICENSE.harmony.txt (Apache License 2.0)
* HOMEPAGE:
  * http://archive.apache.org/dist/harmony/

This product contains a modified portion of 'jbzip2', a Java bzip2 compression and decompression library written by Matthew J. Francis. It can be obtained at:

* LICENSE:
  * license/LICENSE.jbzip2.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct the suffix array and the Burrows-Wheeler transformed string for any input string of a constant-size alphabet written by Yuta Mori. It can be obtained at:

* LICENSE:
  * license/LICENSE.libdivsufsort.txt (MIT License)
* HOMEPAGE:
  * https://github.com/y-256/libdivsufsort

This product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM, which can be obtained at:
This product optionally depends on 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD style License)
* HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

* LICENSE:
  * license/LICENSE.compress-lzf.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:
This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/snappy

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jboss-remoting/jboss-marshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/caliper

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:
This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasternxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

* LICENSE:
  * license/LICENSE.hpack.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/twitter/hpack

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Cory Benfield. It can be obtained at:

* LICENSE:
  * license/LICENSE.hyper-hpack.txt (MIT License)
* HOMEPAGE:
  * https://github.com/python-hyper/hpack/

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Tatsuhiro Tsujikawa. It can be obtained at:

* LICENSE:
  * license/LICENSE.nghttp2-hpack.txt (MIT License)
* HOMEPAGE:
  * https://github.com/nghttp2/nghttp2/

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-lang.txt (Apache License 2.0)
* HOMEPAGE:
  * https://commons.apache.org/proper/commons-lang/

This product contains the Maven wrapper scripts from 'Maven Wrapper', that provides an easy way to ensure a user
has everything necessary to run the Maven build.

* LICENSE:
  * license/LICENSE.mvn-wrapper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/takari/maven-wrapper

The MIT License

Copyright (c) 2009 William Kinney

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This copy of Aalto XML processor is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/

A copy is also included with both the the downloadable source code package and jar that contains class bytecodes, as file "ASL 2.0". In both cases, that file should be located next to this file: in source distribution the location should be "release-notes/asl"; and in jar "META-INF/" (BSD License: http://www.opensource.org/licenses/bsd-license)

Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Webbit nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License (MIT)

Copyright (c) 2014 Cory Benfield

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Harmony

Copyright 2006, 2010 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Copyright (c) 2010-2011 Matthew J. Francis and Contributors of the jBzip2 Project

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

Open Source Used In DNAC Platform Antelope 13635
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright 2009-2010 Ning, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicatars holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuate of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS.
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa
Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT,
INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*
 * Copyright (c) 2004-2007 QOS.ch
 * All rights reserved.
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
 * LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
 * OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
 * WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 */

Protocol Buffers - Google's data interchange format
Copyright 2013 Google Inc. All rights reserved.
https://developers.google.com/protocol-buffers/

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following disclaimer
  in the documentation and/or other materials provided with the
  distribution.
* Neither the name of Google Inc. nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.
"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. 
You may obtain a copy of the License at 

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software 
distributed under the License is distributed on an "AS IS" BASIS, 
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. 
See the License for the specific language governing permissions and 
limitations under the License.

1.3133 to-fast-properties 1.0.3 1.0.3
1.3133.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Petka Antonov
2015 Sindre Sorhus

Permission is hereby granted, free of charge, to any person obtaining a copy 
of this software and associated documentation files (the "Software"), to deal 
in the Software without restriction, including without limitation the rights 
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell 
copies of the Software, and to permit persons to whom the Software is 
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in 
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR 
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, 
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE 
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER 
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, 
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN 
THE SOFTWARE.

1.3134 which-module 2.0.0 2.0.0
1.3134.1 Available under license:
Copyright (c) 2016, Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose 
with or without fee is hereby granted, provided that the above copyright notice 
and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.3135 bean-validation-api 1.1.0.Final

1.3136 to-array 0.1.4
1.3136.1 Available under license:

Copyright (c) 2012 Raynos.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3137 tinycolor2 1.4.1
1.3137.1 Available under license:

Copyright (c), Brian Grinstead, http://briangrinstead.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3138 bunyan 1.8.1 1.8.1

1.3138.1 Available under license :

# This is the MIT license

Copyright 2016 Trent Mick
Copyright 2016 Joyent Inc.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3139 strict-uri-encode 2.0.0 2.0.0
1.3139.1 Available under license:
The MIT License (MIT)

Copyright (c) Kevin Martensson <kevinmartensson@gmail.com> (github.com/kevva)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3140 wordwrap 1.0.0
1.3140.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2018 Colin Arenz

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3141 javax-jdo 3.2.0-m3

1.3141.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2015-2015 DataNucleus

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
1.3143 @babel/plugin-transform-modules-amd 7.13.0

1.3143.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3144 metrics 0.1.21

1.3144.1 Available under license:
No license file was found, but licenses were detected in source scan.

// licensed under CCv3.0: http://creativecommons.org/licenses/by/3.0/

Found in path(s):
* /opt/cola/permits/1198958050_1630433819.43/0/metrics-0-1-21-tgz/package/lib/binary_heap.js
No license file was found, but licenses were detected in source scan.

The MIT License (MIT)
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
1.3145 buffer-shims 1.0.0

1.3145.1 Available under license:

# Copyright (c) 2016 Calvin Metcalf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

1.3146 ansi-styles 2.2.1

1.3146.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3147 read-pkg-up 2.0.0

1.3147.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3148 libvpx 1.8.1-299-gd2a5e26359

1.3148.1 Available under license:
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2005-2012 x264 project

Authors: Loren Merritt <loremm@u.washington.edu>
    Anton Mitrofanov <BugMaster@narod.ru>
    Jason Garrett-Glaser <darkshikari@gmail.com>
    Henrik Gramner <hengar-6@student.ltu.se>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2010, The WebM Project authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google, nor the WebM Project, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2010, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2011 The LibYuv Project Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# This file contains a list of people who've made non-trivial
# contribution to the Google C++ Testing Framework project. People
# who commit code to the project are encouraged to add their names
# here. Please keep the list sorted by first names.

Ajay Joshi <jaj@google.com>
Balzs Dn <balazs.dan@gmail.com>
Bharat Mediratta <bharat@menalto.com>
Chandler Carruth <chandlerc@google.com>
Chris Prince <cprince@google.com>
Chris Taylor <taylorc@google.com>
Dan Egnor <egnor@google.com>
Eric Roman <eroman@chromium.org>
Hady Zalek <hady.zalek@gmail.com>
Jeffrey Yasskin <jyasskin@gmail.com>
Ji Sigursson <joi@google.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kenton Varda <kenton@google.com>
Manuel Klimek <klimek@gmail.com>
Markus Heule <markus.heule@gmail.com>
Mika Raento <mikie@iki.fi>
1.3149 json5 2.2.0
1.3149.1 Available under license:
MIT License

Copyright (c) 2012-2018 Aseem Kishore, and [others].

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[others]: https://github.com/json5/json5/contributors

1.3150 strip-ansi 4.0.0
1.3150.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3151 @babel/template 7.12.13
1.3151.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3152 through 2.3.8

1.3152.1 Available under license:

The MIT License

Copyright (c) 2011 Dominic Tarr

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License, Version 2.0

Copyright (c) 2011 Dominic Tarr

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3153 namespace 1.4.01
1.3153.1 Available under license:

xml-commons/java/external/LICENSE.dom-software.txt $Id: LICENSE.dom-software.txt 734314 2009-01-14 03:33:27Z mrglavas $

This license came from: http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/java-binding.zip (COPYRIGHT.html)

W3C SOFTWARE NOTICE AND LICENSE

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires "Notice of any changes or modifications to the W3C files, including the date changes were made." Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL definitions, the pragma prefix can no longer be 'w3c.org'; in the case of the Java language binding, the package names can no longer be in the 'org.w3c' package.

Note: The original version of the W3C Software Copyright Notice and License could be found at http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

===========================================================================
==  NOTICE file corresponding to section 4(d) of the Apache License,   ==
==  Version 2.0, in this case for the Apache xml-commons xml-apis      ==
==  distribution.                                                      ==
===========================================================================

Apache XML Commons XML APIs
Copyright 1999-2009 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

This license came from: http://www.w3.org/Consortium/Legal/copyright-documents-20021231

W3C DOCUMENT LICENSE
http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231

Public documents on the W3C site are provided by the copyright holders under the following license. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that
you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright [date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.
   http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

---------------------------------------------------------------------

This formulation of W3C's notice and license became active on December 31 2002.
This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised by Reagle $Date: 2005-06-03 18:49:13 -0400 (Fri, 03 Jun 2005) $

This license came from: http://www.megginson.com/SAX/copying.html
However please note future versions of SAX may be covered under http://saxproject.org/?selected=pd

This page is now out of date -- see the new SAX site at http://www.saxproject.org/ for more up-to-date releases and other information. Please change your bookmarks.

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05

**1.3154 netty-resolver 4.1.59.Final**

**1.3154.1 Available under license :**

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2015 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*/

Found in path(s):

* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/HostsFileParser.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/DefaultAddressResolverGroup.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/DefaultHostsFileEntriesResolver.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/AddressResolver.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/HostsFileEntriesResolver.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/DefaultNameResolver.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/InetNameResolver.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/CompositeNameResolver.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/AbstractAddressResolver.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-jar/io/netty/resolver/InetSocketAddressResolver.java

No license file was found, but licenses were detected in source scan.

~ Copyright 2014 The Netty Project
~
~ The Netty Project licenses this file to you under the Apache License,
~ version 2.0 (the "License"); you may not use this file except in compliance
~ with the License. You may obtain a copy of the License at:
~
~ https://www.apache.org/licenses/LICENSE-2.0
~
~ Unless required by applicable law or agreed to in writing, software
~ distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
~ WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
~ License for the specific language governing permissions and limitations
~ under the License.
Open Source Used in DNAC Platform Antelope 13581

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2014 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2017 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * https://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
*/
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-
  jar/io/netty/resolver/HostsFileEntries.java
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-
  jar/io/netty/resolver/ResolvedAddressTypes.java
No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2016 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 * *
 * https://www.apache.org/licenses/LICENSE-2.0
 * *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

Found in path(s):
* /opt/cola/permits/1145449559_1616156337.74/0/netty-resolver-4-1-59-final-sources-
  jar/io/netty/resolver/RoundRobinInetAddressResolver.java

1.3155 commons-pool 2.6.1

1.3156 okio 2.7.0

1.3156.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
*/opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/-Util.kt
*/opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/internal/-Utf8.kt
*/opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/-Platform.kt
*/opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/ByteString.kt
*/opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/internal/ByteString.kt
*/opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/-Platform.kt
*/opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/ByteString.kt

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
*/opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/ByteString.kt

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/ForwardingTimeout.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/SegmentedByteString.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/SegmentedByteString.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Square, Inc.
* *
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* *
* http://www.apache.org/licenses/LICENSE-2.0
* *
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/SegmentPool.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/Segment.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/Sink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/Buffer.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/RealBufferedSource.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/AsyncTimeout.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/JvmOkio.kt
Open Source Used In DNAC Platform Antelope 13585

* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/SegmentPool.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/GzipSource.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/DeflaterSink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/ForwardingSource.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/Source.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/RealBufferedSink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/InflaterSink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/BufferedSource.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/Timeout.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/GzipSink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/BufferedSink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/ForwardingSink.kt

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2017 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *   http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/Utf8.kt

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2018 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 */
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/-
  DeprecatedOkio.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-
  jar/commonMain/okio/PeekSource.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/-
  DeprecatedUpgrade.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/-
  DeprecatedUtf8.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/Throttler.kt

No license file was found, but licenses were detected in source scan.

/ *
* Copyright (C) 2016 Square, Inc.
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/jvmMain/okio/Pipe.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-
  jar/commonMain/okio/Options.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-
  jar/jvmMain/okio/HashingSource.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-
  jar/jvmMain/okio/HashingSink.kt
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/internal/RealBufferedSink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/internal/RealBufferedSource.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/internal/SegmentedByteString.kt

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (C) 2019 Square, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/Sink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/BufferedSink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/BufferedSink.kt

* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/BufferedSink.kt
jar/commonMain/okio/Buffer.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/Source.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/internal/Buffer.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/RealBufferedSource.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/Timeout.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/RealBufferedSink.kt
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio/Okio.kt

No license file was found, but licenses were detected in source scan.

้าง

* Licensed to the Apache Software Foundation (ASF) under one or more
  * contributor license agreements. See the NOTICE file distributed with
  * this work for additional information regarding copyright ownership.
  * The ASF licenses this file to You under the Apache License, Version 2.0
  * (the "License"); you may not use this file except in compliance with
  * the License. You may obtain a copy of the License at
  *
  * http://www.apache.org/licenses/LICENSE-2.0
  *
  * Unless required by applicable law or agreed to in writing, software
  * distributed under the License is distributed on an "AS IS" BASIS,
  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  * See the License for the specific language governing permissions and
  * limitations under the License.
  */

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1074323481_1595745315.39/0/okio-2-7-0-sources-jar/commonMain/okio-/Base64.kt

1.3157 woodstox-core 6.2.0
1.3157.1 Available under license:
This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
1.3160 jaeger-thrift 0.23.0

1.3160.1 Available under license:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic
mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to whom it is addressed.

Open Source Used In DNAC Platform Antelope 13591
inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3161 jackson 2.11.1

1.3161.1 Available under license:

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.
This copy of Jackson JSON processor streaming parser/generator is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.3162 minimatch 3.0.4
1.3162.1 Available under license:
The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.3163 p-defer 1.0.0
1.3163.1 Available under license:
The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3164 path-to-regexp 0.1.7

1.3164.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Blake Embrey (hello@blakeembrey.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3165 base64id 1.0.0 1.0.0

1.3165.1 Available under license:
(The MIT License)

Copyright (c) 2012-2016 Kristian Faeldt <faeldt_kristian@cyberagent.co.jp>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3166 string.prototype-codepointat 0.2.1

1.3166.1 Available under license:

Copyright Mathias Bynens <https://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3167 class-utils 0.3.6

1.3167.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015, 2017-2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3168 hbase-mapreduce 2.0.0-alpha4
1.3168.1 Available under license:

Apache HBase - MapReduce
Copyright 2007-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3169 @babel/plugin-transform-shorthand-properties 7.12.13
1.3169.1 Available under license :

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3170 string_decoder 0.10.31

1.3170.1 Available under license:
Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3171 htmlparser2 3.10.1
1.3171.1 Available under license:

Copyright 2010, 2011, Chris Winberry <chris@winberry.net>. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3172 union-value 1.0.1

1.3172.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2017, Jon Schlinkert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3173 opentracing-api 0.31.0

1.3173.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."
“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
1.3174 generator-object-property 1.2.0

1.3174.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3175 json-smart 2.3

1.3175.1 Available under license:

No license file was found, but licenses were detected in source scan.

/*
 * Copyright 2011 JSON-SMART authors
 * 
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 * 
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONObject.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONNavi.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONArray.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONStreamAwareEx.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/FakeMapper.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONAwareEx.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONValue.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JStylerObj.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONParserString.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONParserReader.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/BeansMapper.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONStreamAware.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/ParseException.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/ParseException.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/JStylerReader.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/JSONParserReader.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/BeansMapper.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONStreamAware.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/JSONParserReader.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/BeansMapper.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/JSONStreamAware.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/JSONParserReader.java
*/
jar/net/minidev/json/parser/JSONParserBase.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/DefaultMapperOrdered.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/JSONParserMemory.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/JSONParser.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/JSONParserInputStream.java
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/JSONParserByteArray.java
No license file was found, but licenses were detected in source scan.

/*
 *    Copyright 2011-2014 JSON-SMART authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * 
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1177952445_1625583906.57/0/json-smart-2-3-sources-4-jar/net/minidev/json/parser/DefaultMapper.java
1.3176 kotlin 1.4.10
1.3176.1 Available under license:

No license file was found, but licenses were detected in source scan.
Open Source Used In DNAC Platform Antelope

- Collections
  - AbstractSet.kt
  - ArrayDeque.kt
  - Arrays.kt
  - IndexedValue.kt
  - MapAccessors.kt
  - MapWithDefault.kt
  - MutableCollections.kt
  - ReversedViews.kt
  - SequenceBuilder.kt
  - SlidingWindow.kt
  - UArraySorting.kt
- Contracts
  - ContractBuilder.kt
- Coroutines
  - ContinuationInterceptor.kt
  - CoroutineContext.kt
  - CoroutineContextImpl.kt
- Experimental
  - bitwiseOperations.kt
  - inferenceMarker.kt
- Internal
  - Annotations.kt
  - Delegates.kt
  - Interfaces.kt
- Properties
  - ObservableProperty.kt
  - PropertyReferenceDelegates.kt
- SourcesContent
  - (function (root, factory) { if (typeof define === 'function' && define.amd) {
    define('kotlin', ['exports'], factory);
  } else if (typeof exports === 'object') {
    factory(module.exports);
  } else {
    root.kotlin = {};
    factory(root.kotlin);
  })(this,
  function (Kotlin) {
    var _ = Kotlin;
    insertContent();
  }));

- License
  - Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
  - Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.

- Kotlin Functions
  - Kotlin.isBooleanArray
  - Kotlin.isByteArray
  - Kotlin.isShortArray
  - Kotlin.isCharArray
  - Kotlin.isIntArray
  - Kotlin.isFloatArray
  - Kotlin.isDoubleArray
  - Kotlin.isLongArray

- Utility Functions
  - Kotlin.arrayToString
  - Kotlin.arrayDeepToString
  - Kotlin.arrayEquals
  - Kotlin.arrayDeepEquals
  - Kotlin.arrayHashCode
  - Kotlin.arrayDeepHashCode
  - Kotlin.primitiveArraySort
null, implementedInterface: function () { return Kotlin.kotlin.reflect.KMutableProperty0 },

immutable: { value: null, implementedInterface: function () { return Kotlin.kotlin.reflect.KProperty0 },

} },

mutable: { value: null, implementedInterface: function () { return Kotlin.kotlin.reflect.KMutableProperty1 },

} },

immutable: { value: null, implementedInterface: function () { return Kotlin.kotlin.reflect.KProperty1 },

} ],

} ]

;]

function getPropertyRefMetadata(cache) {
    if (cache.value === null) {
        cache.value = {
            interfaces: [cache.implementedInterface()],

            baseClass: null,

            functions: {},

            properties: {},

            types: {},

            staticMembers: {}

        };  

    }
    return cache.value;
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.  
 * Use of this source code is governed by the Apache 2.0 license that can be found in the  
 * license/LICENSE.txt file.  
 */

Kotlin.toShort = function (a) {
    return (a & 0xFFFF) << 16 >> 16;
};

Kotlin.toByte = function (a) {
    return (a & 0xFF) << 24 >> 24;
};

Kotlin.toChar = function (a) {
    return a & 0xFFFF;
};

Kotlin.numberToLong = function (a) {
    return a instanceof Kotlin.Long ? a : Kotlin.Long.fromNumber(a);
};

Kotlin.numberToInt = function (a) {
    return a instanceof Kotlin.Long ? a.toInt() : Kotlin.doubleToInt(a);
};

Kotlin.numberToShort = function (a) {
    return Kotlin.toShort(Kotlin.numberToInt(a));
};

Kotlin.numberToByte = function (a) {
    return Kotlin.toByte(Kotlin.numberToInt(a));
};

Kotlin.numberToDouble = function (a) {
    return +a;
};

Kotlin.numberToChar = function (a) {
    return Kotlin.toChar(Kotlin.numberToInt(a));
};

Kotlin.doubleToInt = function (a) {
    if (a > 2147483647) return 2147483647;
    if (a < -2147483648) return -2147483648;
    return a | 0;
};

Kotlin.toBoxedChar = function (a) {
    if (a == null) return a;
    if (a instanceof Kotlin.BoxedChar) return a;
    return new Kotlin.BoxedChar(a);
};

Kotlin.unboxChar = function (a) {
    if (a == null) return a;
    return Kotlin.toChar(a);
};

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.  
 * Use of this source code is governed by the Apache 2.0 license that can be found in the  
 * license/LICENSE.txt file.  
 */

Kotlin.equals = function (obj1, obj2) {
    if (obj1 == null) {
        return obj2 == null;
    }

    if (obj2 == null) {
        return false;
    }

    if (obj1 !== obj1) {
        return obj2 !== obj2;
    }

    if (typeof obj1 === "object" && typeof obj1.equals === "function") {
        return obj1.equals(obj2);
    }

    if (typeof obj1 === "number" && typeof obj2 === "number") {
        return obj1 === obj2 && (obj1 !== 0 || 1 / obj1 === 1 / obj2)
    }

    return obj1 === obj2;
};

Kotlin.hashCode = function (obj) {
    if (obj == null) {
        return 0;
    }

    var objType = typeof obj;

    if ("object" === objType) {
        return "function" === objType ? obj.hashCode() : getObjectHashCode(obj);
    }

    if ("number" === objType) {
        return Kotlin.numberHashCode(obj);
    }

    if ("boolean" === objType) {
        return Number(obj);
    }

    var str = String(obj);
    return getHashCode(obj, str);
};

Kotlin.identityHashCode = getObjectHashCode;

/** @const */
var POW_2_32 = 4294967296;
// TODO: consider switching to Symbol type once we are on ES6.
/** @const */
var OBJECT_HASH_CODE_PROPERTY_NAME = "kotlinHashCodeValue$";

function getObjectHashCode(obj) {
    if (!OBJECT_HASH_CODE_PROPERTY_NAME in obj)) {
        var hash = (Math.random() * POW_2_32) | 0;

        Object.defineProperty(obj, OBJECT_HASH_CODE_PROPERTY_NAME, {
            value: hash, enumerable: false });

        return obj;  

    } /*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.  
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.  
 * You may not use this file except in compliance with the License.  
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 */

Open Source Used In DNAC Platform Antelope 13612
software

// distributed under the License is distributed on an "AS-IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

/**
 * Constructs a 64-bit two's-complement integer,
given its low and high 32-bit values as *signed* integers. See the from* functions below for more
* convenient ways of constructing Longs.

The internal representation of a long is the two given signed, 32-bit values.
We use 32-bit pieces because these are the size of integers on which Javascript performs bit-
operations. For operations like addition and multiplication, we split each number into 16-bit pieces, which can
easily be multiplied within Javascript's floating-point representation without overflow
* or change in sign.

In the algorithms below, we frequently reduce the negative case to the positive case by negating the
input(s) and then post-processing the result.

Note that we must ALWAYS check specially whether those values are MIN_VALUE
*(-2^63) because -MIN_VALUE == MIN_VALUE (since 2^63 cannot be represented as a positive number, it overflows back into a negative). Not handling this case would often result in infinite recursion.

@constructor
Kotlin.Long = function(low, high) {
  this.low_ = low | 0;  // force into 32 signed bits.
  this.high_ = high | 0;  // force into 32 signed bits.
};

Kotlin.Long.$metadata$ = {
    kind: "class",
    simpleName: "Long",
    interfaces:

// NOTE: Common constant values
ZERO, ONE, NEG_ONE, etc. are defined below the from* methods on which they depend.

/**
 * Returns a Long representing the given (32-bit) integer value.
 * @param {number} value The 32-bit integer in question.
 * @return {!Kotlin.Long} The corresponding Long value.
 */
Kotlin.Long.fromInt = function(value) {
  if (-128 <= value && value < 128) {
    var cachedObj = Kotlin.Long.IntCache_[value];
    if (cachedObj) {
      return cachedObj;
    }
  }

  var obj = new Kotlin.Long(value | 0, value < 0 ? -1 : 0);
  if (-128 <= value && value < 128) {
  }
  return obj;
};

/**
 * Returns a Long representation of the given string, written using the given
 * radix.
 * @param {string} str The textual representation of the Long.
 * @param {number=} opt_radix The radix in which the text is written.
 * @return {!Kotlin.Long} The corresponding Long value.
 */
Kotlin.Long.fromString = function(str, opt_radix) {
  if (str.length == 0) {
    throw Error('number format error: empty string');
  }

  var radix = opt_radix || 10;
  if (radix < 2 || 36 < radix) {
    throw Error('radix out of range: ' + radix);
  }

  if (str.charAt(0) == '-') {
    return Kotlin.Long.fromString(str.substring(1), radix).negate();
  } else if (str.indexOf('-') >= 0) {
    throw Error('number format error: interior "-" character: ' + str);
  }

  // Do several (8) digits each time through the loop, so as to
  // minimize the calls to the very expensive emulated div.
  var radixToPower = Kotlin.Long.fromNumber(Math.pow(radix, 8));
  var result = Kotlin.Long.ZERO;
  var power = Kotlin.Long.fromNumber(Math.pow(radix, size));
  for (var i = 0; i < str.length; i += 8) {
    var size = Math.min(8, str.length - i);
    var parsed = parseInt(str.substring(i, i + size), radix);
    if (size < 8) {
      var power = Kotlin.Long.fromNumber(Math.pow(radix, size));
    } else {
      result = result.multiply(power).add(Kotlin.Long.fromNumber(value));
    }
  }
  return result;
};
```javascript
result.multiply(radixToPower);
result = result.add(Kotlin.Long.fromNumber(value));
}


// NOTE: the compiler should inline these constant values below and then remove/" delete these variables, so there should be no runtime penalty for these. Number used repeated below in calculations.

/**
 * Number used repeated below in calculations.
 * This must appear before the first call to any from function below.
 * @type {number}
 * @private
 */
Kotlin.Long.TWO_PWR_16_DBL_ = 1 << 16;
Kotlin.Long.TWO_PWR_24_DBL_ = 1 << 24;
Kotlin.Long.TWO_PWR_32_DBL_ = 1 << 32;
Kotlin.Long.TWO_PWR_48_DBL_ = 1 << 48;
Kotlin.Long.TWO_PWR_64_DBL_ = 1 << 64;

/**
 * @type {number}
 * @private
 */
Kotlin.Long.TWO_PWR_24_DBL_ = Kotlin.Long.TWO_PWR_16_DBL_ * 2;
Kotlin.Long.TWO_PWR_32_DBL_ = Kotlin.Long.TWO_PWR_16_DBL_ * 2;
Kotlin.Long.TWO_PWR_64_DBL_ = Kotlin.Long.TWO_PWR_32_DBL_ * 2;

/**
 * @type {number}
 * @private
 */
Kotlin.Long.ZERO = Kotlin.Long.ZERO;
Kotlin.Long.ONE = Kotlin.Long.ONE;
Kotlin.Long.NEG_ONE = Kotlin.Long.NEG_ONE;
Kotlin.Long.MAX_VALUE = Kotlin.Long.MAX_VALUE;
Kotlin.Long.MIN_VALUE = Kotlin.Long.MIN_VALUE;
Kotlin.Long.TWO_PWR_24_ = Kotlin.Long.TWO_PWR_24_DBL_;
: bit + 1;
}]);

/** @return {boolean} Whether this value is zero. */
Kotlin.Long.prototype.isZero = function() {
  return this.high_ == 0 &
  this低_ == 0;
}

/** @return {boolean} Whether this value is negative. */
Kotlin.Long.prototype.isNegative = function() {
  return this.high_ == 0;
}

/** @return {boolean} Whether this value is odd. */
Kotlin.Long.prototype.isOdd = function() {
  return (this.low_ & 1) == 1;
}

/** @return {boolean} Whether this Long does not equal the other. */
Kotlin.Long.prototype.notEqualsLong = function(other) {
  return (this.high_ != other.high_)
  && (this.low_ != other.low_);
}

/** @return {boolean} Whether this Long is less than the other. */
Kotlin.Long.prototype.lessThan = function(other) {
  return this.compare(other) < 0;
}

/** @return {boolean} Whether this Long is less than or equal to the other. */
Kotlin.Long.prototype.lessThanOrEqual = function(other) {
  return this.compare(other) <= 0;
}

/** @return {boolean} Whether this Long is greater than the other. */
Kotlin.Long.prototype.greaterThan = function(other) {
  return this.compare(other) > 0;
}

/** @return {boolean} Whether this Long is greater than or equal to the other. */
Kotlin.Long.prototype.greaterThanOrEqual = function(other) {
  return this.compare(other) >= 0;
}

/** @return {Kotlin.Long} The negation of this value. */
Kotlin.Long.prototype.negate = function() {
  if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    return Kotlin.Long.MIN_VALUE;
  } else {
    return this.not().add(Kotlin.Long.ONE);
  }
}

/** @return {Kotlin.Long} The sum of this and the given Long. */
Kotlin.Long.prototype.add = function(other) {
  // Divide each number into 4 chunks of 16 bits, and
  // then sum the chunks.
  var a48 = this.high_ >>> 16;
  var a32 = this.high_ & 0xFFFF;
  var a16 = this.low_ >>> 16;
  var a00 = this.low_ & 0xFFFF;

  var b48 = other.high_ >>> 16;
  var b32 = other.high_ & 0xFFFF;
  var b16 = other.low_ >>> 16;
  var b00 = other.low_ & 0xFFFF;

  var c48 = 0, c32 = 0, c16 = 0, c00 = 0;
  var c00 += a00 + b00;
  var c16 += c00 >>> 16;
  var c00 &= 0xFFFF;
  var c16 += a16 + b16;
  var c32 += c16 >>> 16;
  var c16 &= 0xFFFF;
  var c32 += a32 + b32;
  var c48 += c32 >>> 16;
  var c32 &= 0xFFFF;
  var c48 += a48 + b48;
  var c48 &= 0xFFFF;

  return Kotlin.Long.fromBits(c48 << 16, c00, c48 << 16, c32);
}

/** @return {Kotlin.Long} The product of this and the given Long. */
Kotlin.Long.prototype.multiply = function(other) {
  if (this.isZero()) {
    return Kotlin.Long.ZERO;
  } else if (other.isZero()) {
    return Kotlin.Long.ZERO;
  }

  // If both longs are small, use float
  // multiplication
  if (this.lessThan(Kotlin.Long.TWO_PWR_24_) &&
  other.lessThan(Kotlin.Long.TWO_PWR_24_)) {
    return Kotlin.Long.fromNumber(this.toNumber() *

Open Source Used In DNAC Platform Antelope 13615
other.toString());

  // Divide each long into 4 chunks of 16 bits, and then add up 4x4 products.
  // We can skip products that would overflow.

  var a48 = this.high_ >>> 16;
  var a32 = this.high_ & 0xFFFF;
  var a16 = this.low_ >>> 16;
  var a00 = this.low_ & 0xFFFF;

  var b48 = other.high_ >>> 16;
  var b32 = other.high_ & 0xFFFF;
  var b16 = other.low_ >>> 16;
  var b00 = other.low_ & 0xFFFF;

  var c48 = 0, c32 = 0, c16 = 0, c00 = 0;
  c00 += a00 * b00;
  c16 += a16 * b00;
  c32 += a32 * b00;
  c48 += a48 * b00;

  c00 += a00 * b16;
  c16 += a16 * b16;
  c32 += a32 * b16;
  c48 += a48 * b16;

  c00 += a00 * b32;
  c16 += a16 * b32;
  c32 += a32 * b32;
  c48 += a48 * b32;

  c00 += a00 * b48;
  c16 += a16 * b48;
  c32 += a32 * b48;
  c48 += a48 * b48;

  return Kotlin.Long.fromBits(c16 << 16) | c00, (c48 << 16) | c32);
}

/**
 * Returns this Long divided by the given one.
 * @param {Kotlin.Long} other Long by which to divide.
 * @return {!Kotlin.Long} This Long divided by the given one.
 */
Kotlin.Long.prototype.div = function(other) {
  if (other.isZero()) {
    throw Error('division by zero');
  } else if (this.isZero()) {
    return Kotlin.Long.ZERO;
  }

  if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {
    if (other.equalsLong(Kotlin.Long.ONE) ||
        other.equalsLong(Kotlin.Long.NEG_ONE)) {
      return Kotlin.Long.MIN_VALUE;  // recall that -MIN_VALUE
    } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {
      return Kotlin.Long.ONE;
    } else {
      // At this point, we have |other| >= 2, so |this| < |MIN_VALUE|.
      var halfThis = this.shiftRight(1);
      var approx = halfThis.div(other).shiftLeft(1);
      if ((approx.equalsLong(Kotlin.Long.ZERO)) ||
          (other.isNegative() &&
           (other.equalsLong(Kotlin.Long.MIN_VALUE) ||
            other.equalsLong(Kotlin.Long.NEG_ONE)))) {
        return Kotlin.Long.MIN_VALUE;  // recall that -MIN_VALUE
      } else if (other.equalsLong(Kotlin.Long.ZERO)) {
        return Kotlin.Long.ZERO;
      }

      if (this.isNegative()) {
        if (other.isNegative()) {
          return this.negate().div(other.negate());
        } else {
          return this.negate().div(other).negate();
        }
      } else if (other.isNegative()) {
        return this.div(other.negate());
      }

      // Repeat the following until the remainder is less than or equal to the
      // approximate result of division.  This may be a little greater or
      // smaller than the actual value.
      var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
      // We will tweak the approximate result by changing it in the 48-th digit or
      // the smallest non-fractional digit, whichever is larger.

      var log2 = Math.ceil(Math.log(approx) / Math.LN2);
      var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

      var approxRem = approxRem.multiply(other);
      while (approxRem.isNegative() || approxRem.greaterThan(rem)) {
        approxRem = approxRem.subtract(other.multiply(approxRem));
      }

      var result = approxRem.add(other.multiply(approxRem));
      var res = res.add(approxRes);
      return result;
    }
  } else if (this.equalsLong(Kotlin.Long.ZERO)) {
    return Kotlin.Long.ZERO;
  }

  if (rem.randn()) {
    // Approximate the result of division.  This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.

    var log2 = Math.ceil(Math.log(approx) / Math.LN2);
    var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

    // Decrease the approximation until it is smaller than the remainder.
    while (approxRem.randn()) {
      approxRem = approxRem.multiply(other);
    }

    var result = approxRem.add(other.multiply(approxRem));
    var res = res.add(approxRes);
    return result;
  } else if (rem.randn()) {
    // Approximate the result of division.  This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.

    var log2 = Math.ceil(Math.log(approx) / Math.LN2);
    var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

    // Decrease the approximation until it is smaller than the remainder.
    while (approxRem.randn()) {
      approxRem = approxRem.multiply(other);
    }

    var result = approxRem.add(other.multiply(approxRem));
    var res = res.add(approxRes);
    return result;
  } else if (rem.randn()) {
    // Approximate the result of division.  This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.

    var log2 = Math.ceil(Math.log(approx) / Math.LN2);
    var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

    // Decrease the approximation until it is smaller than the remainder.
    while (approxRem.randn()) {
      approxRem = approxRem.multiply(other);
    }

    var result = approxRem.add(other.multiply(approxRem));
    var res = res.add(approxRes);
    return result;
  } else if (rem.randn()) {
    // Approximate the result of division.  This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.

    var log2 = Math.ceil(Math.log(approx) / Math.LN2);
    var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

    // Decrease the approximation until it is smaller than the remainder.
    while (approxRem.randn()) {
      approxRem = approxRem.multiply(other);
    }

    var result = approxRem.add(other.multiply(approxRem));
    var res = res.add(approxRes);
    return result;
  } else if (rem.randn()) {
    // Approximate the result of division.  This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.

    var log2 = Math.ceil(Math.log(approx) / Math.LN2);
    var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

    // Decrease the approximation until it is smaller than the remainder.
    while (approxRem.randn()) {
      approxRem = approxRem.multiply(other);
    }

    var result = approxRem.add(other.multiply(approxRem));
    var res = res.add(approxRes);
    return result;
  } else if (rem.randn()) {
    // Approximate the result of division.  This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.

    var log2 = Math.ceil(Math.log(approx) / Math.LN2);
    var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

    // Decrease the approximation until it is smaller than the remainder.
    while (approxRem.randn()) {
      approxRem = approxRem.multiply(other);
    }

    var result = approxRem.add(other.multiply(approxRem));
    var res = res.add(approxRes);
    return result;
  } else if (rem.randn()) {
    // Approximate the result of division.  This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.

    var log2 = Math.ceil(Math.log(approx) / Math.LN2);
    var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

    // Decrease the approximation until it is smaller than the remainder.
    while (approxRem.randn()) {
      approxRem = approxRem.multiply(other);
    }

    var result = approxRem.add(other.multiply(approxRem));
    var res = res.add(approxRes);
    return result;
  } else if (rem.randn()) {
    // Approximate the result of division.  This may be a little greater or
    // smaller than the actual value.
    var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));
    // We will tweak the approximate result by changing it in the 48-th digit or
    // the smallest non-fractional digit, whichever is larger.

    var log2 = Math.ceil(Math.log(approx) / Math.LN2);
    var delta = (log2 <= 48) ? 1 : Math.pow(2, log2 - 48);

    // Decrease the approximation until it is smaller than the remainder.
    while (approxRem.randn()) {
      approxRem = approxRem.multiply(other);
    }

    var result = approxRem.add(other.multiply(approxRem));
    var res = res.add(approxRes);
    return result;
  }
other.low_\n this.high_ | other.high_);\n returns the bitwise-XOR of this Long and the given one.\n * @param {Kotlin.Long} other The Long with which to XOR.\n * @return {Kotlin.Long} The bitwise-XOR of this and the other.\n *\n Kotlin.Long.prototype.xor = function(other) {\n \n return Kotlin.Long.fromBits(this.low_^ other.low_,\n this.high_^ other.high_);\n \n \n /*\n * Returns the bitwise-XOR of this Long and the given amount.\n * @param {number} numBits The number of bits by which to shift.\n * @return {Kotlin.Long} This shifted to the left by the given amount.\n */\n Kotlin.Long.prototype.shiftLeft = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n return this;\n } else {\n var low = this.low_;\n if (numBits < 32) {\n var high = this.high_;\n return Kotlin.Long.fromBits(\n low << numBits,\n high << numBits);\n } else {\n return Kotlin.Long.fromBits(\n 0, low << (numBits - 32));\n }\n }\n \n Kotlin.Long.prototype.shiftRight = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n return this;\n } else {\n var low = this.low_;\n var high = this.high_;\n if (numBits < 32) {\n return Kotlin.Long.fromBits(\n low >> numBits,\n (high << (32 - numBits)) | (low >>> (32 - numBits)));\n } else {\n return Kotlin.Long.fromBits(\n high >>> numBits);\n }\n }\n \n Kotlin.Long.prototype.shiftRightUnsigned = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n return this;\n } else {\n var low = this.low_;\n var high = this.high_;\n if (numBits < 32) {\n var low = this.low_;\n return Kotlin.Long.fromBits(\n low << numBits,\n (high >>> numBits) | (low >>> (32 - numBits)));\n } else {\n return Kotlin.Long.fromBits(\n high >>> numBits);\n }\n }\n \n /*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be\n * found in the license/LICENSE.txt file.\n */\n \n Kotlin.defineModule = function(id, declaration) {\n \n \n Kotlin.defineInlineFunction = function(tag, fun) {\n return fun;\n }\n \n Kotlin.wrapFunction = function(fun) {\n \n f = (function() {\n return f.apply(this, arguments);\n });\n return f;\n }\n \n Kotlin.orNull = function(fn) {\n return function(object) {\n return object == null ? fn(object) : fn(object);\n }\n }\n \n Kotlin.isTypeOf = function(type) {\n return function(object) {\n return typeof object === type;\n }\n }\n \n Kotlin.isInstanceOf = function(klass) {\n return function(object) {\n return object instanceof klass;\n }\n }\n \n Kotlin.orNull = function(fn) {\n return function(object) {\n return object == null ? fn(object) : fn(object);\n }\n }\n \n Kotlin.andPredicate = function(a, b) {\n return function(object) {\n return a(object) \n & b(object);\n }\n }\n \n /*\n * This marker function
should never been called. \"Looks like compiler did not eliminate it properly.\" \n\nPlease, report an issue if you caught this exception.\" Ön\n
Kotlin.getFunctionById = function(id, defaultValue) {\n    return defaultValue;\n};

Kotlin.getFunctionById = function(id, defaultValue) {\n    return defaultValue;\n};

Kotlin.compareTo = function (a, b) {\n    var typeA = typeof a;\n    var typeB = typeof b;\n    if (typeA === \"number\") {\n        if (typeB === \"number\") {\n            return Kotlin.doubleCompareTo(a, b);\n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n        \n
return a != null && a.__proto__ !== null && a.__proto__.__proto__ === Int8Array.prototype.__proto__;\n};\n\nObject.defineProperty(Array.prototype, 'fill', {\n    \n    value: function (value) {\n        Object.defineProperty(this, 'fill', {\n            value: function (value) {\n                Kaepernick الحقوقه rights of the African-American people.

                Kaepernick is known for his social justice activism, particularly his "I Can't Breathe" protest against systemic racism and police brutality. In 2016, he inspired a national conversation about race and police brutality after he was seen kneeling during the national anthem at an NFL game, a gesture that was interpreted as a protest against police violence.

            Kaepernick's activism has been met with both support and criticism. He has received numerous awards and honors for his contributions to social justice, but has also faced backlash from some who view his actions as unpatriotic or disrespectful.

            Despite the controversy, Kaepernick continues to be a symbol of resistance and a voice for the marginalized. His impact extends far beyond the football field, as he has inspired others to stand up for what they believe in and to use their platform to effect change.

            Kaepernick's story is a reminder of the power of individual action and the importance of standing up for what is right, even in the face of opposition.

            Kaepernick's second chapter, "Casting my Vote," continues his exploration of identity and purpose in the face of adversity.

            Kaepernick's second book, "Casting my Vote," is expected to be released in early 2021.

            Kaepernick's passion for social justice and his commitment to the cause have earned him worldwide recognition and admiration. His message of unity and resistance resonates with millions of people around the world.

            Kaepernick remains an inspiration to those who seek to make a positive impact in the world and to those who are fighting for a better tomorrow.

            Kaepernick’s journey is a testament to the power of resilience and the importance of standing up for what one believes in. His story continues to inspire and challenge, and he remains a defiant voice for social justice and equality.

            Kaepernick’s next book, "Casting my Vote," will be released in early 2021. The book will explore his identity, purpose, and motivation in the face of adversity.

            Kaepernick has an unwavering commitment to using his platform to effect change and to fight for social justice. He continues to be an inspiration to millions of people around the world, and his story will no doubt continue to resonate for years to come.

            Kaepernick’s story is a reminder of the power of individual action and the importance of standing up for what is right, even in the face of opposition. His impact extends far beyond the football field, as he has inspired others to stand up for what they believe in and to use their platform to effect change.

            Kaepernick's story is a testament to the power of resilience and the importance of standing up for what one believes in. His journey continues to inspire and challenge, and he remains a defiant voice for social justice and equality.

            Kaepernick’s next book, "Casting my Vote," will be released in early 2021. The book will explore his identity, purpose, and motivation in the face of adversity.

            Kaepernick has an unwavering commitment to using his platform to effect change and to fight for social justice. He continues to be an inspiration to millions of people around the world, and his story will no doubt continue to resonate for years to come.
Open Source Used In DNAC Platform Antelope 13621

thisObject[propertyName] = value; \nreturn \n
Kotlin.callSetter(thisObject, Object.getPrototypeOf(klass), propertyName, value); \n\n} \n\n\n\nif (ctor === iface) return true; \nvar metadata = ctor.$metadata$; \nif (metadata !== null) { \nvar interfaces = metadata.interfaces; \nfor (var i = 0; i < interfaces.length; i++) { \n    if (isInheritanceFromInterface(interfaces[i], iface)) { \n        return true; \n    } \n} \nreturn superConstructor !== null && isInheritanceFromInterface(superConstructor, iface); \n}\n\n\nfunction isInheritanceFromInterface(ctor, iface) {
    if (ctor === iface) return true;

    var metadata = ctor.$metadata$;
    if (metadata != null) {
        var interfaces = metadata.interfaces;
        for (var i = 0; i < interfaces.length; i++) {
            if (isInheritanceFromInterface(interfaces[i], iface)) {
                return true;
            }
        }
    }

    var superPrototype = ctor.prototype != null ? Object.getPrototypeOf(ctor.prototype) : null;
    var superConstructor = superPrototype.constructor != null ? superPrototype.constructor : null;
    return superConstructor !== null && isInheritanceFromInterface(superConstructor, iface);
}

/**
 * @param {*} object
 * @param {Function|Object} klass
 * @returns {Boolean}
 */
Kotlin.isType = function (object, klass) {
    if (klass === Object) {
        switch (typeof object) {
            case "string":
            case "number":
            case "boolean":
            case "function":
                return true;
            default:
                return object instanceof Object;
        }
    }

    if (object == null || klass == null || (typeof object !== 'object' && typeof object !== 'function')) {
        return false;
    }

    if (typeof klass === 'function' && object instanceof klass) {
        return true;
    }

    var proto = Object.getPrototypeOf(klass);
    var constructor = proto != null ? proto.constructor : null;
    if (constructor != null && "$metadata$" in constructor) {
        var metadata = constructor.$metadata$;
        if (metadata.kind === Kotlin.Kind.OBJECT) {
            return object === klass;
        }
    }

    var klassMetadata = klass.$metadata$;

    // In WebKit (JavaScriptCore) for some interfaces from DOM typeof returns 'object', nevertheless they can be used in RHS of instanceof
    if (klassMetadata == null) {
        return object instanceof klass;
    }

    if (klassMetadata.kind === Kotlin.Kind.INTERFACE && object.constructor != null) {
        return isInheritanceFromInterface(object.constructor, klass);
    }

    return false;
};

Kotlin.isNumber = function (a) {
    return typeof a == "number" || a instanceof Kotlin.Long;
};

Kotlin.isChar = function (value) {
    return value instanceof Kotlin.BoxedChar
};

Kotlin.isComparable = function (value) {
    var type = typeof value;

    return type === "string" ||
           type === "boolean" ||
           Kotlin.isNumber(value) ||
           Kotlin.isType(value, Kotlin.kotlin.Comparable);
};

Kotlin.isCharSequence = function (value) {
    return typeof value === "string" ||
           Kotlin.isType(value, Kotlin.kotlin.CharSequence);
};
result: dynamic = Array<Long>(size) \n result.'$type$' = "LongArray" \n return when (init) { \n null, true -> \n fillArrayVal(result, 0L) \n false -> result else -> fillArrayFun<Long>(result, init)\n }\n@JsName("longArrayF") inline fun longArrayWithFun(size: Int, init: (Int) -> Long): Array<Long> = \n fillArrayFun(longArray(size, false), init)\n
private fun <T> fillArrayVal(array: Array<T>, initValue: T): Array<T> { \n for (i in 0.array.size - 1) { \n array[i] = initValue } \n return array }
`@kotlin.internal.InlineOnly`\n`public inline operator fun ByteArray.component2(): Byte` \n`{ return get(1) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun ShortArray.component2(): Short` \n`{ return get(1) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun IntArray.component2(): Int` \n`{ return get(1) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun LongArray.component2(): Long` \n`{ return get(1) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun FloatArray.component2(): Float` \n`{ return get(1) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun DoubleArray.component2(): Double` \n`{ return get(1) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun BooleanArray.component2(): Boolean` \n`{ return get(1) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun CharArray.component2(): Char` \n`{ return get(1) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun Array<out T>.component3(): T` \n`{ return get(2) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun ByteArray.component3(): Byte` \n`{ return get(2) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun ShortArray.component3(): Short` \n`{ return get(2) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun IntArray.component3(): Int` \n`{ return get(2) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun LongArray.component3(): Long` \n`{ return get(2) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun FloatArray.component3(): Float` \n`{ return get(2) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun DoubleArray.component3(): Double` \n`{ return get(2) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun BooleanArray.component3(): Boolean` \n`{ return get(2) }` \n`@kotlin.internal.InlineOnly`\n`public inline operator fun CharArray.component3(): Char` \n`{ return get(2) }`
[IndexOutOfBoundsException] except in Kotlin/JS

*@kotlin.internal.InlineOnly\public inline operator fun CharArray.component3(): Char {
    return get(2)
}*

*@kotlin.internal.InlineOnly\public inline operator fun <T> Array\<out T\>.component4(): T {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun ByteArray.component4(): Byte {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun ShortArray.component4(): Short {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun IntArray.component4(): Int {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun LongArray.component4(): Long {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun FloatArray.component4(): Float {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun DoubleArray.component4(): Double {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun BooleanArray.component4(): Boolean {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun CharArray.component4(): Char {
    return get(3)
}*

*@kotlin.internal.InlineOnly\public inline operator fun <T> Array\<out T\>.component5(): T {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun ByteArray.component5(): Byte {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun ShortArray.component5(): Short {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun IntArray.component5(): Int {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun LongArray.component5(): Long {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun FloatArray.component5(): Float {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun DoubleArray.component5(): Double {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun BooleanArray.component5(): Boolean {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun CharArray.component5(): Char {
    return get(4)
}*

*@kotlin.internal.InlineOnly\public inline operator fun <T> Array\<out T\>.component6(): T {
    return get(5)
}*

*@kotlin.internal.InlineOnly\public inline operator fun ByteArray.component6(): Byte {
    return get(5)
}*

*@kotlin.internal.InlineOnly\public inline operator fun ShortArray.component6(): Short {
    return get(5)
}*

*@kotlin.internal.InlineOnly\public inline operator fun IntArray.component6(): Int {
    return get(5)
}*

*@kotlin.internal.InlineOnly\public inline operator fun LongArray.component6(): Long {
    return get(5)
}*

*@kotlin.internal.InlineOnly\public inline operator fun FloatArray.component6(): Float {
    return get(5)
}*

*@kotlin.internal.InlineOnly\public inline operator fun DoubleArray.component6(): Double {
    return get(5)
}*

*@kotlin.internal.InlineOnly\public inline operator fun BooleanArray.component6(): Boolean {
    return get(5)
}*

*@kotlin.internal.InlineOnly\public inline operator fun CharArray.component6(): Char {
    return get(5)
}*

*IndexOutOfBoundsException*
@kotlin.internal.InlineOnly
public inline operator fun DoubleArray.component5(): Double {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun BooleanArray.component5(): Boolean {
    return get(4)
}

/**
 * Returns 5th *element* from the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.component5(): Char {
    return get(4)
}

/**
 * Returns `true` if [element] is found in the array.
 */
public operator fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.contains(element: T): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns `true` if [element] is found in the array.
 * 
 * If the size of this array is less than 5, throws an
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@kotlin.internal.InlineOnly
public inline operator fun CharArray.elementAt(index: Int): Char {
    return get(index)
}
an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.

```kotlin
public expect fun BooleanArray.elementAt(index: Int): Boolean

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.

```
samples.collections.Collections.Elements.elementAtOrNull */

@kotlin.internal.InlineOnly
public inline fun ShortArray.elementAtOrNull(index: Int): Short? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample samples.collections.Collections.Elements.elementAtOrNull
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.elementAtOrNull(index: Int): Int? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample samples.collections.Collections.Elements.elementAtOrNull
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.elementAtOrNull(index: Int): Long? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample samples.collections.Collections.Elements.elementAtOrNull
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.elementAtOrNull(index: Int): Float? {
    return this.getOrNull(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample samples.collections.Collections.Elements.elementAtOrNull
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.elementAtOrNull(index: Int): Double? {
    return this.getOrNull(index)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.find(predicate: (T) -> Boolean): T? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.find(predicate: (Byte) -> Boolean): Byte? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.find(predicate: (Short) -> Boolean): Short? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.find(predicate: (Int) -> Boolean): Int? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.find(predicate: (Long) -> Boolean): Long? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.find(predicate: (Float) -> Boolean): Float? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.find(predicate: (Double) -> Boolean): Double? {
    return firstOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.find(predicate: (Boolean) -> Boolean): Boolean? {
    return firstOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.find(predicate: (Char) -> Boolean): Char? {
    return firstOrNull(predicate)
}
samples.collections.Collections.Elements.findLast(predicate: (T) -> Boolean): T? { \n    return lastOrNull(predicate)\n} \n\n/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.findLast(predicate: (T) -> Boolean): T? { \n    return lastOrNull(predicate)\n} \n\n/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.findLast(predicate: (Short) -> Boolean): Short? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.findLast(predicate: (Int) -> Boolean): Int? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.findLast(predicate: (Long) -> Boolean): Long? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.findLast(predicate: (Float) -> Boolean): Float? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.findLast(predicate: (Double) -> Boolean): Double? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or 'null' if no such element was found.\n * @sample
 * samples.collections.Collections.Elements.findLast
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.findLast(predicate: (Char) -> Boolean): Char? {
    return lastOrNull(predicate)
}

/**
 * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */
public inline fun <T> Array<out T>.first(predicate: (T) -> Boolean): T? { \n    for (element in this) if (predicate(element)) return
}
element
    throw NoSuchElementException("Array contains no element matching the predicate.")

Returns the first element matching the given [predicate].

@throws [NoSuchElementException] if no such element is found.

*/

//inline fun ByteArray.first(predicate: (Byte) -> Boolean): Byte {<br>
//    for (element in this) if (predicate(element)) return element
//    throw NoSuchElementException("Array contains no element matching the predicate.")
//}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun ByteArray.first(predicate: (Byte) -> Boolean): Byte {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun ShortArray.first(predicate: (Short) -> Boolean): Short {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun IntArray.first(predicate: (Int) -> Boolean): Int {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun LongArray.first(predicate: (Long) -> Boolean): Long {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun FloatArray.first(predicate: (Float) -> Boolean): Float {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun DoubleArray.first(predicate: (Double) -> Boolean): Double {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun BooleanArray.first(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun CharArray.first(predicate: (Char) -> Boolean): Char {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun <T> Array<out T>.firstOrNull(): T? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun ByteArray.firstOrNull(): Byte? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun ShortArray.firstOrNull(): Short? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun IntArray.firstOrNull(): Int? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun LongArray.firstOrNull(): Long? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun FloatArray.firstOrNull(): Float? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun DoubleArray.firstOrNull(): Double? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun BooleanArray.firstOrNull(): Boolean? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
public fun CharArray.firstOrNull(): Char? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
public inline fun ByteArray.firstOrNull(predicate: (Byte) -> Boolean): Byte? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
public inline fun ShortArray.firstOrNull(predicate: (Short) -> Boolean): Short? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
public inline fun IntArray.firstOrNull(predicate: (Int) -> Boolean): Int? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
public inline fun LongArray.firstOrNull(predicate: (Long) -> Boolean): Long? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
public inline fun FloatArray.firstOrNull(predicate: (Float) -> Boolean): Float? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
public inline fun DoubleArray.firstOrNull(predicate: (Double) -> Boolean): Double? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
public inline fun BooleanArray.firstOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if element was not found.
 */
public inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {
    for (element in this) if (predicate(element)) return element
    return null
}
Boolean): Short? {\n    for (element in this) if (predicate(element)) return element
    return null
}

*\nArray.firstOrNull(predicate: (Int) -> Boolean): Int? {\n    for (element in this) if (predicate(element)) return element
    return null
}

*\nArray.firstOrNull(predicate: (Long) -> Boolean): Long? {\n    for (element in this) if (predicate(element)) return element
    return null
}

*\nArray.firstOrNull(predicate: (Double) -> Boolean): Double? {\n    for (element in this) if (predicate(element)) return element
    return null
}

*\nArray.firstOrNull(predicate: (Char) -> Boolean): Char? {\n    for (element in this) if (predicate(element)) return element
    return null
}

*\nArray.getOrElse(index: Int, defaultValue: (Int) -> Int): Int {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nArray.getOrElse(index: Int, defaultValue: (Long) -> Long): Long {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nArray.getOrElse(index: Int, defaultValue: (Double) -> Double): Double {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nArray.getOrElse(index: Int, defaultValue: (Char) -> Char): Char {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nByteArray.getOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nByteArray.getOrElse(index: Int, defaultValue: (Long) -> Long): Long {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nByteArray.getOrElse(index: Int, defaultValue: (Double) -> Double): Double {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nByteArray.getOrElse(index: Int, defaultValue: (Char) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nByteArray.getOrElse(index: Int, defaultValue: (Boolean) -> Boolean): Boolean {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

*\nByteArray.getOrElse(index: Int, defaultValue: (Char) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

Open Source Used in DNAC Platform Antelope 13630
get(index) else null

* Returns an element at the given [index] or 'null' if the [index] is out of bounds of this array.

`ByteArray.getOrNull(index: Int): Byte?`  
`public fun ByteArray.getOrNull(index: Int): Byte?`  
`public fun ShortArray.getOrNull(index: Int): Short?`  
`public fun IntArray.getOrNull(index: Int): Int?`  
`public fun LongArray.getOrNull(index: Int): Long?`  
`public fun FloatArray.getOrNull(index: Int): Float?`  
`public fun DoubleArray.getOrNull(index: Int): Double?`  
`public fun BooleanArray.getOrNull(index: Int): Boolean?`  
`public fun CharArray.getOrNull(index: Int): Char?`  

getOrNull(index: Int): Byte?  
`@sample samples.collections.Collections.Elements.getOrNull
/**
 * Returns an element at the given [index] or 'null' if the [index] is out of bounds of this array.
 */
public fun ByteArray.getOrNull(index: Int): Byte? {
    return if (index >= 0 && index <= lastIndex) get(index) else null
}
FloatArray.indexOfLast(predicate: (Float) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
public inline fun DoubleArray.indexOfLast(predicate: (Double) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
public inline fun BooleanArray.indexOfLast(predicate: (Boolean) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

/**
 * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.
 */
public inline fun CharArray.indexOfLast(predicate: (Char) -> Boolean): Int {
    for (index in indices.reversed()) {
        if (predicate(this[index])) {
            return index
        }
    }
    return -1
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun <T> Array<out T>.last(): T {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun ByteArray.last(): Byte {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun ShortArray.last(): Short {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun IntArray.last(): Int {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun LongArray.last(): Long {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun FloatArray.last(): Float {
    if (isEmpty())
        throw NoSuchElementException("Array is empty.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
public fun DoubleArray.last(): Double {
    if (isEmpty())
        throw NoSuchElementException("Array contains no element matching the predicate.")
    return this[lastIndex]
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if no such element is found.
 */
public fun IntArray.last(predicate: (Int) -> Boolean): Int {
    for (index in indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if no such element is found.
 */
public fun ShortArray.last(predicate: (Short) -> Boolean): Short {
    for (index in indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if no such element is found.
 */
public fun IntArray.last(predicate: (Int) -> Boolean): Int {
    for (index in indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if no such element is found.
 */
public fun ShortArray.last(predicate: (Short) -> Boolean): Short {
    for (index in indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}
[NoSuchElementException] if no such element is found.

```kotlin
public inline fun LongArray.last(predicate: (Long) -> Boolean): Long {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun FloatArray.last(predicate: (Float) -> Boolean): Float {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun DoubleArray.last(predicate: (Double) -> Boolean): Double {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun BooleanArray.last(predicate: (Boolean) -> Boolean): Boolean {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
public inline fun CharArray.last(predicate: (Char) -> Boolean): Char {
    for (index in this.indices.reversed()) {
        val element = this[index]
        if (predicate(element)) return element
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns last index of [element], or -1 if the array does not contain [element].
 */
public fun <T> Array<out T>.lastIndexOf(element: T): Int {
    if (element == null) {
        for (index in indices.reversed()) {
            if (this[index] == null) {
                return index
            }
        }
    } else {
        for (index in indices.reversed()) {
            if (element == this[index]) {
                return index
            }
        }
    }
    return -1
}
```

Open Source Used In DNAC Platform Antelope 13634
package com.example

import java.util.Random

fun ByteArray.random(): Byte {
    return random(Random)
}

/**
 * Returns a random element from this array. 
 * 
 * @throws NoSuchElementException if this array is empty.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element, or `null` if the array is empty. 
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun <T> Array<out T>.lastOrNull(): T? {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found. 
 */
public fun ByteArray.lastOrNull(predicate: (Byte) -> Boolean): Byte? {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found. 
 */
public fun DoubleArray.lastOrNull(predicate: (Double) -> Boolean): Double? {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found. 
 */
public fun IntArray.lastOrNull(predicate: (Int) -> Boolean): Int? {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found. 
 */
public fun LongArray.lastOrNull(predicate: (Long) -> Boolean): Long? {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found. 
 */
public fun ShortArray.lastOrNull(predicate: (Short) -> Boolean): Short? {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found. 
 */
public fun BooleanArray.lastOrNull(): Boolean? {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found. 
 */
public fun CharArray.lastOrNull(): Char? {
    return if (isEmpty()) null else this[size - 1][index]
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found. 
 */
public fun BooleanArray.lastIndexOf(element: Char): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

/**
 * Returns last index of [element], or -1 if the array does not contain element. 
 */
public fun CharArray.lastIndexOfOf(element: Char): Int {
    for (index in indices.reversed()) {
        if (element == this[index])
            return index
    }
    return -1
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public inline fun Array<out T>.random(): T {
    return random(Random)
}

/**
 * Returns a random element from this array, or `null` if the array is empty. 
 */
public fun ByteArray.random(): Byte {
    return random(Random)
}

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public fun <T> Array<out T>.random(): T {
    return random(Random)
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public inline fun ByteArray.random(): Byte {
    return random(Random)
}

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random)
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public fun ByteArray.random(): Byte {
    return random(Random)
}

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random)
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public fun ByteArray.random(): Byte {
    return random(Random)
}

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random)
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public fun ByteArray.random(): Byte {
    return random(Random)
}

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random)
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public fun ByteArray.random(): Byte {
    return random(Random)
}

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random)
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public fun ByteArray.random(): Byte {
    return random(Random)
}

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random)
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public fun ByteArray.random(): Byte {
    return random(Random)
}

@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.random(): T {
    return random(Random)
}

@Throws(SinceKotlin("1.3")@kotlin.internal.InlineOnly
public fun ByteArray.random(): Byte {
this array.

*\n * \n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.

*\n * @throws NoSuchElementException if this array is empty.
is empty.\n    return get(random.nextInt(size))
\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.randomOrNull(): T? {\n    return randomOrNull(Random)\n    \n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.randomOrNull(): Byte? {\n    return randomOrNull(Random)\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.randomOrNull(): Short? {\n    return randomOrNull(Random)\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.randomOrNull(): Int? {\n    return randomOrNull(Random)\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.randomOrNull(): Long? {\n    return randomOrNull(Random)\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.randomOrNull(): Float? {\n    return randomOrNull(Random)\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.randomOrNull(): Double? {\n    return randomOrNull(Random)\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.randomOrNull(): Boolean? {\n    return randomOrNull(Random)\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.randomOrNull(): Char? {\n    return randomOrNull(Random)\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic fun <T> Array<out T>.randomOrNull(random: Random): T? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))
\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic fun ByteArray.randomOrNull(random: Random): Byte? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))
\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic fun ShortArray.randomOrNull(random: Random): Short? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))
\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic fun IntArray.randomOrNull(random: Random): Int? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))
\n* @SinceKotlin("1.4") @WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic fun LongArray.randomOrNull(random: Random): Long? {\n    if (isEmpty())\n        return null\n    return get(random.nextInt(size))
randomness, or `null` if this array is empty.

```kotlin
*An@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)*
public fun FloatArray.randomOrNull(random: Random): Float? {
    if (isEmpty()) return null
    return get(random.nextInt(size))
}

*An@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)*
public fun DoubleArray.randomOrNull(random: Random): Double? {
    if (isEmpty()) return null
    return get(random.nextInt(size))
}

*An@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)*
public fun BooleanArray.randomOrNull(random: Random): Boolean? {
    if (isEmpty()) return null
    return get(random.nextInt(size))
}

*An@SinceKotlin("1.4")@WasExperimental(ExperimentalStdlibApi::class)*
public fun CharArray.randomOrNull(random: Random): Char? {
    if (isEmpty()) return null
    return get(random.nextInt(size))
}
```

Returns a random element from this array using the specified source of randomness.

```kotlin
/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T: Array<out T>> T.randomOrNull(random: Random): T? {
    if (isEmpty()) return null
    return get(random.nextInt(size))
}
```

Returns a random element from this array using the specified source of randomness.

```kotlin
var single: T? = null
var found = false
for (element in this) {
    if (predicate(element)) {
        if (found) {
            throw IllegalArgumentException("Array has more than one element.")
        }
        single = element
        found = true
    }
}
```

If there is exactly one matching element, returns that element.

```kotlin
public fun <T> Array<out T>.single(): T {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.

```kotlin
public fun ByteArray.single(): Byte {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.

```kotlin
public fun ShortArray.single(): Short {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.

```kotlin
public fun IntArray.single(): Int {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.

```kotlin
public fun LongArray.single(): Long {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.

```kotlin
public fun FloatArray.single(): Float {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.

```kotlin
public fun DoubleArray.single(): Double {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.

```kotlin
public fun BooleanArray.single(): Boolean {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.

```kotlin
public fun CharArray.single(): Char {
    return when (size) {
        0 -> throw NoSuchElementException("Array is empty.")
        1 -> this[0]
        else -> throw IllegalArgumentException("Array has more than one element.")
    }
}
```

Returns the single element, or throws an exception if the array is empty or has more than one element.
Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.

*\public inline fun ByteArray.single(predicate: (Byte) -> Boolean): Byte {
    var single: Byte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        } else if (!found) throw NoSuchElementException("Array contains no matching element.")
    }
    return single
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun ShortArray.single(predicate: (Short) -> Boolean): Short {
    var single: Short? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        } else if (!found) throw NoSuchElementException("Array contains no matching element.")
    }
    return single
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun IntArray.single(predicate: (Int) -> Boolean): Int {
    var single: Int? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        } else if (!found) throw NoSuchElementException("Array contains no matching element.")
    }
    return single
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun LongArray.single(predicate: (Long) -> Boolean): Long {
    var single: Long? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        } else if (!found) throw NoSuchElementException("Array contains no matching element.")
    }
    return single
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun FloatArray.single(predicate: (Float) -> Boolean): Float {
    var single: Float? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        } else if (!found) throw NoSuchElementException("Array contains no matching element.")
    }
    return single
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
 */
public inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {
    var single: Double? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        } else if (!found) throw NoSuchElementException("Array contains no matching element.")
    }
    return single
}
```kotlin
public inline fun DoubleArray.singleOrNull(predicate: (Double) -> Boolean): Double? {
    var single: Double? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public inline fun FloatArray.singleOrNull(predicate: (Float) -> Boolean): Float? {
    var single: Float? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public inline fun LongArray.singleOrNull(predicate: (Long) -> Boolean): Long? {
    var single: Long? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public inline fun IntArray.singleOrNull(predicate: (Int) -> Boolean): Int? {
    var single: Int? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public inline fun ShortArray.singleOrNull(predicate: (Short) -> Boolean): Short? {
    var single: Short? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public inline fun ByteArray.singleOrNull(predicate: (Byte) -> Boolean): Byte? {
    var single: Byte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public inline fun <T> Array<out T>.singleOrNull(predicate: (T) -> Boolean): T? {
    return if (size == 1) this[0] else null
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public fun CharArray.singleOrNull(): Char? {
    return if (size == 1) this[0] else null
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public fun BooleanArray.singleOrNull(): Boolean? {
    return if (size == 1) this[0] else null
}

/**
 * Returns the single element matching the given 
 * predicate, or `null` if element was not found or more than one element was found.
 */
public fun DoubleArray.singleOrNull(): Double? {
    return if (size == 1) this[0] else null
}
```
[predicate], or `null` if element was not found or more than one element was found.

```kotlin
public inline fun BooleanArray.singleOrNull(predicate: (Boolean) -> Boolean): Boolean? {
    var single: Boolean? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        } else if (found) return single
    }
    return single
}
```

```kotlin
/**
 * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.
 */
public inline fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? {
    var single: Char? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        } else if (found) return single
    }
    return single
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun <T> Array<out T>.drop(n: Int): List<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ByteArray.drop(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ShortArray.drop(n: Int): List<Short> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun IntArray.drop(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun LongArray.drop(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun FloatArray.drop(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun DoubleArray.drop(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun BooleanArray.drop(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except first [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun CharArray.drop(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun <T> Array<out T>.dropLast(n: Int): List<T> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ByteArray.dropLast(n: Int): List<Byte> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun ShortArray.dropLast(n: Int): List<Short> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun IntArray.dropLast(n: Int): List<Int> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun LongArray.dropLast(n: Int): List<Long> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun FloatArray.dropLast(n: Int): List<Float> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun DoubleArray.dropLast(n: Int): List<Double> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun BooleanArray.dropLast(n: Int): List<Boolean> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```

```kotlin
/**
 * Returns a list containing all elements except last [n] elements.
 * 
 * @throws IllegalArgumentException if [n] is negative.
 * 
 * @sample samples.collections.Collections.Transformations.drop
 */
public fun CharArray.dropLast(n: Int): List<Char> {
    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))
}
```
n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun ByteArray.dropLast(n: Int): List<Byte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun ShortArray.dropLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun IntArray.dropLast(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun LongArray.dropLast(n: Int): List<Long> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun FloatArray.dropLast(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun DoubleArray.dropLast(n: Int): List<Double>
{\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun BooleanArray.dropLast(n: Int):
List<Boolean> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic fun CharArray.dropLast(n: Int): List<Char> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given
[predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T>
Array<out T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n for (index in lastIndex downTo 0) {\n
if
(!predicate(this[index])) {\n
return take(index + 1)\n
}\n }\n return emptyList()\n}\n\n/**\n * Returns
a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic inline fun ByteArray.dropLastWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n
return take(index + 1)\n
}\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic inline fun ShortArray.dropLastWhile(predicate:
(Short) -> Boolean): List<Short> {\n for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n
return take(index + 1)\n
}\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements
except last elements that satisfy the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.drop\n */\npublic inline fun IntArray.dropLastWhile(predicate:
(Int) -> Boolean): List<Int> {\n for (index in lastIndex downTo 0) {\n
if (!predicate(this[index])) {\n
return take(index + 1)\n
}\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements

Open Source Used In DNAC Platform Antelope 13642


except last elements that satisfy the given [predicate].

* @sample

variables.forEach { if (!predicate(it)) {
    list.add(it)
} else {
    yielding = true
    list.add(it)
} return list
}

// Drop last elements that satisfy the given [predicate].

for (index in lastIndex downTo 0) {
    if (!predicate(this[index])) {
        return take(index + 1)
    }
}

return emptyList()
*/

public inline fun LongArray.dropLastWhile(predicate: (Long) -> Boolean): List<Long> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun FloatArray.dropLastWhile(predicate: (Float) -> Boolean): List<Float> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun DoubleArray.dropLastWhile(predicate: (Double) -> Boolean): List<Double> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun BooleanArray.dropLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except last elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun CharArray.dropLastWhile(predicate: (Char) -> Boolean): List<Char> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun ByteArray.dropWhile(predicate: (Byte) -> Boolean): List<Byte> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun ShortArray.dropWhile(predicate: (Short) -> Boolean): List<Short> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun LongArray.dropWhile(predicate: (Long) -> Boolean): List<Long> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun DoubleArray.dropWhile(predicate: (Double) -> Boolean): List<Double> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun CharArray.dropWhile(predicate: (Char) -> Boolean): List<Char> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun ByteArray.dropWhile(predicate: (Byte) -> Boolean): List<Byte> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun ShortArray.dropWhile(predicate: (Short) -> Boolean): List<Short> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun LongArray.dropWhile(predicate: (Long) -> Boolean): List<Long> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun DoubleArray.dropWhile(predicate: (Double) -> Boolean): List<Double> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun CharArray.dropWhile(predicate: (Char) -> Boolean): List<Char> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun ByteArray.dropWhile(predicate: (Byte) -> Boolean): List<Byte> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun ShortArray.dropWhile(predicate: (Short) -> Boolean): List<Short> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun LongArray.dropWhile(predicate: (Long) -> Boolean): List<Long> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample

samples.collections.Collections.Transformations.drop
 */

public inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {
    for (index in this) {
        if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    }
    return list
}

/**
 * Returns a list containing all elements except first elements that satisfy the given [predicate].
 * 
 * @sample
Open Source Used In DNAC Platform Antelope 13645

```kotlin
/**
* Returns a list containing only elements matching the given [predicate].
* @param [predicate] function that takes the index of an element and the element itself
* and returns the result of predicate evaluation on the element.
* @sample samples.collections.Collections.Filtering.filterIndexedTo
*/
public inline fun IntArray.filterIndexedTo(destination: Collection<Int>, predicate: (index: Int, Int) -> Boolean): Collection<Int> {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
* Returns a list containing only elements matching the given [predicate].
* @param [predicate] function that takes the index of an element and the element itself
* and returns the result of predicate evaluation on the element.
* @sample samples.collections.Collections.Filtering.filterIndexedTo
*/
public inline fun FloatArray.filterIndexedTo(destination: Collection<Float>, predicate: (index: Int, Float) -> Boolean): Collection<Float> {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
* Returns a list containing only elements matching the given [predicate].
* @param [predicate] function that takes the index of an element and the element itself
* and returns the result of predicate evaluation on the element.
* @sample samples.collections.Collections.Filtering.filterIndexedTo
*/
public inline fun DoubleArray.filterIndexedTo(destination: Collection<Double>, predicate: (index: Int, Double) -> Boolean): Collection<Double> {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
* Returns a list containing only elements matching the given [predicate].
* @param [predicate] function that takes the index of an element and the element itself
* and returns the result of predicate evaluation on the element.
* @sample samples.collections.Collections.Filtering.filterIndexedTo
*/
public inline fun BooleanArray.filterIndexedTo(destination: Collection<Boolean>, predicate: (index: Int, Boolean) -> Boolean): Collection<Boolean> {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

/**
* Returns a list containing only elements matching the given [predicate].
* @param [predicate] function that takes the index of an element and the element itself
* and returns the result of predicate evaluation on the element.
* @sample samples.collections.Collections.Filtering.filterIndexedTo
*/
public inline fun CharArray.filterIndexedTo(destination: Collection<Char>, predicate: (index: Int, Char) -> Boolean): Collection<Char> {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    }
    return destination
}

// More inline functions for other types...
```
Int, Int) -> Boolean): C {\n    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    } \n    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Long>> LongArray.filterIndexedTo(destination: C, predicate: (index: Int, Long) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    } \n    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Float>> FloatArray.filterIndexedTo(destination: C, predicate: (index: Int, Float) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    } \n    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Double>> DoubleArray.filterIndexedTo(destination: C, predicate: (index: Int, Double) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    } \n    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterIndexedTo(destination: C, predicate: (index: Int, Boolean) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    } \n    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * @param [predicate] function that takes the index of an element and the element itself
 * and returns the result of predicate evaluation on the element.
 * @sample samples.collections.Collections.Filtering.filterIndexedTo
 */
public inline fun <C : MutableCollection<in Char>> CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.add(element)
    } \n    return destination
}

/**
 * Returns a list containing all elements that are instances of specified type parameter R.
 * @sample samples.collections.Collections.Filtering.filterIsInstance
 */
public inline fun <reified R> Array<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {
    return filterIsInstanceTo(ArrayList<R>())
}

/**
 * Appends all elements that are instances of specified type parameter R to the given [destination].
 * @sample samples.collections.Collections.Filtering.filterIsInstanceTo
 */
public inline fun <reified R, C : MutableCollection<in R>> Array<*>.filterIsInstanceTo(destination: C): C {
    for (element in this) if (element is R) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun <T> Array<T>.filterNot(predicate: (T) -> Boolean): List<T> {
    return filterNotTo(ArrayList<T>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun ByteArray.filterNot(predicate: (Byte) -> Boolean): List<Byte> {
    return filterNotTo(ArrayList<Byte>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun ShortArray.filterNot(predicate: (Short) -> Boolean): List<Short> {
    return filterNotTo(ArrayList<Short>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun IntArray.filterNot(predicate: (Int) -> Boolean): List<Int> {
    return filterNotTo(ArrayList<Int>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun LongArray.filterNot(predicate: (Long) -> Boolean): List<Long> {
    return filterNotTo(ArrayList<Long>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun FloatArray.filterNot(predicate: (Float) -> Boolean): List<Float> {
    return filterNotTo(ArrayList<Float>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun DoubleArray.filterNot(predicate: (Double) -> Boolean): List<Double> {
    return filterNotTo(ArrayList<Double>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun BooleanArray.filterNot(predicate: (Boolean) -> Boolean): List<Boolean> {
    return filterNotTo(ArrayList<Boolean>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun CharArray.filterNot(predicate: (Char) -> Boolean): List<Char> {
    return filterNotTo(ArrayList<Char>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun IntArray.filterNot(predicate: (Int) -> Boolean): List<Int> {
    return filterNotTo(ArrayList<Int>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
 */
public inline fun LongArray.filterNot(predicate: (Long) -> Boolean): List<Long> {
    return filterNotTo(ArrayList<Long>(), predicate)
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 * @sample samples.collections.Collections.Filtering.filter
samples.collections.Collections.Filtering.filter

*//\public inline fun LongArray.filterNot(predicate: (Long) -> Boolean): List<Long> {
    return filterNotTo(ArrayList<Long>(), predicate)\n}\n/**
 * Returns a list containing all elements not matching the given \[predicate\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun LongArray.filterNot(predicate: (Long) -> Boolean): List<Long> {
    return filterNotTo(ArrayList<Long>(), predicate)\n}\n
/**
 * Returns a list containing all elements not matching the given \[predicate\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun FloatArray.filterNot(predicate: (Float) -> Boolean): List<Float> {
    return filterNotTo(ArrayList<Float>(), predicate)\n}\n
/**
 * Returns a list containing all elements not matching the given \[predicate\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun DoubleArray.filterNot(predicate: (Double) -> Boolean): List<Double> {
    return filterNotTo(ArrayList<Double>(), predicate)\n}\n
/**
 * Returns a list containing all elements not matching the given \[predicate\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun BooleanArray.filterNot(predicate: (Boolean) -> Boolean): List<Boolean> {
    return filterNotTo(ArrayList<Boolean>(), predicate)\n}\n
/**
 * Returns a list containing all elements that are not `null`.\n
 * @sample \
 */

fun <T : Any> Array<out T>.filterNotNullTo(destination: MutableList<T>): T {
    for (element in this) if (element != null) destination.add(element)\n    return destination\n}\n
/**
 * Appends all elements not matching the given \[predicate\] to the given \[destination\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun <C : MutableList<in T>, T : Any> Array<out T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}\n
/**
 * Appends all elements not matching the given \[predicate\] to the given \[destination\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun <C : MutableList<in Byte>> ByteArray.filterNotTo(destination: C, predicate: (Byte) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}\n
/**
 * Appends all elements not matching the given \[predicate\] to the given \[destination\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun <C : MutableList<in Short>> ShortArray.filterNotTo(destination: C, predicate: (Short) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}\n
/**
 * Appends all elements not matching the given \[predicate\] to the given \[destination\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun <C : MutableList<in Int>> IntArray.filterNotTo(destination: C, predicate: (Int) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}\n
/**
 * Appends all elements not matching the given \[predicate\] to the given \[destination\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun <C : MutableList<in Long>> LongArray.filterNotTo(destination: C, predicate: (Long) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}\n
/**
 * Appends all elements not matching the given \[predicate\] to the given \[destination\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun <C : MutableList<in Float>> FloatArray.filterNotTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}\n
/**
 * Appends all elements not matching the given \[predicate\] to the given \[destination\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun <C : MutableList<in Double>> DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}\n
/**
 * Appends all elements not matching the given \[predicate\] to the given \[destination\].\n
 * @sample samples.collections.Collections.Filtering.filter
 */

public inline fun <C : MutableList<in Char>> CharArray.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)\n    return destination\n}
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Boolean>>
BooleanArray.filterNotTo(destination: C, predicate: (Boolean) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Char>>
CharArray.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterTo(destination: C, predicate: (Short) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Int>> IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Long>> LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Double>> DoubleArray.filterTo(destination: C, predicate: (Double) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Char>> CharArray.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.slice(indices: IntRange): List<T> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.slice(indices: IntRange): List<Byte> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Char>> CharArray.slice(indices: IntRange): List<Char> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.slice(indices: IntRange): List<T> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()"\n\n*\n\n@sample
samples.collections.Collections.Filtering.filterTo
\n*\n\npublic inline fun <C : MutableCollection<in Char>> CharArray.slice(indices: IntRange): List<Char> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive +
### Java Code Snippet:

```java
import java.util.Arrays;
import java.util.Collections;
import java.util.List;

public class ArrayUtils {
    public static <T> List<T> sliceArray(T[] array, int[] indices) {
        int size = indices.length;
        return Arrays.asList(array, indices.sliceArray(indices));
    }
}
```

### Description:

This Java code snippet defines a method `sliceArray` that takes a `T` array and an integer array `indices` as parameters and returns a list containing elements at specified indices. The method uses the `sliceArray` method from the `ArrayUtils` class, which is provided by the `java.util.Arrays` class. The `sliceArray` method is overloaded to work with different types of arrays, including `Byte[]`, `Char[]`, `Double[]`, `Float[]`, `Int[]`, `Short[]`, and `Byte[]`. It returns a list containing the elements at the specified indices, with the indices being specified as an `IntRange` or an `IntCollection`.

### Usage Example:

```java
import java.util.Arrays;
import java.util.Collections;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        Integer[] array = {1, 2, 3, 4, 5};
        int[] indices = {1, 3};
        List<Integer> result = ArrayUtils.sliceArray(array, indices);
        System.out.println(result); // Output: [2, 4]
    }
}
```

This example demonstrates how to use the `sliceArray` method to extract elements from a `Integer` array at specified indices `{1, 3}`. The method returns a list containing the values at those indices, `[2, 4]` in this case.
Open Source Used In DNAC Platform Antelope

```kotlin
@throws IllegalArgumentException if \([n]\) is negative.

@sample
```
samples.collections.Collections.Transformations.take(n) *\public fun ByteArray.take(n: Int): List<Byte> {`n
require(n >= 0) { "Requested element count $n is less than zero." }`n `n if (n == 0) return emptyList()`n `n if (n >= size) return toList()`n `n if (n == 1) return listOf(this[0])`n `n var count = 0`n `n val list = ArrayList<Byte> (n)`n `n for (item in this) `{`n `n   list.add(item)`n `n   if (++count == n)`n `n   break`n `n `n return list`n `n }n
* Returns a list containing first [n] elements.`n `n * @throws IllegalArgumentException if [n] is negative.`n `n * @sample samples.collections.Collections.Transformations.take(n) *\public fun ShortArray.take(n: Int): List<Short> {`n
require(n >= 0) { "Requested element count $n is less than zero." }`n `n if (n == 0) return emptyList()`n `n if (n >= size) return toList()`n `n if (n == 1) return listOf(this[0])`n `n var count = 0`n `n val list = ArrayList<Short> (n)`n `n for (item in this) `{`n `n   list.add(item)`n `n   if (++count == n)`n `n   break`n `n `n return list`n `n }n
* Returns a list containing first [n] elements.`n `n * @throws IllegalArgumentException if [n] is negative.`n `n * @sample samples.collections.Collections.Transformations.take(n) *\public fun IntArray.take(n: Int): List<Int> {`n
require(n >= 0) { "Requested element count $n is less than zero." }`n `n if (n == 0) return emptyList()`n `n if (n >= size) return toList()`n `n if (n == 1) return listOf(this[0])`n `n var count = 0`n `n val list = ArrayList<Int> (n)`n `n for (item in this) `{`n `n   list.add(item)`n `n   if (++count == n)`n `n   break`n `n `n return list`n `n }n
* Returns a list containing first [n] elements.`n `n * @throws IllegalArgumentException if [n] is negative.`n `n * @sample samples.collections.Collections.Transformations.take(n) *\public fun FloatArray.take(n: Int): List<Float> {`n
require(n >= 0) { "Requested element count $n is less than zero." }`n `n if (n == 0) return emptyList()`n `n if (n >= size) return toList()`n `n if (n == 1) return listOf(this[0])`n `n var count = 0`n `n val list = ArrayList<Float> (n)`n `n for (item in this) `{`n `n   list.add(item)`n `n   if (++count == n)`n `n   break`n `n `n return list`n `n }n
* Returns a list containing first [n] elements.`n `n * @throws IllegalArgumentException if [n] is negative.`n `n * @sample samples.collections.Collections.Transformations.take(n) *\public fun DoubleArray.take(n: Int): List<Double> {`n
require(n >= 0) { "Requested element count $n is less than zero." }`n `n if (n == 0) return emptyList()`n `n if (n >= size) return toList()`n `n if (n == 1) return listOf(this[0])`n `n var count = 0`n `n val list = ArrayList<Double> (n)`n `n for (item in this) `{`n `n   list.add(item)`n `n   if (++count == n)`n `n   break`n `n `n return list`n `n }n
* Returns a list containing first [n] elements.`n `n * @throws IllegalArgumentException if [n] is negative.`n `n * @sample samples.collections.Collections.Transformations.take(n) *\public fun CharArray.take(n: Int): List<Char> {`n
require(n >= 0) { "Requested element count $n is less than zero." }`n `n if (n == 0) return emptyList()`n `n if (n >= size) return toList()`n `n if (n == 1) return listOf(this[0])`n `n var count = 0`n `n val list = ArrayList<Char> (n)`n `n for (item in this) `{`n `n   list.add(item)`n `n   if (++count == n)`n `n   break`n `n `n return list`n `n }n
* Returns a list containing last [n] elements.`n `n * @throws IllegalArgumentException if [n] is negative.`n `n * @sample samples.collections.Collections.Transformations.take(n) *\public fun TArray.take(n: Int): List<T> {`n
require(n >= 0) { "Requested element count $n is less than zero." }`n `n if (n == 0) return emptyList()`n `n if (n >= size) return toList()`n `n if (n == 1) return listOf(0)`n `n var count = 0`n `n val list = ArrayList<T> (n)`n `n for (index in size - n until size) `{`n `n   list.add(this[index])`n `n   if (++count == n)`n `n   break`n `n `n return list`n `n }n
* Returns a list containing last [n] elements.`n `n * @throws IllegalArgumentException if [n] is negative.`n `n * @sample samples.collections.Collections.Transformations.take(n) *\public fun ByteArray.take(n: Int): List<Byte> {`n
require(n >= 0) { "Requested element count $n is less than zero." }`n `n if (n == 0) return emptyList()`n `n if (n >= size) return toList()`n `n if (n == 1) return listOf(this[0])`n `n var count = 0`n `n val list = ArrayList<Byte> (n)`n `n for (item in this) `{`n `n   list.add(item)`n `n   if (++count == n)`n `n   break`n `n `n return list`n `n }n
* Returns a list containing last [n] elements.`n `n * @throws IllegalArgumentException if [n] is negative.`n `n * @sample samples.collections.Collections.Transformations
size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Byte>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Boolean>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Double>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Float>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Int>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Long>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Short>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Byte>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Double>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Float>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Int>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Long>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.

size
  if (n <= size) return toList();
  if (n == 1) return listOf(this[size - 1]);
  val list = ArrayList<Short>(n);
  for (index in size - n until size)
    list.add(this[index]);
  return list
}

@throws IllegalArgumentException if [n] is negative.
toList()  
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun IntArray.takeLastWhile(predicate: (Int) -> Boolean): List<Int> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index]))
            return drop(index + 1)
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun LongArray.takeLastWhile(predicate: (Long) -> Boolean): List<Long> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index]))
            return drop(index + 1)
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun FloatArray.takeLastWhile(predicate: (Float) -> Boolean): List<Float> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index]))
            return drop(index + 1)
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun DoubleArray.takeLastWhile(predicate: (Double) -> Boolean): List<Double> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index]))
            return drop(index + 1)
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun BooleanArray.takeLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index]))
            return drop(index + 1)
    }
    return toList()
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun CharArray.takeLastWhile(predicate: (Char) -> Boolean): List<Char> {
    for (index in lastIndex downTo 0) {
        if (!predicate(this[index]))
            return drop(index + 1)
    }
    return toList()
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun ByteArray.takeWhile(predicate: (Byte) -> Boolean): List<Byte> {
    val list = ArrayList<Byte>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun ShortArray.takeWhile(predicate: (Short) -> Boolean): List<Short> {
    val list = ArrayList<Short>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun IntArray.takeWhile(predicate: (Int) -> Boolean): List<Int> {
    val list = ArrayList<Int>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun LongArray.takeWhile(predicate: (Long) -> Boolean): List<Long> {
    val list = ArrayList<Long>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun FloatArray.takeWhile(predicate: (Float) -> Boolean): List<Float> {
    val list = ArrayList<Float>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline fun DoubleArray.takeWhile(predicate: (Double) -> Boolean): List<Double> {
    val list = ArrayList<Double>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * 
 * @sample samples.collections.Collections.Transformations.take
 */
public inline funcollection
(Boolean) -> Boolean): List<Char> {\n    val list = ArrayList<Char>() {\n        for (item in this) {\n            if (!predicate(item)) break\n        }\n        list.add(item)\n    }\n    return list\n}

* @ sample samples.collections.Collections.Transformations.take\n * \n *

/**
 * Reverses elements in the array in-place.\n */\npublic fun <T> Array<T>.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }
}

/**
 * Reverses elements in the array in-place.\n */\npublic fun CharArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }
}

/**
 * Reverses elements in the array in-place.\n */\npublic fun BooleanArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }
}

/**
 * Reverses elements in the array in-place.\n */\npublic fun FloatArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }
}

/**
 * Reverses elements in the array in-place.\n */\npublic fun IntArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }
}

/**
 * Reverses elements in the array in-place.\n */\npublic fun ShortArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }
}
this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public fun ShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public fun IntArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public fun LongArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public fun FloatArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public fun DoubleArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
}

/**
 * Reverses elements of the array in the specified range in-place.
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
public fun BooleanArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    if (fromIndex == midPoint) return
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]

this[index] = this[reverseIndex]
this[reverseIndex] = tmp
reverseIndex--
} 
} 

/**
 * Reverses elements of the array in the specified range in-place.
 * @param fromIndex the start of the range (inclusive) to reverse.
 * @param toIndex the end of the range (exclusive) to reverse.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public fun IntArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
    this[fromIndex] = this[toIndex - 1]
    this[toIndex - 1] = tmp
    tmp = 0
    for (index in midPoint until toIndex) {
        val tmp = this[index]
        this[index] = this[index - 1]
        this[index - 1] = tmp
    }
    this[toIndex] = this[toIndex - 1]
    this[toIndex - 1] = tmp
}

/**
 * Returns a list with elements in reversed order.
 */
public fun Array<T>.reversed(): List<T> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun ByteArray.reversedArray(): ByteArray {
    if (isEmpty()) return this
    val result = ByteArray(size)
    val lastIndex = lastIndex
    for (i in 0..lastIndex)
        result[lastIndex - i] = this[i]
    return result
}

/**
 * Returns a list with elements in reversed order.
 */
public fun BooleanArray.reversed(): List<Boolean> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun CharArray.reversed(): List<Char> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun DoubleArray.reversed(): List<Double> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun FloatArray.reversed(): List<Float> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun IntArray.reversed(): List<Int> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun LongArray.reversed(): List<Long> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun ShortArray.reversed(): List<Short> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun <T> Array<out T>.reversed(): List<T> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

/**
 * Returns a list with elements in reversed order.
 */
public fun CharArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
    this[fromIndex] = this[toIndex - 1]
    this[toIndex - 1] = tmp
    tmp = 0
    for (index in midPoint until toIndex) {
        val tmp = this[index]
        this[index] = this[index - 1]
        this[index - 1] = tmp
    }
    this[toIndex] = this[toIndex - 1]
    this[toIndex - 1] = tmp
}

/**
 * Returns a list with elements in reversed order.
 */
public fun IntArray.reversed(fromIndex: Int, toIndex: Int): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val midPoint = (fromIndex + toIndex) / 2
    var reverseIndex = toIndex - 1
    for (index in fromIndex until midPoint) {
        val tmp = this[index]
        this[index] = this[reverseIndex]
        this[reverseIndex] = tmp
        reverseIndex--
    }
    this[fromIndex] = this[toIndex - 1]
    this[toIndex - 1] = tmp
    tmp = 0
    for (index in midPoint until toIndex) {
        val tmp = this[index]
        this[index] = this[index - 1]
        this[index - 1] = tmp
    }
    this[toIndex] = this[toIndex - 1]
    this[toIndex - 1] = tmp
}
```kotlin
/**
 * Returns an array with elements of this array in reversed order.
 */
@SinceKotlin("1.4")
fun CharArray.reversedArray(): CharArray
{
    if (isEmpty()) return this
    val result = CharArray(size)
    val lastIndex = lastIndex
    for (i in 0..lastIndex)
        result[lastIndex - i] = this[i]
    return result
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun Array<T>.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun ByteArray.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun ShortArray.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun IntArray.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun LongArray.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun FloatArray.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun DoubleArray.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun BooleanArray.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun CharArray.shuffle(): Unit
{
    shuffle(Random)
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun <T> Array<T>.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun ByteArray.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun ShortArray.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun IntArray.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun LongArray.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun FloatArray.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun DoubleArray.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun BooleanArray.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}

/**
 * Returns an array with elements of this array in reversed order.
 */
public fun CharArray.shuffle(random: Random): Unit
{
    for (i in lastIndex downTo 1)
        { val j = random.nextInt(i + 1)
            val copy = this[i]
            this[i] = this[j]
            this[j] = copy
        }
}
```

Open Source Used In DNAC Platform Antelope 13657
DoubleArray.sorted(): List<Double> {
    return toTypedArray().apply { sort() }.asList() // Returns a list of all elements sorted according to their natural sort order.
}

fun CharArray.sorted(): List<Char> {
    return toTypedArray().apply { sort() }.asList() // Returns an array with all elements of this array sorted according to their natural sort order.
}

fun <T : Comparable<T>> Array<T>.sortedArray(): Array<T> {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }.asList() // Returns an array with all elements of this array sorted according to their natural sort order.
}

fun <T : Comparable<T>> Array<T>.sortedArrayDescending(): Array<T> {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }.asList() // Returns an array with all elements of this array sorted descending according to their natural sort order.
}

fun Array<out T>.sortedArrayWith(comparator: Comparator<in T>): Array<out T> {
    if (isEmpty()) return this
    return this.copyOf().apply { sortWith(comparator) } // Returns an array with all elements of this array sorted according to the specified comparator.
}

fun Array<out T>.sortedBy(selector: (T) -> R?): List<T> {
    // Returns a list of all elements sorted according to the value returned by specified selector function.
}
**Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

ByteArray.sortedBy(crossinline selector: (Byte) -> R?): List<Byte> {
\n    return sortedWith(compareBy(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

ShortArray.sortedBy(crossinline selector: (Short) -> R?): List<Short> {
\n    return sortedWith(compareBy(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

IntArray.sortedBy(crossinline selector: (Int) -> R?): List<Int> {
\n    return sortedWith(compareBy(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

LongArray.sortedBy(crossinline selector: (Long) -> R?): List<Long> {
\n    return sortedWith(compareBy(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

FloatArray.sortedBy(crossinline selector: (Float) -> R?): List<Float> {
\n    return sortedWith(compareBy(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

DoubleArray.sortedBy(crossinline selector: (Double) -> R?): List<Double> {
\n    return sortedWith(compareBy(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

BooleanArray.sortedBy(crossinline selector: (Boolean) -> R?): List<Boolean> {
\n    return sortedWith(compareBy(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

CharArray.sortedBy(crossinline selector: (Char) -> R?): List<Char> {
\n    return sortedWith(compareBy(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

ByteArray.sortedByDescending(crossinline selector: (Byte) -> R?): List<Byte> {
\n    return sortedWith(compareByDescending(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

ShortArray.sortedByDescending(crossinline selector: (Short) -> R?): List<Short> {
\n    return sortedWith(compareByDescending(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

IntArray.sortedByDescending(crossinline selector: (Int) -> R?): List<Int> {
\n    return sortedWith(compareByDescending(selector))\n}\n
/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n\n* @sample

samples.collections.Collections.Sorting.sortedBy

*public inline fun \n<R : Comparable<R>>

LongArray.sortedByDescending(crossinline selector: (Long) -> R?): List<Long> {
\n    return sortedWith(compareByDescending(selector))\n}
sortedWith(compareByDescending(selector)) // Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.

public fun <R : Comparable<R>> FloatArray.sortedByDescending(crossinline selector: (Float) -> R?): List<Float> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.
 */
public fun <R : Comparable<R>> DoubleArray.sortedByDescending(crossinline selector: (Double) -> R?): List<Double> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.
 */
public fun <R : Comparable<R>> BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> R?): List<Boolean> {
    return sortedWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T : Comparable<T>> Array<out T>.sortedDescending(): List<T> {
    return sortedArrayWith(reverseOrder()).asList()
}

/**
 * Returns a list of all elements sorted according to the specified [comparator].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun ByteArray.sortedWith(comparator: Comparator<in Byte>): List<Byte> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified [comparator].
 */
public fun ShortArray.sortedWith(comparator: Comparator<in Short>): List<Short> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified [comparator].
 */
public fun IntArray.sortedWith(comparator: Comparator<in Int>): List<Int> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified [comparator].
 */
public fun LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified [comparator].
 */
public fun FloatArray.sortedWith(comparator: Comparator<in Float>): List<Float> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified [comparator].
 */
public fun DoubleArray.sortedWith(comparator: Comparator<in Double>): List<Double> {
    return toTypedArray().apply { sortWith(comparator) }.asList()
}

/**
 * Returns a list of all elements sorted according to the specified [comparator].
 */
public fun BooleanArray.sortedWith(comparator: Comparator<in Boolean>): List<Boolean> {
    return toTypedArray().apply { sortWith(comparator) }
fun CharArray.sortedWith(comparator: Comparator<in Char>): List<Char> {
    return toTypedArray().apply {
        sortWith(comparator)
    }.asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
public fun <T> Array<T>.asList(): List<T> {
    return this.asList()
}

/**
 * Returns a [List] that wraps the original array.
 */
public expect fun IntArray.asList(): List<Int>

/**
 * Returns a [List] that wraps the original array.
 */
public expect fun BooleanArray.asList(): List<Boolean>

/**
 * Returns a [List] that wraps the original array.
 */
public expect fun CharArray.asList(): List<Char>

/**
 * Returns `true` if the two specified arrays are *deeply* equal to one another,
 * i.e. contain the same number of the same elements in the same order.
 * If two corresponding elements are nested arrays, they are also compared deeply.
 * If any of arrays contains itself on any nesting level the behavior is undefined.
 * The elements of other types are compared for equality with the
 * [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 * If any of arrays contains itself on any nesting level
 * the behavior is undefined.
 * The elements of other types are compared for equality with the
 * [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 * If any of arrays contains itself on any nesting level
 * the behavior is undefined.
 * The elements of other types are compared for equality with the
 * [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 * If any of arrays contains itself on any nesting level
 * the behavior is undefined.
 * The elements of other types are compared for equality with the
 * [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 * If any of arrays contains itself on any nesting level
 * the behavior is undefined.

/**
 * Returns a hash code based on the contents of this array
 * as if it is [List].
 * Nested arrays are treated as lists too.
 * If any of arrays contains itself on any nesting level the behavior is undefined.
 * The elements of other types are compared for equality with the
 * [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 * If any of arrays contains itself on any nesting level the behavior is undefined.
 * The elements of other types are compared for equality with the
 * [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 * If any of arrays contains itself on any nesting level the behavior is undefined.
 * The elements of other types are compared for equality with the
 * [equals][Any.equals] function.
 * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.
 * If any of arrays contains itself on any nesting level the behavior is undefined.
* Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. The elements are compared for equality with the `==` operator.

For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

The elements are compared for equality with the `==` operator.

**Boolean**

```kotlin
public expect fun ShortArray?.contentEquals(other: ShortArray?): Boolean
```

**Int**

```kotlin
public expect fun IntArray?.contentEquals(other: IntArray?): Boolean
```

**Long**

```kotlin
public expect fun LongArray?.contentEquals(other: LongArray?): Boolean
```

**Float**

```kotlin
public expect fun FloatArray?.contentEquals(other: FloatArray?): Boolean
```

**Double**

```kotlin
public expect fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean
```

**BooleanArray**

```kotlin
public expect infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean
```

**CharArray**

```kotlin
public expect infix fun CharArray?.contentEquals(other: CharArray?): Boolean
```

**ByteArray**

```kotlin
public expect fun ByteArray.contentHashCode(): Int
```

**ShortArray**

```kotlin
public expect fun ShortArray.contentHashCode(): Int
```

**IntArray**

```kotlin
public expect fun IntArray.contentHashCode(): Int
```

**LongArray**

```kotlin
public expect fun LongArray.contentHashCode(): Int
```

**FloatArray**

```kotlin
public expect fun FloatArray.contentHashCode(): Int
```

**DoubleArray**

```kotlin
public expect fun DoubleArray.contentHashCode(): Int
```

**CharArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**ByteArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**BooleanArray**

```kotlin
public expect fun BooleanArray.contentHashCode(): Int
```

**CharArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**ByteArray**

```kotlin
public expect fun ByteArray.contentHashCode(): Int
```

**ShortArray**

```kotlin
public expect fun ShortArray.contentHashCode(): Int
```

**IntArray**

```kotlin
public expect fun IntArray.contentHashCode(): Int
```

**LongArray**

```kotlin
public expect fun LongArray.contentHashCode(): Int
```

**FloatArray**

```kotlin
public expect fun FloatArray.contentHashCode(): Int
```

**DoubleArray**

```kotlin
public expect fun DoubleArray.contentHashCode(): Int
```

**CharArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**ByteArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**BooleanArray**

```kotlin
public expect fun BooleanArray.contentHashCode(): Int
```

**CharArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**ByteArray**

```kotlin
public expect fun ByteArray.contentHashCode(): Int
```

**ShortArray**

```kotlin
public expect fun ShortArray.contentHashCode(): Int
```

**IntArray**

```kotlin
public expect fun IntArray.contentHashCode(): Int
```

**LongArray**

```kotlin
public expect fun LongArray.contentHashCode(): Int
```

**FloatArray**

```kotlin
public expect fun FloatArray.contentHashCode(): Int
```

**DoubleArray**

```kotlin
public expect fun DoubleArray.contentHashCode(): Int
```

**CharArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**ByteArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**BooleanArray**

```kotlin
public expect fun BooleanArray.contentHashCode(): Int
```

**CharArray**

```kotlin
public expect fun CharArray.contentHashCode(): Int
```

**ByteArray**

```kotlin
public expect fun ByteArray.contentHashCode(): Int
```

**ShortArray**

```kotlin
public expect fun ShortArray.contentHashCode(): Int
```

**IntArray**

```kotlin
public expect fun IntArray.contentHashCode(): Int
```

**LongArray**

```kotlin
public expect fun LongArray.contentHashCode(): Int
```

**FloatArray**

```kotlin
public expect fun FloatArray.contentHashCode(): Int
```

**DoubleArray**

```kotlin
public expect fun DoubleArray.contentHashCode(): Int
```
DoubleArray.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")

* Since Kotlin(\"1.1\")

* DeprecatedSinceKotlin(hiddenSince = \"1.4\")

* public expect fun BooleanArray.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* @Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")

* Since Kotlin(\"1.1\")

* DeprecatedSinceKotlin(hiddenSince = \"1.4\")

* public expect fun CharArray.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun <T> Array<out T>?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun ByteArray?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun ShortArray?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun IntArray?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun LongArray?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun FloatArray?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun DoubleArray?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun BooleanArray?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun CharArray?.contentToString(): String\n\n* Returns a string representation of the contents of the specified array as if it is \([List]\)

* @sample samples.collections.Arrays.ContentOperations.contentToString

* Since Kotlin(\"1.4\")

* public expect fun <T> Array<out T>?.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T>\n\n* Copies this array or its subrange into the [destination] array and returns that array.
\n* It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
\n* @param destination the array to copy to.
\n* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
\n* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
\n* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
\n* @throws IndexOutOfBoundsException or [IllegalArgumentException] when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset].
\n* When that index is out of the [destination] array indices range, it's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
\n* @param destination the array to copy to.
\n* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
\n* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
\n* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
\n* @param destination the array to copy to.
\n* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
\n* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
\n* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
\n* @throws
IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset].
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.
* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @return the [destination] array.
*/
@SinceKotlin("1.3")
public expect fun ByteArray.copyInto(destination: ByteArray,
    destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
public expect fun ShortArray.copyInto(destination: ShortArray,
    destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
public expect fun IntArray.copyInto(destination: IntArray,
    destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
public expect fun LongArray.copyInto(destination: LongArray,
    destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
public expect fun FloatArray.copyInto(destination: FloatArray,
    destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray
to.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.

* @param destinationOffset the position in the [destination] array to copy to, 0 by default.
* @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
* @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],
* or when that index is out of the [destination] array indices range.
* @return the [destination] array.
CharArray\n
\n
* Returns new array which is a copy of the original array, resized to the given [newSize].
* The copy is either truncated or padded at the end with zero values if necessary.
* If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
  * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.

@sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf

\n
*\n
public expect fun ByteArray.copyOf(newSize: Int): ByteArray

\n
*\n
public expect fun ShortArray.copyOf(newSize: Int): ShortArray

\n
*\n
public expect fun IntArray.copyOf(newSize: Int): IntArray

\n
*\n
public expect fun LongArray.copyOf(newSize: Int): LongArray

\n
*\n
public expect fun FloatArray.copyOf(newSize: Int): FloatArray

\n
*\n
public expect fun DoubleArray.copyOf(newSize: Int): DoubleArray

\n
*\n
public expect fun BooleanArray.copyOf(newSize: Int): BooleanArray

\n
*\n
public expect fun CharArray.copyOf(newSize: Int): CharArray

\n
*\n
@Suppress("NO ACTUAL FOR EXPECT")

\n
public expect fun <T> Array<T>.copyOf(newSize: Int): Array<T>


Array\lt T\rangle\n\n/**
 * Returns a new array which is a copy of the specified range of the original array.\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */
@Suppress("NO_ACTUAL_FOR_EXPECT")
public expect fun \langle T\rangle.copyOfRange(fromIndex: Int, toIndex: Int): Array\lt T\rangle

/**
 * Returns a new array which is a copy of the specified range of the original array.\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */
public expect fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray

/**
 * Returns a new array which is a copy of the specified range of the original array.\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */
public expect fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray

/**
 * Returns a new array which is a copy of the specified range of the original array.\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */
public expect fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray

/**
 * Returns a new array which is a copy of the specified range of the original array.\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */
public expect fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray

/**
 * Returns a new array which is a copy of the specified range of the original array.\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */
public expect fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray

/**
 * Returns a new array which is a copy of the specified range of the original array.\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */
public expect fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray

/**
 * Returns a new array which is a copy of the specified range of the original array.\n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */
public expect fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray

/**
 * Fills this array or its subrange with the specified \[element\] value.\n * @param fromIndex the start of the range (inclusive) to fill.\n * @param toIndex the end of the range (exclusive) to fill.\n * size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n */
public expect fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray
Returns the range of valid indices for the array.

```kotlin
public val LongArray.indices: IntRange
    get() = IntRange(0, lastIndex)

/**
 * Returns the range of valid indices for the array.
 */
public val FloatArray.indices: IntRange
    get() = IntRange(0, lastIndex)

/**
 * Returns the range of valid indices for the array.
 */
public val DoubleArray.indices: IntRange
    get() = IntRange(0, lastIndex)

/**
 * Returns the range of valid indices for the array.
 */
public val BooleanArray.indices: IntRange
    get() = IntRange(0, lastIndex)

/**
 * Returns the range of valid indices for the array.
 */
public val CharArray.indices: IntRange
    get() = IntRange(0, lastIndex)

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns `true` if the array is empty.
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.isEmpty(): Boolean {
    return size == 0
}

/**
 * Returns the last valid index for the array.
 */
public val LongArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ByteArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ShortArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val IntArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val LongArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ByteArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ShortArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val IntArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val LongArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ByteArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ShortArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val IntArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val LongArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ByteArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val ShortArray.lastIndex: Int
    get() = size - 1

/**
 * Returns the last valid index for the array.
 */
public val IntArray.lastIndex: Int
    get() = size - 1
ShortArray.plus(element: Short): ShortArray

public expect operator fun IntArray.plus(element: Int): IntArray

public expect operator fun LongArray.plus(element: Long): LongArray

public expect operator fun FloatArray.plus(element: Float): FloatArray

public expect operator fun DoubleArray.plus(element: Double): DoubleArray

public expect operator fun BooleanArray.plus(element: Boolean): BooleanArray

public expect operator fun CharArray.plus(element: Char): CharArray

public expect operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray

public expect operator fun ShortArray.plus(elements: Collection<Short>): ShortArray

public expect operator fun IntArray.plus(elements: Collection<Int>): IntArray

public expect operator fun LongArray.plus(elements: Collection<Long>): LongArray

public expect operator fun FloatArray.plus(elements: Collection<Float>): FloatArray

public expect operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray

public expect operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray

public expect operator fun CharArray.plus(elements: Collection<Char>): CharArray

public expect operator fun ByteArray().plus(elements: Array<out T>): ByteArray

public expect operator fun ShortArray().plus(elements: ShortArray): ShortArray

public expect operator fun IntArray().plus(elements: IntArray): IntArray

public expect operator fun LongArray().plus(elements: LongArray): LongArray

public expect operator fun FloatArray().plus(elements: FloatArray): FloatArray

public expect operator fun DoubleArray().plus(elements: DoubleArray): DoubleArray

@Suppress("NO_ACTUAL_FOR_EXPECT")
public expect fun <T> Array<T>.plus(elements: Collection<T>): Array<T>

@Suppress("NO_ACTUAL_FOR_EXPECT")
public expect fun <T> Array<T>.plus(elements: Collection<T>): Array<T>
Array<T>.plusElement(element: T): Array<T>

### Examples

```kotlin
Array<Number> numbers = new Array<Number>()
numbers.plusElement(10)
println(numbers) // [0, 10]
```

### Note

The `plusElement` method adds an element to the beginning of the array if it is not null.

### Usage

This method is useful when you need to add an element at the start of an array, and you want to avoid using the full array operations with `addFirst` method.
FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit

/**
 * Sorts a range in the array in-place.
 * The elements are sorted descending according to their natural sort order.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * @param fromIndex the start of the range (inclusive) to sort.
 * @param toIndex the end of the range (exclusive) to sort.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentOutOfRangeException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.4")
public expect fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size):

*/
@SinceKotlin("1.4")
public expect fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size):

*/
@SinceKotlin("1.4")
public expect fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size):

*/
@SinceKotlin("1.4")
public expect fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size):

*/
@SinceKotlin("1.4")
public expect fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size):

*/
@SinceKotlin("1.4")
public expect fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size):

*/
@SinceKotlin("1.4")
public expect fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size):

*/
@SinceKotlin("1.4")
public expect fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size):

*
sort(fromIndex, toIndex) // reverse(fromIndex, toIndex)\n\n*\n* The elements are sorted descending according to their natural sort order.\n* \n* @param fromIndex the start of the range (inclusive) to sort.\n* \n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* \n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n* \n* @param toIndex the end of the range (exclusive) to sort.\n* \n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* \n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n* \n* @param sort(fromIndex, toIndex)\n\n*\n* Sorts elements of the array in the specified range in-place.\n* \n* The elements are sorted descending according to their natural sort order.\n* \n* @param fromIndex the start of the range (inclusive) to sort.\n* \n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* \n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n* \n* @param toIndex the end of the range (exclusive) to sort.\n* \n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* \n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n* \n* @since Kotlin("1.4")\npublic fun DoubleArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n
/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun CharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n\n    sort(fromIndex, toIndex)\n    reverse(fromIndex, toIndex)\n}\n
/**\n * Sorts the array in-place according to the order specified by the given [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param comparator a Comparator\n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>): Unit {\n\n    sortWith(comparator)\n}\n
/**\n * Returns an array of Boolean containing all of the elements of this generic array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun Array<out Boolean>.toBooleanArray(): BooleanArray {\n\n    return BooleanArray(size) { index -> this[index] }\n}\n
/**\n * Returns an array of Byte containing all of the elements of this generic array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun Array<out Byte>.toByteArray(): ByteArray {\n\n    return ByteArray(size) { index -> this[index] }\n}\n
/**\n * Returns an array of Char containing all of the elements of this generic array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun Array<out Char>.toCharArray(): CharArray {\n\n    return CharArray(size) { index -> this[index] }\n}\n
/**\n * Returns an array of Double containing all of the elements of this generic array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun Array<out Double>.toDoubleArray(): DoubleArray {\n\n    return DoubleArray(size) { index -> this[index] }\n}\n
/**\n * Returns an array of Float containing all of the elements of this generic array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun Array<out Float>.toFloatArray(): FloatArray {\n\n    return FloatArray(size) { index -> this[index] }\n}\n
/**\n * Returns an array of Int containing all of the elements of this generic array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun Array<out Int>.toIntArray(): IntArray {\n\n    return IntArray(size) { index -> this[index] }\n}\n
/**\n * Returns an array of Long containing all of the elements of this generic array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun Array<out Long>.toLongArray(): LongArray {\n\n    return LongArray(size) { index -> this[index] }\n}\n
/**\n * Returns an array of String containing all of the elements of this generic array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic fun Array<out String>.toStringArray(): StringArray {\n\n    return StringArray(size) { index -> this[index] }\n}\n
/**\n * Returns a typed object array containing all of the elements of this primitive array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic expect fun ByteArray.toTypedArray(): Array<Byte> {\n\n    return ToByteArray()\n}\n
/**\n * Returns a typed object array containing all of the elements of this primitive array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic expect fun ShortArray.toTypedArray(): Array<Short> {\n\n    return ToShortArray()\n}\n
/**\n * Returns a typed object array containing all of the elements of this primitive array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic expect fun IntArray.toTypedArray(): Array<Int> {\n\n    return ToIntArray()\n}\n
/**\n * Returns a typed object array containing all of the elements of this primitive array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic expect fun FloatArray.toTypedArray(): Array<Float> {\n\n    return ToFloatArray()\n}\n
/**\n * Returns a typed object array containing all of the elements of this primitive array.\n * \n * @param fromIndex the start of the range (inclusive) to sort. 0 by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * \n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @since Kotlin("1.4")\npublic expect fun DoubleArray.toTypedArray(): Array<Double> {\n\n    return ToDoubleArray()\n}\n
*/
CharArray.toTypedArray(): Array<Char>

public inline fun <T, K, V> Array<out T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

ByteArray.associate(transform: (Byte) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

ShortArray.associate(transform: (Short) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

IntArray.associate(transform: (Int) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

LongArray.associate(transform: (Long) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

DoubleArray.associate(transform: (Double) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}
transform)

/**
 * Returns a [Map] containing key-value pairs provided by [transform] function
 * applied to elements of the given array.
 * If any of two pairs would have the same key the last one gets added to the
 * map. The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K, V> CharArray.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity),
    transform)
}

/**
 * Returns a [Map] containing the elements from the given array indexed by the key
 * returned from [keySelector] function applied to each element. If any two elements would have the same key
 * returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration
 * order of the original array.
 */
public inline fun <T, K> Array<out T>.associateBy(keySelector: (T) -> K): Map<K, T> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, T>(capacity),
    keySelector)
}

/**
 * Returns a [Map] containing the elements from the given array indexed by the key.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K> ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Byte>(capacity),
    keySelector)
}

/**
 * Returns a [Map] containing the elements from the given array indexed by the key.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K> ShortArray.associateBy(keySelector: (Short) -> K): Map<K, Short> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Short>(capacity),
    keySelector)
}

/**
 * Returns a [Map] containing the elements from the given array indexed by the key.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K> IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Int>(capacity),
    keySelector)
}

/**
 * Returns a [Map] containing the elements from the given array indexed by the key.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K> LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Long>(capacity),
    keySelector)
}

/**
 * Returns a [Map] containing the elements from the given array indexed by the key.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 */
public inline fun <K> FloatArray.associateBy(keySelector: (Float) -> K): Map<K, Float> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Float>(capacity),
    keySelector)
}
keySelector)

Returns a [Map] containing the elements from the given array indexed by the key returned from [keySelector] function applied to each element. If any two elements would have the same key returned by [keySelector] the last one gets added to the map. The returned map preserves the entry iteration order of the original array.

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy

public inline fun <K> DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Double>(capacity), keySelector)
}

public inline fun <K> BooleanArray.associateBy(keySelector: (Boolean) -> K): Map<K, Boolean> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Boolean>(capacity), keySelector)
}

public inline fun <K> CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)
}

public inline fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, Double> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Double>(capacity), keySelector)
}

public inline fun <K> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, Boolean> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Boolean>(capacity), keySelector)
}

public inline fun <K> CharArray.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, Char> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)
}

public inline fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, Double> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Double>(capacity), keySelector)
}

public inline fun <K> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, Boolean> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Boolean>(capacity), keySelector)
}

public inline fun <K> CharArray.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, Char> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)
}

public inline fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

public inline fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, Double> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, Double>(capacity), keySelector)
}
capacity = mapCapacity(size).coerceAtLeast(16)
        return associateByTo(LinkedHashMap<K, V>(capacity),
            keySelector, valueTransform) * Returns a [Map] containing the values provided by [valueTransform]
            and indexed by [keySelector] functions applied to elements of the given array.
        * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
        * The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
        keySelector, valueTransform) * Returns a [Map] containing the values provided by [valueTransform]
        and indexed by [keySelector] functions applied to elements of the given array.
        * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
        * The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
        keySelector, valueTransform) * Returns a [Map] containing the values provided by [valueTransform]
        and indexed by [keySelector] functions applied to elements of the given array.
        * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
        * The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
        keySelector, valueTransform) * Returns a [Map] containing the values provided by [valueTransform]
        and indexed by [keySelector] functions applied to elements of the given array.
        * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
        * The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

fun <K, V> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
        keySelector, valueTransform) * Returns a [Map] containing the values provided by [valueTransform]
        and indexed by [keySelector] functions applied to elements of the given array.
        * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
        * The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform

fun <K, V> CharArray.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {
    val capacity = mapCapacity(size).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity),
        keySelector, valueTransform) * Populates and returns the [destination] mutable map with key-value pairs.
        * where key is provided by the [keySelector] function applied to each element of the given array.
        * value is the element itself.
        * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
        * The returned map preserves the entry iteration order of the original array.

@sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo

fun <T, K, M : MutableMap<K, V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        destination.put(keySelector(element), element)
    }
    return destination
}

fun <M : MutableMap<K, V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K): M { for
(element in this) \{\n    destination.put(keySelector(element), element)\n\} \n\nreturn destination
*\n*
Populates and returns the [destination] mutable map with key-value pairs.\n*\nwhere key is provided by the [keySelector] function applied to each element of the given array\n*\nand value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in T>> TArray.associateByTo(destination: M, keySelector: (T) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Short>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Int>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Long>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Float>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Double>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Boolean>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Char>> CharArray.associateByTo(destination: M, keySelector: (Char) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Any>> AnyArray.associateByTo(destination: M, keySelector: (Any) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Any>> AnyArray.associateByTo(destination: M, keySelector: (Any) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Any>> AnyArray.associateByTo(destination: M, keySelector: (Any) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Any>> AnyArray.associateByTo(destination: M, keySelector: (Any) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Any>> AnyArray.associateByTo(destination: M, keySelector: (Any) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Any>> AnyArray.associateByTo(destination: M, keySelector: (Any) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : MutableMap<in K, in Any>> AnyArray.associateByTo(destination: M, keySelector: (Any) -> K): M \{\n    for (element in this) \{\n        destination.put(keySelector(element), element)\n    \}\n    return destination
\}

Populates and returns the [destination] mutable map with key-value pairs.\n* where key is provided by the [keySelector] function applied to each element of the given array\n* and value is the element itself.\n* *\nIf any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n* *\n*\n@sample samples.collections.Transforms.associateArrayOfPrimitivesByTo\n*\n\npublic inline fun <K, M : ….}
inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs, where key is provided by the [keySelector] function and value is provided by the [valueTransform] function applied to elements of the given array.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 */
public inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        destination.put(keySelector(element),
        valueTransform(element))
    }
    return destination
}
/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
 */
public inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo
 */
public inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        destination.put(keySelector(element), valueTransform(element))
    }
    return destination
}

/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
 */
public inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
 */
public inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform: (Byte) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
 */
public inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
 */
public inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
 */
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
 */
public inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}

/**
 * Populates and returns the \[destination\] mutable map with key-value pairs
 * provided by [transform] function applied to each element of the given array.
 * 
 * If any of two pairs would have the same key the last one gets added to the map.
 * 
 * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo
 */
public inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform: (Double) -> Pair<K, V>): M {
    for (element in this) {
        destination += transform(element)
    }
    return destination
}
returned map preserves the entry iteration order of the original array.

@sample

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <V>

FloatArray.associateWith(valueSelector: (Float) -> V): Map<Float, V>

{ 
    val result = LinkedHashMap<Float, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/**
 * Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.
 * If any two elements are equal, the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <V>

DoubleArray.associateWith(valueSelector: (Double) -> V): Map<Double, V>

{ 
    val result = LinkedHashMap<Double, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/**
 * Returns a [Map] where keys are elements from the given array and values are produced by the [valueSelector] function applied to each element.
 * If any two elements are equal, the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original array.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <V>

BooleanArray.associateWith(valueSelector: (Boolean) -> V): Map<Boolean, V>

{ 
    val result = LinkedHashMap<Boolean, V>(mapCapacity(size.coerceAtMost(128)).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,
 * where key is the element itself and value is provided by the [valueSelector] function applied to that key.
 * If any two elements are equal, the last one overwrites the former value in the map.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <V, M : MutableMap<in Byte, in V>>

ByteArray.associateWithTo(destination: M, valueSelector: (Byte) -> V): M

{ 
    for (element in this)

        destination.put(element, valueSelector(element))

    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,
 * where key is the element itself and value is provided by the [valueSelector] function applied to that key.
 * If any two elements are equal, the last one overwrites the former value in the map.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <V, M : MutableMap<in Short, in V>>

ShortArray.associateWithTo(destination: M, valueSelector: (Short) -> V): M

{ 
    for (element in this)

        destination.put(element, valueSelector(element))

    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,
 * where key is the element itself and value is provided by the [valueSelector] function applied to that key.
 * If any two elements are equal, the last one overwrites the former value in the map.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <V, M : MutableMap<in Char, in V>>

CharArray.associateWithTo(destination: M, valueSelector: (Char) -> V): M

{ 
    for (element in this)

        destination.put(element, valueSelector(element))

    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,
 * where key is the element itself and value is provided by the [valueSelector] function applied to that key.
 * If any two elements are equal, the last one overwrites the former value in the map.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <K, V, M : MutableMap<in K, in V>>

Array<out K>.associateWithTo(destination: M, valueSelector: (K) -> V): M

{ 
    for (element in this)

        destination.put(element, valueSelector(element))

    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,
 * where key is the element itself and value is provided by the [valueSelector] function applied to that key.
 * If any two elements are equal, the last one overwrites the former value in the map.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <K, V, M : MutableMap<in Short, in V>>

ShortArray.associateWithTo(destination: M, valueSelector: (Short) -> V): M

{ 
    for (element in this)

        destination.put(element, valueSelector(element))

    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,
 * where key is the element itself and value is provided by the [valueSelector] function applied to that key.
 * If any two elements are equal, the last one overwrites the former value in the map.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <V, M : MutableMap<in Byte, in V>>

ByteArray.associateWithTo(destination: M, valueSelector: (Byte) -> V): M

{ 
    for (element in this)

        destination.put(element, valueSelector(element))

    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,
 * where key is the element itself and value is provided by the [valueSelector] function applied to that key.
 * If any two elements are equal, the last one overwrites the former value in the map.
 *
 * @sample
 */

@SinceKotlin("1.4")
@kotlin.internal.InlineOnly

public inline fun <V, M : MutableMap<in Short, in V>>

ShortArray.associateWithTo(destination: M, valueSelector: (Short) -> V): M

{ 
    for (element in this)

        destination.put(element, valueSelector(element))

    return destination
}
equal, the last one overwrites the former value in the map.

```kotlin
/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, \n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * If any two elements are equal, the last one overwrites the former value in the map.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Int, in V>>
IntArray.associateWithTo(destination: M, valueSelector: (Int) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, \n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * If any two elements are equal, the last one overwrites the former value in the map.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Long, in V>>
LongArray.associateWithTo(destination: M, valueSelector: (Long) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, \n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * If any two elements are equal, the last one overwrites the former value in the map.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Float, in V>>
FloatArray.associateWithTo(destination: M, valueSelector: (Float) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, \n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * If any two elements are equal, the last one overwrites the former value in the map.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Double, in V>>
DoubleArray.associateWithTo(destination: M, valueSelector: (Double) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, \n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * If any two elements are equal, the last one overwrites the former value in the map.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <V, M : MutableMap<in Boolean, in V>>
BooleanArray.associateWithTo(destination: M, valueSelector: (Boolean) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <T, C : MutableCollection<in T>>
Array<out T>.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Byte>>
ByteArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Short>>
ShortArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Int>>
IntArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}
```

Open Source Used In DNAC Platform Antelope 13686
C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Long>> LongArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Float>> FloatArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Double>> DoubleArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 */
public fun <C : MutableCollection<in Boolean>> BooleanArray.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun <T> Array<out T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>(mapCapacity(size)))
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun ByteArray.toHashSet(): HashSet<Byte> {
    return toCollection(HashSet<Byte>(mapCapacity(size)))
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun ShortArray.toHashSet(): HashSet<Short> {
    return toCollection(HashSet<Short>(mapCapacity(size)))
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun IntArray.toHashSet(): HashSet<Int> {
    return toCollection(HashSet<Int>(mapCapacity(size)))
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun LongArray.toHashSet(): HashSet<Long> {
    return toCollection(HashSet<Long>(mapCapacity(size)))
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun FloatArray.toHashSet(): HashSet<Float> {
    return toCollection(HashSet<Float>(mapCapacity(size)))
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun DoubleArray.toHashSet(): HashSet<Double> {
    return toCollection(HashSet<Double>(mapCapacity(size)))
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun BooleanArray.toHashSet(): HashSet<Boolean> {
    return toCollection(HashSet<Boolean>(mapCapacity(size)))
}

/**
 * Returns a new [HashSet] of all elements.
 */
public fun CharArray.toHashSet(): HashSet<Char> {
    return toCollection(HashSet<Char>(mapCapacity(size.coerceAtMost(128)))
}

/**
 * Returns a new [List] containing all elements.
 */
public fun <T> Array<out T>.toList(): List<T> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toList()<List<T>>()
    }
}

/**
 * Returns a new [List] containing all elements.
 */
public fun ByteArray.toList(): List<Byte> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toList()<List<Byte>>()
    }
}

/**
 * Returns a new [List] containing all elements.
 */
public fun ShortArray.toList(): List<Short> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toList()<List<Short>>()
    }
}

/**
 * Returns a new [List] containing all elements.
 */
public fun IntArray.toList(): List<Int> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toList()<List<Int>>()
    }
}

/**
 * Returns a new [List] containing all elements.
 */
public fun LongArray.toList(): List<Long> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toList()<List<Long>>()
    }
}

/**
 * Returns a new [List] containing all elements.
 */
public fun FloatArray.toList(): List<Float> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toList()<List<Float>>()
    }
}

/**
 * Returns a new [List] containing all elements.
 */
public fun DoubleArray.toList(): List<Double> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toList()<List<Double>>()
    }
}

/**
 * Returns a new [List] containing all elements.
 */
public fun BooleanArray.toList(): List<Boolean> {
    return when (size) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toList()<List<Boolean>>()
    }
}

/**
 * Returns a new [List] containing all elements.
 */
public fun CharArray.toList(): List<Char> {
    return when (size) {
        0 -> emptyList()
        1 ->
listOf(this[0])

else -> this.toMutableList()

return this.asList() as *MutableList*

Return a new [MutableList] filled with all elements of this array.

Array.asList(): *MutableList*

Returns a new [MutableList] filled with all elements of this array.

ByteArray.asList(): *MutableList*

Returns a new [MutableList] filled with all elements of this array.

CharArray.asList(): *MutableList*

Returns a new [MutableList] filled with all elements of this array.

CharArray.toMutableList(): *MutableList*

Returns a new [MutableList] filled with all elements of this array.

ByteArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

BooleanArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

CharArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

IntArray.toMutableList(): *MutableList*

Returns a new [MutableList] filled with all elements of this array.

IntArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

LongArray.toMutableList(): *MutableList*

Returns a new [MutableList] filled with all elements of this array.

CharArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

DoubleArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

FloatArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

ByteArray.toMutableList(): *MutableList*

Returns a new [MutableList] filled with all elements of this array.

ByteArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.

ByteArray.toMutableList(): *MutableList*

Returns a new [MutableList] filled with all elements of this array.

ByteArray.toSet(): *Set*

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
`flatMap(transform: (T) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMap(transform: (Byte) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMap(transform: (Short) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMap(transform: (Int) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMap(transform: (Long) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMap(transform: (Float) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMap(transform: (Double) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMap(transform: (Boolean) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMap(transform: (Char) -> Iterable<R>): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMap
```

`flatMapTo(ArrayList<R>(), transform): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMapTo
```

`flatMapIndexedTo(ArrayList<R>(), transform): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMapIndexedTo
```

`flatMapTo(ArrayList<R>(), transform): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMapTo
```

`flatMapIndexedTo(ArrayList<R>(), transform): List<R>`

Returns a single list of all elements yielded from results of `transform` function being invoked on each element of original array.

```kotlin
@sample samples.collections.Collections.Transformations.flatMapIndexedTo
```
flatMapIndexedTo(ArrayList<R>(), transform)\n\n/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original array.
 *
@sample samples.collections.Collections.Transformations.flatMapIndexed
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedTo")
@kotlin.internal.InlineOnly
public inline fun <R> ShortArray.flatMapIndexed(transform: (index: Int, Short) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original array.
 *
@sample samples.collections.Collections.Transformations.flatMapIndexed
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedTo")
@kotlin.internal.InlineOnly
public inline fun <R> IntArray.flatMapIndexed(transform: (index: Int, Int) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original array.
 *
@sample samples.collections.Collections.Transformations.flatMapIndexed
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedTo")
@kotlin.internal.InlineOnly
public inline fun <R> LongArray.flatMapIndexed(transform: (index: Int, Long) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original array.
 *
@sample samples.collections.Collections.Transformations.flatMapIndexed
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedTo")
@kotlin.internal.InlineOnly
public inline fun <R> FloatArray.flatMapIndexed(transform: (index: Int, Float) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original array.
 *
@sample samples.collections.Collections.Transformations.flatMapIndexed
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedTo")
@kotlin.internal.InlineOnly
public inline fun <R> DoubleArray.flatMapIndexed(transform: (index: Int, Double) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original array.
 *
@sample samples.collections.Collections.Transformations.flatMapIndexed
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedTo")
@kotlin.internal.InlineOnly
public inline fun <R> BooleanArray.flatMapIndexed(transform: (index: Int, Boolean) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element and its index in the original array.
 *
@sample samples.collections.Collections.Transformations.flatMapIndexed
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedTo")
@kotlin.internal.InlineOnly
public inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}
flatMapIndexedTo(ArrayList<R>(), transform)\n
/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapIndexedTo(destination: C, transform: (index: Int, Byte) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, Short) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> IntArray.flatMapIndexedTo(destination: C, transform: (index: Int, Int) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> LongArray.flatMapIndexedTo(destination: C, transform: (index: Int, Long) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapIndexedTo(destination: C, transform: (index: Int, Float) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("flatMapIndexedIterableTo")
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapIndexedTo(destination: C, transform: (index: Int, Double) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

Open Source Used In DNAC Platform Antelope 13691
Int, Double) -> Iterable<R>: C {\n    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
        return destination
    }
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4") @OptIn(kotlin.experimental.ExperimentalTypeInference::class) @OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapIndexedTo(destination: C, transform: (index: Int, Boolean) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
        return destination
    }
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4") @OptIn(kotlin.experimental.ExperimentalTypeInference::class) @OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedIterableTo")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> CharArray.flatMapIndexedTo(destination: C, transform: (index: Int, Char) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
        return destination
    }
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element\n and its index in the original array, to the given [destination].
 */
@SinceKotlin("1.4") @OptIn(kotlin.experimental.ExperimentalTypeInference::class) @OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName("flatMapIndexedSequenceTo")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
        return destination
    }
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapTo(destination: C, transform: (Byte) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapTo(destination: C, transform: (Short) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> IntArray.flatMapTo(destination: C, transform: (Int) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> LongArray.flatMapTo(destination: C, transform: (Long) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapTo(destination: C, transform: (Float) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapTo(destination: C, transform: (Double) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}
destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> CharArray.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("flatMapSequenceTo")
public inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <T, K> Array<out T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K> ByteArray.groupBy(keySelector: (Byte) -> K): Map<K, List<Byte>> {
    return groupByTo(LinkedHashMap<K, MutableList<Byte>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K> ShortArray.groupBy(keySelector: (Short) -> K): Map<K, List<Short>> {
    return groupByTo(LinkedHashMap<K, MutableList<Short>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K> IntArray.groupBy(keySelector: (Int) -> K): Map<K, List<Int>> {
    return groupByTo(LinkedHashMap<K, MutableList<Int>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K> LongArray.groupBy(keySelector: (Long) -> K): Map<K, List<Long>> {
    return groupByTo(LinkedHashMap<K, MutableList<Long>>(), keySelector)
}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function
 * applied to each element and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K> FloatArray.groupBy(keySelector: (Float) -> K): Map<K, List<Float>> {
    return groupByTo(LinkedHashMap<K, MutableList<Float>>(), keySelector)
}
```kotlin
/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K, V>
LongArray.groupBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, List<V>> {
    return
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K, V>
FloatArray.groupBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, List<V>> {
    return
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K, V>
DoubleArray.groupBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, List<V>> {
    return
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K, V>
BooleanArray.groupBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, List<V>> {
    return
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array
 * by the key returned by the given [keySelector] function applied to the element
 * and returns a map where each group key is associated with a list of corresponding values.
 * The returned map preserves the entry iteration order of the keys produced from the original array.
 */
public inline fun <K, V>
CharArray.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {
    return
}
```

---

**Open Source Used In DNAC Platform Antelope 13695**
destination.getOrPut(key) { ArrayList<Long>() } \n        list.add(element) \n    } \n    return destination \n}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function \n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. \n * \n * @return The [destination] map. \n * \n * @public inline fun <K, M : MutableMap<K, MutableList<Int>>> IntArray.groupByTo(destination: M, \n * keySelector: (Int) -> K): M \n * \n * for (element in this) \n * val key = keySelector(element) \n * val list = \n * destination.getOrPut(key) \n * ArrayList<Int>() \n * list.add(element) \n * return destination \n}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function \n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. \n * \n * @return The [destination] map. \n * \n * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M : MutableMap<K, Long>> LongArray.groupByTo(destination: M, \n * keySelector: (Long) -> K): M \n * \n * for (element in this) \n * val key = keySelector(element) \n * val list = \n * destination.getOrPut(key) \n * ArrayList<Long>() \n * list.add(element) \n * return destination \n}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function \n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. \n * \n * @return The [destination] map. \n * \n * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M : MutableMap<K, Float>> FloatArray.groupByTo(destination: M, \n * keySelector: (Float) -> K): M \n * \n * for (element in this) \n * val key = keySelector(element) \n * val list = \n * destination.getOrPut(key) \n * ArrayList<Float>() \n * list.add(element) \n * return destination \n}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function \n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. \n * \n * @return The [destination] map. \n * \n * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M : MutableMap<K, Double>> DoubleArray.groupByTo(destination: M, \n * keySelector: (Double) -> K): M \n * \n * for (element in this) \n * val key = keySelector(element) \n * val list = \n * destination.getOrPut(key) \n * ArrayList<Double>() \n * list.add(element) \n * return destination \n}

/**
 * Groups elements of the original array by the key returned by the given [keySelector] function \n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements. \n * \n * @return The [destination] map. \n * \n * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <K, M : MutableMap<K, Boolean>> BooleanArray.groupByTo(destination: M, \n * keySelector: (Boolean) -> K): M \n * \n * for (element in this) \n * val key = keySelector(element) \n * val list = \n * destination.getOrPut(key) \n * ArrayList<Boolean>() \n * list.add(element) \n * return destination \n}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array \n * by the key returned by the given [keySelector] function \n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding values. \n * \n * @return The [destination] map. \n * \n * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <T, K, M : MutableMap<K, MutableList<T>>> Array<out T>.groupByTo(destination: M, \n * keySelector: (T) -> K): M \n * \n * for (element in this) \n * val key = keySelector(element) \n * val list = \n * destination.getOrPut(key) \n * ArrayList<T>() \n * list.add(element) \n * return destination \n}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array \n * by the key returned by the given [keySelector] function \n * applied to the element \n * and puts to the [destination] map each group key associated with a list of corresponding values. \n * \n * @return The [destination] map. \n * \n * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <T, K, M : MutableMap<K, MutableList<T>>> Array<out T> groupByTo(destination: M, \n * keySelector: (T) -> K, \n * valueTransform: (T) -> V): M \n * \n * for (element in this) \n * val key = keySelector(element) \n * val list = \n * destination.getOrPut(key) \n * ArrayList<T>() \n * list.add(valueTransform(element)) \n * return destination \n}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array \n * by the key returned by the given [keySelector] function \n * applied to the element \n * and puts to the [destination] map each group key associated with a list of corresponding values. \n * \n * @return The [destination] map. \n * \n * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <T, K, M : MutableMap<K, MutableList<T>>> Array<out T> groupByTo(destination: M, \n * keySelector: (T) -> K, \n * valueTransform: (T) -> V): M \n * \n * for (element in this) \n * val key = keySelector(element) \n * val list = \n * destination.getOrPut(key) \n * ArrayList<T>() \n * list.add(valueTransform(element)) \n * return destination \n}
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ShortArray.groupByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original array by the key returned by the given [keySelector] function applied to the element and puts to the [destination] map each group key associated with a list of corresponding values.
 * @return The [destination] map.
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}
original array

* by the key returned by the given [keySelector] function applied to the element

* and puts to the

[destination] map each group key associated with a list of corresponding values.

* @return The [destination]

map.

* * @sample samples.collections.Collections.Transformations.groupByKeysAndValues*/

*/*

infun <K, V, M : MutableMap<K, MutableList<V>>>

CharArray.groupByTo(destination: M, keySelector:

(Char) -> K, valueTransform: (Char) -> V): M { val key = keySelector(element) val list = destination.getOrElse(key) { ArrayList<V>() } list.add(valueTransform(element)) } return destination

*/*

* @return The [destination]

* @sample samples.collections.Grouping.groupingByEachCount*/

*@

SinceKotlin("1.1")*/

public inline fun <T, K>

Array<out T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {

return object : Grouping<T,

K> {

sourceIterator(): Iterator<T> = this@groupingBy.iterator()

keyOf(element: T): K = keySelector(element)

}

}*/

public inline fun <T, R> Array<out T>.map(transform: (T) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

public inline fun <R> ByteArray.map(transform: (Byte) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

public inline fun <R> ShortArray.map(transform: (Short) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

public inline fun <R> IntArray.map(transform: (Int) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

public inline fun <R> LongArray.map(transform: (Long) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

public inline fun <R> FloatArray.map(transform: (Float) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

public inline fun <R> DoubleArray.map(transform: (Double) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

public inline fun <R> BooleanArray.map(transform: (Boolean) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

public inline fun <R> CharArray.map(transform: (Char) -> R): List<R> {

return mapTo(ArrayList<R>(size), transform)

}*/

mapIndexedTo(transform: (index: Int, T) -> R): List<R> { val return mapIndexedTo(ArrayList<R>(size), transform)

}*/

To(CharArray绎transformation: (index: Int, T) -> R): List<R> { val return mapIndexedTo(ArrayList<R>(size), transform)

}*/

@openSourceUsedInDNACPlatformAntelope
this \n destination.add(transform(index++, item)) \n return destination\n }\n
* Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\n T T.mapNonNull(transform : (T -> R)) : List\ T { \n return mapNonNullTo(Array\ List\ T\ )\ ,\ transform\n }\n
* Applies the given [transform] function to each element in the original array\n * and appends only the non-null results to the given [destination].\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\ T T.mapNonNullTo(destination : C, transform : (T -> R)) : C { \n forEach \n element -> transform(element)?.let { destination.add(it) } \n return destination\n }\n
* Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\ T T.mapNonNullTo(destination : C, transform : (T -> R)) : C { \n for (item in this) \n destination.add(transform(index++, item)) \n return destination\n }\n
* Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\ T T.mapNonNullTo(destination : C, transform : (T -> R)) : C { \n for (item in this) \n destination.add(transform(index++, item)) \n return destination\n }\n
* Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\ T T.mapNonNullTo(destination : C, transform : (T -> R)) : C { \n for (item in this) \n destination.add(transform(index++, item)) \n return destination\n }\n
* Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\ T T.mapNonNullTo(destination : C, transform : (T -> R)) : C { \n for (item in this) \n destination.add(transform(index++, item)) \n return destination\n }\n
* Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\ T T.mapNonNullTo(destination : C, transform : (T -> R)) : C { \n for (item in this) \n destination.add(transform(index++, item)) \n return destination\n }\n
* Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\ T T.mapNonNullTo(destination : C, transform : (T -> R)) : C { \n for (item in this) \n destination.add(transform(index++, item)) \n return destination\n }\n
* Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @sample samples.collections.Collections.Transformations.mapNonNull\n * An <T, R : Any> Array\ T T.mapNonNullTo(destination : C, transform : (T -> R)) : C { \n for (item in this) \n destination.add(transform(index++, item)) \n return destination\n }
transform: (Byte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return
}

/**
 * Applies the given [transform] function to each element of the original array and
 * appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> ShortArray.mapTo(destination: C, transform: (Short) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and
 * appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> IntArray.mapTo(destination: C, transform: (Int) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and
 * appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and
 * appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and
 * appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> DoubleArray.mapTo(destination: C, transform: (Double) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and
 * appends the results to the given [destination].
 */
public inline fun <R, C : MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array and
 * appends the results to the given [destination].
 */
public inline fun <R, C : MutableList<in R>> CharArray.mapTo(destination: C, transform: (Char) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun <T> Array<out T>.withIndex(): Iterable<IndexedValue<T>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun ByteArray.withIndex(): Iterable<IndexedValue<Byte>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun ShortArray.withIndex(): Iterable<IndexedValue<Short>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun IntArray.withIndex(): Iterable<IndexedValue<Int>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun LongArray.withIndex(): Iterable<IndexedValue<Long>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun FloatArray.withIndex(): Iterable<IndexedValue<Float>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun DoubleArray.withIndex(): Iterable<IndexedValue<Double>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun BooleanArray.withIndex(): Iterable<IndexedValue<Boolean>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array into an
 * [IndexedValue] containing the index of that element and the element itself.
 */
public fun CharArray.withIndex(): Iterable<IndexedValue<Char>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a list containing only distinct elements from the given array.
 */
public fun * * *
be present in the resulting list. The elements in the resulting list are in the same order as they were in the source array. Returns a list containing only distinct elements from the given array. The elements in the resulting list are in the same order as they were in the source array. The elements in the resulting list are in the same order as they were in the source array.
/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> TArray.distinctBy(selector: (T) -> K): List<T> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> ByteArray.distinctBy(selector: (Byte) -> K): List<Byte> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> ShortArray.distinctBy(selector: (Short) -> K): List<Short> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}

/** Returns a list containing only elements from the given array
 * having distinct keys returned by the given [selector] function.
 * The elements in the resulting list are in the same order as they were in the source array.
 */
public inline fun <T> CharArray.distinctBy(selector: (Char) -> K): List<Char> {
    val set = this.toMutableSet()
    set.retainAll(set)
    return set
}
public infix fun BooleanArray.subtract(other: Iterable<Boolean>):
    Set<Boolean> { val set = this.toMutableSet() set.removeAll(other) return set } /* The returned set preserves the element iteration order of the original array. */

public infix fun CharArray.subtract(other:Iterable<Char>):
    Set<Char> { val set = this.toMutableSet() set.removeAll(other) return set } /* Returns a new [MutableSet] containing all distinct elements from the given array. */

public fun <T> Array<out T>.toMutableSet(): MutableSet<T> {
    return toCollection(LinkedHashSet<T>(mapCapacity(size)))
}

public fun ByteArray.toMutableSet():
    MutableSet<Byte> {
    return toCollection(LinkedHashSet<Byte>(mapCapacity(size)))
}

public fun ShortArray.toMutableSet():
    MutableSet<Short> {
    return toCollection(LinkedHashSet<Short>(mapCapacity(size)))
}

public fun IntArray.toMutableSet():
    MutableSet<Int> {
    return toCollection(LinkedHashSet<Int>(mapCapacity(size)))
}

public fun LongArray.toMutableSet():
    MutableSet<Long> {
    return toCollection(LinkedHashSet<Long>(mapCapacity(size)))
}

public fun FloatArray.toMutableSet():
    MutableSet<Float> {
    return toCollection(LinkedHashSet<Float>(mapCapacity(size)))
}

public fun DoubleArray.toMutableSet():
    MutableSet<Double> {
    return toCollection(LinkedHashSet<Double>(mapCapacity(size)))
}

public fun BooleanArray.toMutableSet():
    MutableSet<Boolean> {
    return toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))
}

public fun CharArray.toMutableSet():
    MutableSet<Char> {
    return toCollection(LinkedHashSet<Char>(mapCapacity(size).coerceAtMost(128)))
}

public infix fun <T> Array<out T>.union(other: Iterable<T>): Set<T> {
    return toCollection(LinkedHashSet<T>(mapCapacity(size)))
}

public infix fun ByteArray.union(other: Iterable<Byte>): Set<Byte> {
    return toCollection(LinkedHashSet<Byte>(mapCapacity(size)))
}

public infix fun ShortArray.union(other: Iterable<Short>): Set<Short> {
    return toCollection(LinkedHashSet<Short>(mapCapacity(size)))
}

public infix fun IntArray.union(other: Iterable<Int>): Set<Int> {
    return toCollection(LinkedHashSet<Int>(mapCapacity(size)))
}

public infix fun LongArray.union(other: Iterable<Long>): Set<Long> {
    return toCollection(LinkedHashSet<Long>(mapCapacity(size)))
}

public infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {
    return toCollection(LinkedHashSet<Float>(mapCapacity(size)))
}

public infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {
    return toCollection(LinkedHashSet<Double>(mapCapacity(size)))
}

public infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {
    return toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))
}

public infix fun CharArray.union(other: Iterable<Char>): Set<Char> {
    return toCollection(LinkedHashSet<Char>(mapCapacity(size).coerceAtMost(128)))
}
end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */
public infix fun IntArray.union(other: Iterable<Int>): Set<Int> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.\n * \n * Those elements of the [other] collection that are unique are iterated in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */
public infix fun LongArray.union(other: Iterable<Long>): Set<Long> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.\n * \n * Those elements of the [other] collection that are unique are iterated in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */
public infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.\n * \n * Those elements of the [other] collection that are unique are iterated in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */
public infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.\n * \n * Those elements of the [other] collection that are unique are iterated in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */
public infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns a set containing all distinct elements from both collections.\n * \n * Those elements of the [other] collection that are unique are iterated in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */
public infix fun CharArray.union(other: Iterable<Char>): Set<Char> {
    val set = this.toMutableSet()
    set.addAll(other)
    return set
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun <T> Array<out T>.all(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun ByteArray.all(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun ShortArray.all(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun IntArray.all(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun LongArray.all(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun FloatArray.all(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun DoubleArray.all(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun BooleanArray.all(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].\n * \n * @sample samples.collection...
Returns `true` if all elements match the given [predicate].

```kotlin
public inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}
```

```kotlin
/**
 * Returns `true` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun <T> Array<out T>.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun ByteArray.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun ShortArray.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun IntArray.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun LongArray.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun FloatArray.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun DoubleArray.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if array has at least one element.
 * 
 * @sample samples.collections.Collections.Aggregates.any
 */
public fun BooleanArray.any(): Boolean {
    return !isEmpty()
}
```

```kotlin
/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun ByteArray.any(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun ShortArray.any(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun IntArray.any(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun LongArray.any(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun FloatArray.any(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun DoubleArray.any(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun BooleanArray.any(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
/**
 * Returns `true` if at least one element matches the given [predicate].
 * 
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun CharArray.any(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}
```

```kotlin
/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline
fun <T> Array<out T>.count(): Int {return size}

/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun ByteArray.count(): Int {return size}

/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun ShortArray.count(): Int {return size}

/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun IntArray.count(): Int {return size}

/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun LongArray.count(): Int {return size}

/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun FloatArray.count(): Int {return size}

/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun DoubleArray.count(): Int {return size}

/**
 * Returns the number of elements in this array.
 */
@kotlin.internal.InlineOnly
public inline fun BooleanArray.count(): Int {return size}

/**
 * Returns the number of elements matching the given [predicate].
 */
@kotlin.internal.InlineOnly
public inline fun CharArray.count(predicate: (Char) -> Boolean): Int {var count = 0
    for (element in this) if (predicate(element)) ++count
    return count}

/**
 * Accumulates value starting with [initial] value and
 * applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <T, R> Array<out T>.fold(initial: R, operation: (acc: R, T) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <R> ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <R> ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <R> IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <R> LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <R> FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <R> DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.
 * Returns the specified [initial] value if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 */
public inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator}
Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.\n
* \n
* \n
* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n
* public inline fun <R> IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {\n    var accumulator = initial\n    for (element in this)\n        accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n
* public inline fun <R> LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R {\n    var accumulator = initial\n    for (element in this)\n        accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n
* public inline fun <R> FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {\n    var accumulator = initial\n    for (element in this)\n        accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n
* public inline fun <R> DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {\n    var accumulator = initial\n    for (element in this)\n        accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* public inline fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this)\n        accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* public inline fun <R> ByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): R {
    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* public inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* public inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {
    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.\n
* %public inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++, accumulator, element)\n    return accumulator\n}\n
* * *\n
* Returns the specified [initial] value if the array is empty.\n
* \n
* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
applying [operation] from left to right * to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value\n *

 agli inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
} // Accumulates value starting with initial value and applying [operation] from left to right * to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value\n *

agli inline fun <R> LongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
} // Accumulates value starting with initial value and applying [operation] from left to right * to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value\n *

agli inline fun <R> DoubleArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
} // Accumulates value starting with initial value and applying [operation] from left to right * to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value\n *

agli inline fun <R> CharArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R) {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
} // Accumulates value starting with initial value and applying [operation] from left to right * to current accumulator value and each element with its index in the original array.

* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value\n *

agli inline fun <R> ByteArray.foldRight(initial: R, operation: (Byte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
} // Accumulates value starting with initial value and applying [operation] from right to left * to each element and current accumulator value.

* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value\n *

agli inline fun <T: array, R> Array<out T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
} // Accumulates value starting with initial value and applying [operation] from right to left * to each element and current accumulator value.

* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value\n *

agli inline fun <T: array, R> Array<out T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
} // Accumulates value starting with
public inline fun <T, R> Array<out T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R
{
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        index--
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @return Returns the specified [initial] value if the array is empty.
 */
public inline fun <R> IntArray.foldRight(initial: R, operation: (Int, acc: R) -> R): R
{
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @return Returns the specified [initial] value if the array is empty.
 */
public inline fun <R> CharArray.foldRight(initial: R, operation: (Char, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @return Returns the specified [initial] value if the array is empty.
 */
public inline fun <R> ShortArray.foldRight(initial: R, operation: (Short, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @return Returns the specified [initial] value if the array is empty.
 */
public inline fun <R> DoubleArray.foldRight(initial: R, operation: (Double, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @return Returns the specified [initial] value if the array is empty.
 */
public inline fun <R> FloatArray.foldRight(initial: R, operation: (Float, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @return Returns the specified [initial] value if the array is empty.
 */
public inline fun <R> BooleanArray.foldRight(initial: R, operation: (Boolean, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 * @return Returns the specified [initial] value if the array is empty.
 */
public inline fun <T, R> Array<out T>.foldRightIndexed(initial: R, operation: (index: Int, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from right to left
 * to each element and current accumulator value.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator
 * @return Returns the specified [initial] value if the array is empty.
 */
public inline fun <T> Array<out T>.foldRightIndexed(initial: T, operation: (index: Int, T, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
accumulator

/**
 * Performs the given [action] on each element.
 */
public inline fun <T> Array<T>.forEach(action: (T) -> Unit): Unit {
    forEachIndexed(action = action)
}

/**
 * Performs the given [action] on each element.
 */
public inline fun ByteArray.forEach(action: (Byte) -> Unit): Unit {
    forEachIndexed(action = action)
}

/**
 * Performs the given [action] on each element.
 */
public inline fun ShortArray.forEach(action: (Short) -> Unit): Unit {
    forEachIndexed(action = action)
}

/**
 * Performs the given [action] on each element.
 */
public inline fun IntArray.forEach(action: (Int) -> Unit): Unit {
    forEachIndexed(action = action)
}

/**
 * Performs the given [action] on each element.
 */
public inline fun LongArray.forEach(action: (Long) -> Unit): Unit {
    forEachIndexed(action = action)
}

/**
 * Performs the given [action] on each element.
 */
public inline fun FloatArray.forEach(action: (Float) -> Unit): Unit {
    forEachIndexed(action = action)
}

/**
 * Performs the given [action] on each element.
 */
public inline fun DoubleArray.forEach(action: (Double) -> Unit): Unit {
    forEachIndexed(action = action)
}

/**
 * Performs the given [action] on each element.
 */
public inline fun BooleanArray.forEach(action: (Boolean) -> Unit): Unit {
    forEachIndexed(action = action)
}

/**
 * Performs the given [action] on each element.
 */
public inline fun CharArray.forEach(action: (Char) -> Unit): Unit {
    forEachIndexed(action = action)
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun Array<Double>.max(): Double? {
    return maxOrNull()
}

Open Source Used In DNAC Platform Antelope 13713
ReplaceWith("maxOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4")\n@SinceKotlin("1.1")\npublic fun Array<out Float>.max(): Float? {\n    return maxOrNull()\n}\n
@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun ByteArray.max(): Byte? {\n    return maxOrNull()\n}\n
@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun ShortArray.max(): Short? {\n    return maxOrNull()\n}\n
@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun IntArray.max(): Int? {\n    return maxOrNull()\n}\n
@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun LongArray.max(): Long? {\n    return maxOrNull()\n}\n
@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun FloatArray.max(): Float? {\n    return maxOrNull()\n}\n
@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun DoubleArray.max(): Double? {\n    return maxOrNull()\n}\n
@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun CharArray.max(): Char? {\n    return maxOrNull()\n}\n
/**
 * Returns the first element yielding the largest value of the given function or
 * `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */\n@SinceKotlin("1.4")\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxBy(selector: (T) -> R): T? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R): Short? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> LongArray.maxBy(selector: (Long) -> R): Long? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R): Float? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) -> R): Double? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.maxBy(selector: (Boolean) -> R): Boolean? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> CharArray.maxBy(selector: (Char) -> R): Char? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R): Short? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> LongArray.maxBy(selector: (Long) -> R): Long? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R): Float? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) -> R): Double? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.maxBy(selector: (Boolean) -> R): Boolean? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> CharArray.maxBy(selector: (Char) -> R): Char? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R): Short? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> LongArray.maxBy(selector: (Long) -> R): Long? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R): Float? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) -> R): Double? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable<R>> BooleanArray.maxBy(selector: (Boolean) -> R): Boolean? {\n    return maxByOrNull(selector)\n}\n
@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic inline fun <R : Comparable>R Array<out T>.maxByOrNull(selector: (T) -> R): T? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex
== 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are
 * no elements.
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> ByteArray.maxByOrNull(selector: (Byte) -> R): Byte? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are
 * no elements.
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> ShortArray.maxByOrNull(selector: (Short) -> R): Short? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are
 * no elements.
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> IntArray.maxByOrNull(selector: (Int) -> R): Int? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are
 * no elements.
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> LongArray.maxByOrNull(selector: (Long) -> R): Long? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are
 * no elements.
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> FloatArray.maxByOrNull(selector: (Float) -> R): Float? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are
 * no elements.
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> DoubleArray.maxByOrNull(selector: (Double) -> R): Double? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are
 * no elements.
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> BooleanArray.maxByOrNull(selector: (Boolean) -> R): Boolean? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are
 * no elements.
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> CharArray.maxByOrNull(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val
maxElem = this[0]
val lastIndex = this.lastIndex
if (lastIndex == 0) return maxElem
var maxValue = selector(maxElem)
for (i in 1..lastIndex) {
    val e = this[i]
    val v = selector(e)
    if (maxValue < v) {
        maxElem = e
        maxValue = v
    }
}
return maxElem}

//*@SinceKotlin("1.4")*/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.maxOf(selector: (T) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is ` NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun ByteArray.maxOf(selector: (Byte) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is ` NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun ShortArray.maxOf(selector: (Short) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is ` NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun IntArray.maxOf(selector: (Int) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is ` NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun LongArray.maxOf(selector: (Long) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array.
 * If any of values produced by [selector] function is `NaN`, the returned result is ` NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun FloatArray.maxOf(selector: (Float) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@unpublic inline fun DoubleArray.maxOf(selector: (Double) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * For any values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @return the largest value among all values produced by [selector] function\n * applied to each element in the array.\n */
@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@unpublic inline fun CharArray.maxOf(selector: (Char) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * For any values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @return the largest value among all values produced by [selector] function\n * applied to each element in the array.\n */
@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@unpublic inline fun IntArray.maxOf(selector: (Int) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * For any values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @return the largest value among all values produced by [selector] function\n * applied to each element in the array.\n */
@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@unpublic inline fun ByteArray.maxOf(selector: (Byte) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * For any values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @return the largest value among all values produced by [selector] function\n * applied to each element in the array.\n */
@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@unpublic inline fun ShortArray.maxOf(selector: (Short) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * For any values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @return the largest value among all values produced by [selector] function\n * applied to each element in the array.\n */
@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@unpublic inline fun IntArray.maxOf(selector: (Int) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * For any values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @return the largest value among all values produced by [selector] function\n * applied to each element in the array.\n */
@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@unpublic inline fun IntArray.maxOf(selector: (Int) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * For any values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @return the largest value among all values produced by [selector] function\n * applied to each element in the array.\n */
@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@unpublic inline fun IntArray.maxOf(selector: (Int) -> Float): Float {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        maxValue = maxOf(maxValue, v)\n    }\n    return maxValue\n}\n/**
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * If any of values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * For any values produced by [selector] function is `NaN`, the returned result is
 * `NaN`.\n * @return the largest value among all values produced by [selector] function\n * applied to each element in the array.\n */
`NaN`.

@throws NoSuchElementException if the array is empty.

*\@Since Kotlin 1.4*
*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*
*\@OverloadResolution*
*\@LambdaReturnType*
*\@kotlin.internal.InlineOnly*

```
public inline fun LongArray.maxOf(selector: (Long) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

*\@Since Kotlin 1.4*
*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*
*\@OverloadResolution*
*\@LambdaReturnType*
*\@kotlin.internal.InlineOnly*

```
public inline fun FloatArray.maxOf(selector: (Float) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

*\@Since Kotlin 1.4*
*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*
*\@OverloadResolution*
*\@LambdaReturnType*
*\@kotlin.internal.InlineOnly*

```
public inline fun DoubleArray.maxOf(selector: (Double) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

*\@Since Kotlin 1.4*
*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*
*\@OverloadResolution*
*\@LambdaReturnType*
*\@kotlin.internal.InlineOnly*

```
public inline fun BooleanArray.maxOf(selector: (Boolean) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
```

*\@Since Kotlin 1.4*
*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*
*\@OverloadResolution*
*\@LambdaReturnType*
*\@kotlin.internal.InlineOnly*

```
public inline fun <T, R : Comparable<R>> Array<T>.maxOf(selector: (T) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}
```

*\@Since Kotlin 1.4*
*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*
*\@OverloadResolution*
*\@LambdaReturnType*
*\@kotlin.internal.InlineOnly*

```
public inline fun <R : Comparable<R>> ByteArray.maxOf(selector: (Byte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}
```

*\@Since Kotlin 1.4*
*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)*
*\@OverloadResolution*
*\@LambdaReturnType*
*\@kotlin.internal.InlineOnly*

```
ByteArray.maxOf(selector: (Byte) -> R) { if (isEmpty()) throw NoSuchElementException() } for (i in 1..lastIndex) {
    val v = selector(this[i])
    if (maxValue < v) {
        maxValue = v
    }
} return maxValue
```
NoSuchElementException if the array is empty.

ShortArray.maxOf(selector: (Short) -> R): R { return maxOf { selector(this[0]) } for (i in 1..lastIndex) { val v = selector(this[i]) if (maxValue < v) { maxValue = v } return maxValue } **} * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
IntArray.maxOf(selector: (Int) -> R): R { return maxOf { selector(this[0]) } for (i in 1..lastIndex) { val v = selector(this[i]) if (maxValue < v) { maxValue = v } return maxValue } **} * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
LongArray.maxOf(selector: (Long) -> R): R { return maxOf { selector(this[0]) } for (i in 1..lastIndex) { val v = selector(this[i]) if (maxValue < v) { maxValue = v } return maxValue } **} * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
FloatArray.maxOf(selector: (Float) -> R): R { return maxOf { selector(this[0]) } for (i in 1..lastIndex) { val v = selector(this[i]) if (maxValue < v) { maxValue = v } return maxValue } **} * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
DoubleArray.maxOf(selector: (Double) -> R): R { return maxOf { selector(this[0]) } for (i in 1..lastIndex) { val v = selector(this[i]) if (maxValue < v) { maxValue = v } return maxValue } **} * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
CharArray.maxOf(selector: (Char) -> R): R { return maxOf { selector(this[0]) } for (i in 1..lastIndex) { val v = selector(this[i]) if (maxValue < v) { maxValue = v } return maxValue } **} * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
BooleanArray.maxOf(selector: (Boolean) -> R): R { return maxOf { selector(this[0]) } for (i in 1..lastIndex) { val v = selector(this[i]) if (maxValue < v) { maxValue = v } return maxValue } **} * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n
Open Source Used In DNAC Platform Antelope 13719
produced by [selector] function

* applied to each element in the array or `null` if there are no elements.\n
* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

*\n
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array\n
<out T> .\n
maxOfOrNull\n
(selector: (T) -> Double) : Double? \n
{ \n
  if (isEmpty()) return null \n
  var maxV = selector\n
  (this[0]) \n
  for (i in 1..lastIndex) \n
  { \n
    val v = selector\n
    (this[i]) \n
    maxV = maxOf\n
    (maxV, v) \n
  } \n
  return maxV \n
  \n
  **\n
  * Returns the largest value among all values produced by [selector] function

  * applied to each element in the array or `null` if there are no elements.\n
  * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

*\n
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.

maxOfOrNull\n
(selector: (Byte) -> Double) : Double? \n
{ \n
  if (isEmpty()) return null \n
  var maxV = selector\n
  (this[0]) \n
  for (i in 1..lastIndex) \n
  { \n
    val v = selector\n
    (this[i]) \n
    maxV = maxOf\n
    (maxV, v) \n
  } \n
  return maxV \n
  \n
  **\n
  * Returns the largest value among all values produced by [selector] function

  * applied to each element in the array or `null` if there are no elements.\n
  * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

*\n
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.

maxOfOrNull\n
(selector: (Short) -> Double) : Double? \n
{ \n
  if (isEmpty()) return null \n
  var maxV = selector\n
  (this[0]) \n
  for (i in 1..lastIndex) \n
  { \n
    val v = selector\n
    (this[i]) \n
    maxV = maxOf\n
    (maxV, v) \n
  } \n
  return maxV \n
  \n
  **\n
  * Returns the largest value among all values produced by [selector] function

  * applied to each element in the array or `null` if there are no elements.\n
  * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

*\n
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.

maxOfOrNull\n
(selector: (Int) -> Double) : Double? \n
{ \n
  if (isEmpty()) return null \n
  var maxV = selector\n
  (this[0]) \n
  for (i in 1..lastIndex) \n
  { \n
    val v = selector\n
    (this[i]) \n
    maxV = maxOf\n
    (maxV, v) \n
  } \n
  return maxV \n
  \n
  **\n
  * Returns the largest value among all values produced by [selector] function

  * applied to each element in the array or `null` if there are no elements.\n
  * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

*\n
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.

maxOfOrNull\n
(selector: (Long) -> Double) : Double? \n
{ \n
  if (isEmpty()) return null \n
  var maxV = selector\n
  (this[0]) \n
  for (i in 1..lastIndex) \n
  { \n
    val v = selector\n
    (this[i]) \n
    maxV = maxOf\n
    (maxV, v) \n
  } \n
  return maxV \n
  \n
  **\n
  * Returns the largest value among all values produced by [selector] function

  * applied to each element in the array or `null` if there are no elements.\n
  * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
val v = selector(this[i])
maxValue = maxOf(maxValue, v)
}
return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun FloatArray.maxOfOrNull(selector: (Float) -> Float): Float? {
if (isEmpty()) return null
var maxValue = selector(this[0])
for (i in 1..lastIndex) {
val v = selector(this[i])
maxValue = maxOf(maxValue, v)
}
return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun DoubleArray.maxOfOrNull(selector: (Double) -> Float): Float? {
if (isEmpty()) return null
var maxValue = selector(this[0])
for (i in 1..lastIndex) {
val v = selector(this[i])
maxValue = maxOf(maxValue, v)
}
return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun BooleanArray.maxOfOrNull(selector: (Boolean) -> Float): Float? {
if (isEmpty()) return null
var maxValue = selector(this[0])
for (i in 1..lastIndex) {
val v = selector(this[i])
maxValue = maxOf(maxValue, v)
}
return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun CharArray.maxOfOrNull(selector: (Char) -> Float): Float? {
if (isEmpty()) return null
var maxValue = selector(this[0])
for (i in 1..lastIndex) {
val v = selector(this[i])
maxValue = maxOf(maxValue, v)
}
return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Array<out T>.maxOfOrNull(selector: (T) -> R): R? {
if (isEmpty()) return null
var maxValue = selector(this[0])
for (i in 1..lastIndex) {
val v = selector(this[i])
if (maxValue < v) {
maxValue = v
}
}
return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ByteArray.maxOfOrNull(selector: (Byte) -> R): R? {
if (isEmpty()) return null
var maxValue = selector(this[0])
for (i in 1..lastIndex) {
val v = selector(this[i])
if (maxValue < v) {
maxValue = v
}
}
return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ShortArray.maxOfOrNull(selector: (Short) -> R): R? {
if (isEmpty()) return null
var maxValue = selector(this[0])
for (i in 1..lastIndex) {
val v = selector(this[i])
if (maxValue < v) {
maxValue = v
}
}
return maxValue
}

by [selector] function

*\@SinceKotlin("1.4\")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun <R : Comparable<R>>
IntArray.maxOfOrNull(selector: (Int) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
*\@SinceKotlin("1.4\")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun <R : Comparable<R>>
LongArray.maxOfOrNull(selector: (Long) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
*\@SinceKotlin("1.4\")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun <R : Comparable<R>>
FloatArray.maxOfOrNull(selector: (Float) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
*\@SinceKotlin("1.4\")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun <R : Comparable<R>>
DoubleArray.maxOfOrNull(selector: (Double) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
*\@SinceKotlin("1.4\")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun <R : Comparable<R>>
BooleanArray.maxOfOrNull(selector: (Boolean) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator]
 * among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
*\@SinceKotlin("1.4\")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.internal.InlineOnly\@public inline fun <T, R> Array<out
T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}
Comparator<in R>, selector: (Byte) -> R): R

if (isEmpty()) throw NoSuchElementException()

val maxValue = selector(this[0])

for (i in 1..lastIndex) {
    if (comparator.compare(maxValue, v) < 0) {
        maxValue = v
    }
}

return maxValue

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.

Comparator<in R>, selector: (Short) -> R): R

if (isEmpty()) throw NoSuchElementException()

val maxValue = selector(this[0])

for (i in 1..lastIndex) {
    if (comparator.compare(maxValue, v) < 0) {
        maxValue = v
    }
}

return maxValue

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.

Comparator<in R>, selector: (Int) -> R): R

if (isEmpty()) throw NoSuchElementException()

val maxValue = selector(this[0])

for (i in 1..lastIndex) {
    if (comparator.compare(maxValue, v) < 0) {
        maxValue = v
    }
}

return maxValue

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.

Comparator<in R>, selector: (Long) -> R): R

if (isEmpty()) throw NoSuchElementException()

val maxValue = selector(this[0])

for (i in 1..lastIndex) {
    if (comparator.compare(maxValue, v) < 0) {
        maxValue = v
    }
}

return maxValue

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.

Comparator<in R>, selector: (Float) -> R): R

if (isEmpty()) throw NoSuchElementException()

val maxValue = selector(this[0])

for (i in 1..lastIndex) {
    if (comparator.compare(maxValue, v) < 0) {
        maxValue = v
    }
}

return maxValue

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.

Comparator<in R>, selector: (Double) -> R): R

if (isEmpty()) throw NoSuchElementException()

val maxValue = selector(this[0])

for (i in 1..lastIndex) {
    if (comparator.compare(maxValue, v) < 0) {
        maxValue = v
    }
}

return maxValue

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.

Comparator<in R>, selector: (Boolean) -> R): R

if (isEmpty()) throw NoSuchElementException()

val maxValue = selector(this[0])

for (i in 1..lastIndex) {
    if (comparator.compare(maxValue, v) < 0) {
        maxValue = v
    }
}

return maxValue

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.
By Lambda ReturnType \n@kotlin.internal.InlineOnly\n\npublic inline fun <R> CharArray.maxOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n
/**\n * Returns the largest value according to the provided [comparator] among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nBy Lambda ReturnType \n@kotlin.internal.InlineOnly\n\npublic inline fun <T, R> Array<out T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n
/**\n * Returns the largest value according to the provided [comparator] among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nBy Lambda ReturnType \n@kotlin.internal.InlineOnly\n\npublic inline fun <R> ByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n
/**\n * Returns the largest value according to the provided [comparator] among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nBy Lambda ReturnType \n@kotlin.internal.InlineOnly\n\npublic inline fun <R> ShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n
/**\n * Returns the largest value according to the provided [comparator] among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nBy Lambda ReturnType \n@kotlin.internal.InlineOnly\n\npublic inline fun <R> IntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Int) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n
/**\n * Returns the largest value according to the provided [comparator] among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nBy Lambda ReturnType \n@kotlin.internal.InlineOnly\n\npublic inline fun <R> LongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n
/**\n * Returns the largest value according to the provided [comparator] among all values produced by [selector]\n * function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nBy Lambda ReturnType \n@kotlin.internal.InlineOnly\n\npublic inline fun <R> FloatArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n
Open Source Used In DNAC Platform Antelope 13725
DoubleArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

BooleanArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.

CharArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.

Array<out Double>.maxOrNull(): Double? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        max = maxOf(max, e)
    }
    return max
}

Returns the largest element or `null` if there are no elements. * Returns the largest element or `null` if there are no elements.

Array<out Float>.maxOrNull(): Float? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        max = maxOf(max, e)
    }
    return max
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.

Array<out T : Comparable<T>>.maxOrNull(): T? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.

ByteArray.maxOrNull(): Byte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.

ShortArray.maxOrNull(): Short? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.

IntArray.maxOrNull(): Int? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.

LongArray.maxOrNull(): Long? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (max < e) max = e
    }
    return max
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.

FloatArray.maxOrNull(): Float? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        max = maxOf(max, e)
    }
    return max
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.

DoubleArray.maxOrNull(): Double? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        max = maxOf(max, e)
    }
    return max
}

Returns the largest element or `null` if there are no elements. * If any of elements is `NaN` returns `NaN`.
@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun <T> Array<out T>.maxWith(comparator: Comparator<in T>): T? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun ByteArray.maxWith(comparator: Comparator<in Byte>): Byte? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun ShortArray.maxWith(comparator: Comparator<in Short>): Short? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun IntArray.maxWith(comparator: Comparator<in Int>): Int? {
    return maxWithOrNull(comparator)
}

@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharArray.maxWith(comparator: Comparator<in Char>): Char? {
    return maxWithOrNull(comparator)
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
public fun <T> Array<out T>.maxWithOrNull(comparator: Comparator<in T>): T? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
public fun ByteArray.maxWithOrNull(comparator: Comparator<in Byte>): Byte? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
public fun ShortArray.maxWithOrNull(comparator: Comparator<in Short>): Short? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
public fun IntArray.maxWithOrNull(comparator: Comparator<in Int>): Int? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
public fun LongArray.maxWithOrNull(comparator: Comparator<in Long>): Long? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
public fun DoubleArray.maxWithOrNull(comparator: Comparator<in Double>): Double? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
public fun BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>): Boolean? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided [comparator] or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
public fun CharArray.maxWithOrNull(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}
public fun FloatArray.maxWithOrNull(comparator: Comparator<in Float>): Float? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided
 * comparator or `null` if there are no elements.
 */
public fun DoubleArray.maxWithOrNull(comparator: Comparator<in Double>): Double? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided
 * comparator or `null` if there are no elements.
 */
public fun BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>): Boolean? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

/**
 * Returns the first element having the largest value according to the provided
 * comparator or `null` if there are no elements.
 */
public fun CharArray.maxWithOrNull(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun ByteArray.min(): Byte? {
    return minOrNull()
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun ShortArray.min(): Short? {
    return minOrNull()
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun IntArray.min(): Int? {
    return minOrNull()
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun LongArray.min(): Long? {
    return minOrNull()
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun FloatArray.min(): Float? {
    return minOrNull()
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun DoubleArray.min(): Double? {
    return minOrNull()
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun CharArray.min(): Char? {
    return minOrNull()
}

@Deprecated("Use minByOrNull instead.",
ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <T : Comparable<T>, R : Comparable<R>> Array<out T>.minBy(selector: (T) -> R): T? {
    return minByOrNull(selector)
}

@Deprecated("Use minByOrNull instead.",
ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <R : Comparable<R>, Byte: Comparable<Byte>> ByteArray.minBy(selector: (Byte) -> R): Byte? {
    return minByOrNull(selector)
}

@Deprecated("Use minByOrNull instead.",
ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <R : Comparable<R>, Short: Comparable<Short>> ShortArray.minBy(selector: (Short) -> R): Short? {
    return minByOrNull(selector)
}

@Deprecated("Use minByOrNull instead.",
ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <R : Comparable<R>, Int: Comparable<Int>> IntArray.minBy(selector: (Int) -> R): Int? {
    return minByOrNull(selector)
}

@Deprecated("Use minByOrNull instead.",
ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <R : Comparable<R>, Long: Comparable<Long>> LongArray.minBy(selector: (Long) -> R): Long? {
    return minByOrNull(selector)
}

@Deprecated("Use minByOrNull instead.",
ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <R : Comparable<R>, Float: Comparable<Float>> FloatArray.minBy(selector: (Float) -> R): Float? {
    return minByOrNull(selector)
}

@Deprecated("Use minByOrNull instead.",
ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <R : Comparable<R>, Double: Comparable<Double>> DoubleArray.minBy(selector: (Double) -> R): Double? {
    return minByOrNull(selector)
}

@Deprecated("Use minByOrNull instead.",
ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <R : Comparable<R>, Char: Comparable<Char>> CharArray.minBy(selector: (Char) -> R): Char? {
    return minByOrNull(selector)
}
Comparable<R>> LongArray.minBy(selector: (Long) -> R): Long?

{
    return minByOrNonNull(selector)[n]@Deprecated("Use minByOrNonNull instead.",
    ReplaceWith("minByOrNull(selector)"))@DeprecatedSinceKotlin(warningSince = "1.4")
    } public inline fun <R : Comparable<R>> FloatArray.minBy(selector: (Float) -> R): Float?

{
    return minByOrNonNull(selector)[n]@Deprecated("Use minByOrNonNull instead.",
    ReplaceWith("minByOrNull(selector)"))@DeprecatedSinceKotlin(warningSince = "1.4")
    } public inline fun <R : Comparable<R>> DoubleArray.minBy(selector: (Double) -> R): Double?

{
    return minByOrNonNull(selector)[n]@Deprecated("Use minByOrNonNull instead.",
    ReplaceWith("minByOrNull(selector)"))@DeprecatedSinceKotlin(warningSince = "1.4")
    } public inline fun <R : Comparable<R>> BooleanArray.minBy(selector: (Boolean) -> R): Boolean?

{
    return minByOrNonNull(selector)[n]@Deprecated("Use minByOrNonNull instead.",
    ReplaceWith("minByOrNull(selector)"))@DeprecatedSinceKotlin(warningSince = "1.4")
    } public inline fun <R : Comparable<R>> CharArray.minBy(selector: (Char) -> R): Char?

{
    return minByOrNonNull(selector)[n]@Deprecated("Use minByOrNonNull instead.",
    ReplaceWith("minByOrNull(selector)"))@DeprecatedSinceKotlin(warningSince = "1.4")
    } public inline fun <T, R : Comparable<R>> Array<out T>.minByOrNull(selector: (T) -> R): T?

{
    if (isEmpty()) return null
    var minElem = this[0] val lastIndex = this.lastIndex if (lastIndex == 0) return minElem
    var minVal = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i] val v = selector(e) if (minVal > v) {
            minElem = e
            minVal = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or null if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.minByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> ByteArray.minByOrNull(selector: (Byte) -> R): Byte?

{
    if (isEmpty()) return null
    var minElem = this[0] val lastIndex = this.lastIndex if (lastIndex == 0) return minElem
    var minVal = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i] val v = selector(e) if (minVal > v) {
            minElem = e
            minVal = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or null if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.minByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> ShortArray.minByOrNull(selector: (Short) -> R): Short?

{
    if (isEmpty()) return null
    var minElem = this[0] val lastIndex = this.lastIndex if (lastIndex == 0) return minElem
    var minVal = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i] val v = selector(e) if (minVal > v) {
            minElem = e
            minVal = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or null if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.minByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> IntArray.minByOrNull(selector: (Int) -> R): Int?

{
    if (isEmpty()) return null
    var minElem = this[0] val lastIndex = this.lastIndex if (lastIndex == 0) return minElem
    var minVal = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i] val v = selector(e) if (minVal > v) {
            minElem = e
            minVal = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or null if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.minByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> LongArray.minByOrNull(selector: (Long) -> R): Long?

{
    if (isEmpty()) return null
    var minElem = this[0] val lastIndex = this.lastIndex if (lastIndex == 0) return minElem
    var minVal = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i] val v = selector(e) if (minVal > v) {
            minElem = e
            minVal = v
        }
    }
    return minElem
}

/**
 * Returns the first element yielding the smallest value of the given function or null if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.minByOrNull
 */
@SinceKotlin("1.4")
public inline fun <R : Comparable<R>> FloatArray.minByOrNull(selector: (Float) -> R): Float?

{
    if (isEmpty()) return null
    var minElem = this[0] val lastIndex = this.lastIndex if (lastIndex == 0) return minElem
    var minVal = selector(minElem)
    for (i in 1..lastIndex) {
        val e = this[i] val v = selector(e) if (minVal > v)
{\n  minElem = e\n  minValue = v\n  \n  return minElem\n}\n*/\n* Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n* @sample\n\nCollections.Aggregates.minByOrNull\n*\nAn@SinceKotlin("1.4")\n\npublic inline fun <R : Comparable<R>> DoubleArray.minByOrNull(selector: (Double) -> R): Double? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}*/\n* Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n* @sample\n\nCollections.Aggregates.minByOrNull\n*\nAn@SinceKotlin("1.4")\n\npublic inline fun <R : Comparable<R>> BooleanArray.minByOrNull(selector: (Boolean) -> R): Boolean? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}*/\n* Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n* @sample\n\nCollections.Aggregates.minByOrNull\n*\nAn@SinceKotlin("1.4")\n\npublic inline fun <R : Comparable<R>> CharArray.minByOrNull(selector: (Char) -> R): Char? {\n    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}*/\n* Returns the smallest value among all values produced by [selector] function\n* applied to each element in the array.\n* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n* \n  * \n  * @throws NoSuchElementException if the array is empty.\n*\n* An@SinceKotlin("1.4")\n* OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n* OverloadResolution\n* ByLambdaReturnType\n* kotlin.internal.InlineOnly\n* public inline fun <T> Array<out T>.minOf(selector: (T) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}*/\n* Returns the smallest value among all values produced by [selector] function\n* applied to each element in the array.\n* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n* \n  * \n  * @throws NoSuchElementException if the array is empty.\n*\n* An@SinceKotlin("1.4")\n* OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n* OverloadResolution\n* ByLambdaReturnType\n* kotlin.internal.InlineOnly\n* public inline fun ByteArray.minOf(selector: (Byte) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}*/\n* Returns the smallest value among all values produced by [selector] function\n* applied to each element in the array.\n* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n* \n  * \n  * @throws NoSuchElementException if the array is empty.\n*\n* An@SinceKotlin("1.4")\n* OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n* OverloadResolution\n* ByLambdaReturnType\n* kotlin.internal.InlineOnly\n* public inline fun ShortArray.minOf(selector: (Short) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}*/\n* Returns the smallest value among all values produced by [selector] function\n* applied to each element in the array.\n* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n* \n  * \n  * @throws NoSuchElementException if the array is empty.\n*\n* An@SinceKotlin("1.4")\n* OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n* OverloadResolution\n* ByLambdaReturnType\n* kotlin.internal.InlineOnly\n* public inline fun IntArray.minOf(selector: (Int) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}*/\n* Returns the smallest value among all values produced by [selector] function\n* applied to each element in the array.\n* If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n* \n  * \n  * @throws NoSuchElementException if the array is empty.
each element in the array.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

* @SinceKotlin("1.4")
* @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
* @OverloadResolution
* @LambdaReturnType
* @kotlin.internal.InlineOnly
* public inline fun LongArray.minOf(selector: (Long) -> Double): Double {

    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

* @SinceKotlin("1.4")
* @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
* @OverloadResolution
* @LambdaReturnType
* @kotlin.internal.InlineOnly
* public inline fun FloatArray.minOf(selector: (Float) -> Double): Double {

    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

* @SinceKotlin("1.4")
* @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
* @OverloadResolution
* @LambdaReturnType
* @kotlin.internal.InlineOnly
* public inline fun DoubleArray.minOf(selector: (Double) -> Double): Double {

    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

* @SinceKotlin("1.4")
* @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
* @OverloadResolution
* @LambdaReturnType
* @kotlin.internal.InlineOnly
* public inline fun BooleanArray.minOf(selector: (Boolean) -> Double): Double {

    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

* @SinceKotlin("1.4")
* @OptIn(kotlin.experimental.ExperimentalTypeInference::class)
* @OverloadResolution
* @LambdaReturnType
* @kotlin.internal.InlineOnly
* public inline fun <T> Array<out T>.minOf(selector: (T) -> Float): Float {

    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}
Returns the smallest value among all values produced by [selector] function applied to each element in the array.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.

If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
Open Source Used In DNAC Platform Antelope 13733

```kotlin
1..lastIndex) {
    val v = selector(this[i])
    minValue = minOf(minValue, v)
} return

minValue\n\n\n
/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\n\npublic inline fun <T, R> Array<T>.minOf(selector: (T) -> R) R {\n    if (isEmpty()) throw NoSuchElementException()\n    var min\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) \n            minValue = v\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\n\npublic inline fun <R> ByteArray.minOf(selector: (Byte) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) \n            minValue = v\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\n\npublic inline fun <R> ShortArray.minOf(selector: (Short) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) \n            minValue = v\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\n\npublic inline fun <R> IntArray.minOf(selector: (Int) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) \n            minValue = v\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\n\npublic inline fun <R> LongArray.minOf(selector: (Long) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) \n            minValue = v\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\n\npublic inline fun <R> FloatArray.minOf(selector: (Float) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) \n            minValue = v\n    }\n    return minValue\n}

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly\n\npublic inline fun <R> DoubleArray.minOf(selector: (Double) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (minValue > v) \n            minValue = v\n    }\n    return minValue\n}
```kotlin
val minValue = selector(this[0])
for (i in 1..lastIndex) {
    val v = selector(this[i])
    if (minValue > v) {
        minValue = v
    }
}
return minValue

/**
 * Returns the smallest value among all values
 * produced by [selector] function
 * applied to each element in the array.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
public inline fun <R : Comparable<R>> BooleanArray.minOf(selector: (Boolean) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function
 * applied to each element in the array.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
public inline fun <R : Comparable<R>> CharArray.minOf(selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.minOfOrNull(selector: (T) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun ByteArray.minOfOrNull(selector: (Byte) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun ShortArray.minOfOrNull(selector: (Short) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun IntArray.minOfOrNull(selector: (Int) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values
 * produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun LongArray.minOfOrNull(selector: (Long) ->
> Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *
 */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType@n@kotlin.experimental.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *
 */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType@n@kotlin.experimental.InlineOnly\npublic inline fun DoubleArray.minOfOrNull(selector: (Double) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *
 */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType@n@kotlin.experimental.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector: (Boolean) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *
 */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType@n@kotlin.experimental.InlineOnly\npublic inline fun CharArray.minOfOrNull(selector: (Char) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *
 */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType@n@kotlin.experimental.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector: (Byte) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function \n * applied to each element in the array or `null` if there are no elements.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *
 */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType@n@kotlin.experimental.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector: (T) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
ByLambdaReturnType\n@kotlin.internal:inline fun ShortArray.minOfOrNull(selector: (Short) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal:inline fun IntArray.minOfOrNull(selector: (Int) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal:inline fun LongArray.minOfOrNull(selector: (Long) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal:inline fun FloatArray.minOfOrNull(selector: (Float) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal:inline fun DoubleArray.minOfOrNull(selector: (Double) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal:inline fun BooleanArray.minOfOrNull(selector: (Boolean) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
*\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal:inline fun CharArray.minOfOrNull(selector: (Char) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n
Open Source Used in DNAC Platform Antelope 13736
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <T : Comparable<T>> Array<out T>.minOfOrNull(selector: (T) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function*
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ByteArray.minOfOrNull(selector: (Byte) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function*
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ShortArray.minOfOrNull(selector: (Short) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function*
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> IntArray.minOfOrNull(selector: (Int) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function*
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> LongArray.minOfOrNull(selector: (Long) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function*
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> FloatArray.minOfOrNull(selector: (Float) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function*
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> DoubleArray.minOfOrNull(selector: (Double) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function*
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> BooleanArray.minOfOrNull(selector: (Boolean) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>>
CharArray.minOfOrNull(selector: (Char) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <T, R> Array<T>.minOfWith(comparator: Comparator<R>, selector: (T) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.minOfWith(comparator: Comparator<R>, selector: (Byte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> ShortArray.minOfWith(comparator: Comparator<R>, selector: (Short) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> IntArray.minOfWith(comparator: Comparator<R>, selector: (Int) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> LongArray.minOfWith(comparator: Comparator<R>, selector: (Long) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <R> FloatArray.minOfWith(comparator: Comparator<R>, selector: (Float) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.

*Throws NoSuchElementException if the array is empty.

Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.

*Throws NoSuchElementException if the array is empty.
function applied to each element in the array.*n * @throws NoSuchElementException if the array is empty.*n *\n *\n *@SinceKotlin("1.4")\n *@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n *@OverloadResolution\n *@LambdaReturnType\n *@InlineOnly\n *public inline fun <R> DoubleArray.minOfWith(comparator: Comparator<R>, selector: (Double) -> R): R {\n *\n * if (isEmpty()) throw NoSuchElementException()\n * var minValue = selector(this[0])\n * for (i in 1..lastIndex) {\n * val v = selector(this[i])\n * if (comparator.compare(minValue, v) > 0) {\n * minValue = v\n * }\n * return minValue\n * }\n *\n *\n * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array.*n *\n *\n *@SinceKotlin("1.4")\n *@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n *@OverloadResolution\n *@LambdaReturnType\n *@InlineOnly\n *public inline fun <R> BooleanArray.minOfWith(comparator: Comparator<R>, selector: (Boolean) -> R): R {\n *\n * if (isEmpty()) throw NoSuchElementException()\n * var minValue = selector(this[0])\n * for (i in 1..lastIndex) {\n * val v = selector(this[i])\n * if (comparator.compare(minValue, v) > 0) {\n * minValue = v\n * }\n * return minValue\n * }\n *\n *\n * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.*n *\n *\n *@SinceKotlin("1.4")\n *@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n *@OverloadResolution\n *@LambdaReturnType\n *@InlineOnly\n *public inline fun <T, R> Array<out T>.minOfWithOrNull(comparator: Comparator<R>, selector: (T) -> R): R? {\n *\n * return null\n * var minValue = selector(this[0])\n * for (i in 1..lastIndex) {\n * val v = selector(this[i])\n * if (comparator.compare(minValue, v) > 0) {\n * minValue = v\n * }\n * return minValue\n * }\n *\n *\n * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.*n *\n *\n *@SinceKotlin("1.4")\n *@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n *@OverloadResolution\n *@LambdaReturnType\n *@InlineOnly\n *public inline fun <R> ByteArray.minOfWithOrNull(comparator: Comparator<R>, selector: (Byte) -> R): R? {\n *\n * return null\n * var minValue = selector(this[0])\n * for (i in 1..lastIndex) {\n * val v = selector(this[i])\n * if (comparator.compare(minValue, v) > 0) {\n * minValue = v\n * }\n * return minValue\n * }\n *\n *\n * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.*n *\n *\n *@SinceKotlin("1.4")\n *@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n *@OverloadResolution\n *@LambdaReturnType\n *@InlineOnly\n *public inline fun <R> ShortArray.minOfWithOrNull(comparator: Comparator<R>, selector: (Short) -> R): R? {\n *\n * return null\n * var minValue = selector(this[0])\n * for (i in 1..lastIndex) {\n * val v = selector(this[i])\n * if (comparator.compare(minValue, v) > 0) {\n * minValue = v\n * }\n * return minValue\n * }\n *\n *\n * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.*n *\n *\n *@SinceKotlin("1.4")\n *@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n *@OverloadResolution\n *@LambdaReturnType\n *@InlineOnly\n *public inline fun <R> IntArray.minOfWithOrNull(comparator: Comparator<R>, selector: (Int) -> R): R? {\n *\n * return null\n * var minValue = selector(this[0])\n * for (i in 1..lastIndex) {\n * val v = selector(this[i])\n * if (comparator.compare(minValue, v) > 0) {\n * minValue = v\n * }\n * return minValue\n * }\n *\n *\n * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.*n
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nLongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nFloatArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nDoubleArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nBooleanArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R>\nCharArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n    if (isEmpty()) return null\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(minValue, v) > 0) {\n            minValue = v\n        }\n    }\n    return minValue\n}\n
Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic fun Array<out Double>.minOrNull(): Double? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n
Returns the smallest element or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic fun Array<out Float>.minOrNull(): Float? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n
Returns the smallest element or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic fun <T : Comparable<T>> Array<out T>.minOrNull(): T? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n
Returns the smallest element or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic fun Array<out Double>.minOrNull(): Double? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        min = minOf(min, e)\n    }\n    return min\n}\n
Returns the smallest element or ‘null’ if there are no elements.\n
`@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic fun ByteArray.minOrNull(): Byte? {\n    if (isEmpty()) return null\n    var min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n
Returns the smallest element or ‘null’ if there are no elements.
ShortArray.minOrNull(): Short? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (min > e) min = e\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun IntArray.minOrNull(): Int? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (min > e) min = e\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun LongArray.minOrNull(): Long? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (min > e) min = e\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun FloatArray.minOrNull(): Float? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun DoubleArray.minOrNull(): Double? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun CharArray.minOrNull(): Char? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (min > e) min = e\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun ByteArray.minOrNull(): Byte? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun ShortArray.minOrNull(): Short? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun IntArray.minOrNull(): Int? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun LongArray.minOrNull(): Long? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun FloatArray.minOrNull(): Float? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun DoubleArray.minOrNull(): Double? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun BooleanArray.minOrNull(): Boolean? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (min > e) min = e\n    } return min\n}\n
*\n@SinceKotlin("1.4")\npublic fun CharArray.minOrNull(): Char? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (min > e) min = e\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun <T> Array<out T>.minWith(comparator: Comparator<in T>): T? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (comparator.compare(min, e) > 0) min = e\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun ByteArray.minWith(comparator: Comparator<in Byte>): Byte? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun ShortArray.minWith(comparator: Comparator<in Short>): Short? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun IntArray.minWith(comparator: Comparator<in Int>): Int? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun LongArray.minWith(comparator: Comparator<in Long>): Long? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun FloatArray.minWith(comparator: Comparator<in Float>): Float? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun DoubleArray.minWith(comparator: Comparator<in Double>): Double? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] min = minOf(min, e)\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun BooleanArray.minWith(comparator: Comparator<in Boolean>): Boolean? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (min > e) min = e\n    } return min\n}\n
@Deprecated("Use minWithOrNull instead.", \n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun CharArray.minWith(comparator: Comparator<in Char>): Char? {\n    if (isEmpty()) return null\n    var min = this[0] for (i in 1..lastIndex) {\n        val e = this[i] if (comparator.compare(min, e) > 0) min = e\n    } return min\n}\n
Open Source Used In DNAC Platform Antelope 13741
`null` if there are no elements.

```kotlin
@SinceKotlin("1.4")
public fun ByteArray.minWithOrNull(comparator: Comparator<in Byte>): Byte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun ShortArray.minWithOrNull(comparator: Comparator<in Short>): Short? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun IntArray.minWithOrNull(comparator: Comparator<in Int>): Int? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun LongArray.minWithOrNull(comparator: Comparator<in Long>): Long? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun FloatArray.minWithOrNull(comparator: Comparator<in Float>): Float? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun DoubleArray.minWithOrNull(comparator: Comparator<in Double>): Double? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun BooleanArray.minWithOrNull(comparator: Comparator<in Boolean>): Boolean? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun CharArray.minWithOrNull(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

@SinceKotlin("1.4")
public fun ShortArray.none(): Boolean {
    return isEmpty()
}

@SinceKotlin("1.4")
public fun IntArray.none(): Boolean {
    return isEmpty()
}

@SinceKotlin("1.4")
public fun LongArray.none(): Boolean {
    return isEmpty()
}

@SinceKotlin("1.4")
public fun FloatArray.none(): Boolean {
    return isEmpty()
}

@SinceKotlin("1.4")
public fun DoubleArray.none(): Boolean {
    return isEmpty()
}

@SinceKotlin("1.4")
public fun BooleanArray.none(): Boolean {
    return isEmpty()
}

@SinceKotlin("1.4")
public fun CharArray.none(): Boolean {
    return isEmpty()
}
```

Open Source Used In DNAC Platform Antelope 13742
isEmpty() * Returns `true` if no elements match the given [predicate].

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun <T> Array<out T>.none(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun ByteArray.none(predicate: (Byte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun ShortArray.none(predicate: (Short) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun IntArray.none(predicate: (Int) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun LongArray.none(predicate: (Long) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun FloatArray.none(predicate: (Float) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun DoubleArray.none(predicate: (Double) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 * 
 * @sample samples.collections.Aggregates.noneWithPredicate
 */
public inline fun CharArray.none(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.onEach(action: (T) -> Unit): Array<out T> {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ByteArray.onEach(action: (Byte) -> Unit): ByteArray {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ShortArray.onEach(action: (Short) -> Unit): ShortArray {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun IntArray.onEach(action: (Int) -> Unit): IntArray {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun LongArray.onEach(action: (Long) -> Unit): LongArray {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun FloatArray.onEach(action: (Float) -> Unit): FloatArray {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.onEach(action: (Double) -> Unit): DoubleArray {
    return apply { for (element in this) action(element) }
}
[action] on each element and returns the array itself afterwards.

*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun BooleanArray.onEach(action: (Boolean) -> Unit): BooleanArray {
    return apply { for (element in this) action(element) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun CharArray.onEach(action: (Char) -> Unit): CharArray {
    return apply { for (element in this) action(element) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun ByteArray.onEachIndexed(action: (Int, Byte) -> Unit): ByteArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun ShortArray.onEachIndexed(action: (Int, Short) -> Unit): ShortArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun IntArray.onEachIndexed(action: (Int, Int) -> Unit): IntArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun LongArray.onEachIndexed(action: (Int, Long) -> Unit): LongArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun FloatArray.onEachIndexed(action: (Int, Float) -> Unit): FloatArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun DoubleArray.onEachIndexed(action: (Int, Double) -> Unit): DoubleArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun BooleanArray.onEachIndexed(action: (Int, Boolean) -> Unit): BooleanArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun CharArray.onEachIndexed(action: (Int, Char) -> Unit): CharArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
\n
Open Source Used In DNAC Platform Antelope 13744

*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun BooleanArray.onEach(action: (Boolean) -> Unit): BooleanArray {
    return apply { for (element in this) action(element) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun CharArray.onEach(action: (Char) -> Unit): CharArray {
    return apply { for (element in this) action(element) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun ByteArray.onEachIndexed(action: (Int, Byte) -> Unit): ByteArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun IntArray.onEachIndexed(action: (Int, Int) -> Unit): IntArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun LongArray.onEachIndexed(action: (Int, Long) -> Unit): LongArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun FloatArray.onEachIndexed(action: (Int, Float) -> Unit): FloatArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun DoubleArray.onEachIndexed(action: (Int, Double) -> Unit): DoubleArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun BooleanArray.onEachIndexed(action: (Int, Boolean) -> Unit): BooleanArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun CharArray.onEachIndexed(action: (Int, Char) -> Unit): CharArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
\n
Open Source Used In DNAC Platform Antelope 13744

*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun BooleanArray.onEach(action: (Boolean) -> Unit): BooleanArray {
    return apply { for (element in this) action(element) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\npublic inline fun CharArray.onEach(action: (Char) -> Unit): CharArray {
    return apply { for (element in this) action(element) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun ByteArray.onEachIndexed(action: (Int, Byte) -> Unit): ByteArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun IntArray.onEachIndexed(action: (Int, Int) -> Unit): IntArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun LongArray.onEachIndexed(action: (Int, Long) -> Unit): LongArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun FloatArray.onEachIndexed(action: (Int, Float) -> Unit): FloatArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun DoubleArray.onEachIndexed(action: (Int, Double) -> Unit): DoubleArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun BooleanArray.onEachIndexed(action: (Int, Boolean) -> Unit): BooleanArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}

\n
*\@SinceKotlin("1.4")*\n@kotlin.internal.InlineOnly\n\ninline fun CharArray.onEachIndexed(action: (Int, Char) -> Unit): CharArray {
    return apply { for (index in 0 until this.size) action(index, this[index]) }\n}
receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
public inline fun <S, T : S> Array<out T>.reduce(operation: (acc: S, T) -> S): S {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator: S = this[0] for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n
Accumulates value starting with the first element and applying [operation] from left to right.\n * \n * Thrown if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
ByteArray.reduce(operation: (acc: Byte, Byte) -> Byte): Byte {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0] for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n
ShortArray.reduce(operation: (acc: Short, Short) -> Short): Short {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0] for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n
IntArray.reduce(operation: (acc: Int, Int) -> Int): Int {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0] for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n
LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0] for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n
FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {\n    if (isEmpty())\n        throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = this[0] for (index in 1..lastIndex) {\n        accumulator = operation(accumulator, this[index])\n    }\n    return accumulator\n}\n
UnsupportedOperationException("Empty array can't be reduced.")
@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
public inline fun DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

```kotlin
public inline fun BooleanArray.reduce(operation: (acc: Boolean, Boolean) -> Boolean): Boolean {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

```kotlin
public inline fun CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

```kotlin
public inline fun ByteArray.reduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. Throws an exception if this array is empty. If the array can be empty in an expected way, please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

```kotlin
public inline fun ShortArray.reduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): Short {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```
accumulator

Accumulates value starting with the first element and applying [operation] from left to right. It returns `null` if its receiver is empty. Instead, it returns `null` when its receiver is empty.

@sample samples.collections.Collections.Aggregates.reduce

```
public inline fun IntArray.reduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): Int {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

```
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

```
public inline fun FloatArray.reduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): Float {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

```
public inline fun DoubleArray.reduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): Double {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

```
public inline fun BooleanArray.reduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```
reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.

```kotlin
fun CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

It returns `null` when its receiver is empty.

```kotlin
fun <S, T : S> Array<out T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {
    if (isEmpty())
        return null
    var accumulator: S = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

It returns `null` when its receiver is empty.

```kotlin
fun ByteArray.reduceIndexedOrNull(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

It returns `null` when its receiver is empty.

```kotlin
fun ShortArray.reduceIndexedOrNull(operation: (index: Int, acc: Short, Short) -> Short): Short? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

It returns `null` when its receiver is empty.

```kotlin
fun IntArray.reduceIndexedOrNull(operation: (index: Int, acc: Int, Int) -> Int): Int? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

It returns `null` when its receiver is empty.

```kotlin
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}
```

It returns `null` when its receiver is empty.
FloatArray.reduceIndexedOrNull(operation: (index: Int, acc: Float, Float) -> Float): Float? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * 
 * @param samples.collections.Collections.Aggregates.reduceIndexedOrNull
 */
@SinceKotlin("1.4")
public inline fun DoubleArray.reduceIndexedOrNull(operation: (index: Int, acc: Double, Double) -> Double): Double? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * 
 * @param samples.collections.Collections.Aggregates.reduceIndexedOrNull
 */
@SinceKotlin("1.4")
public inline fun BooleanArray.reduceIndexedOrNull(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original array.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * 
 * @param samples.collections.Collections.Aggregates.reduceIndexedOrNull
 */
@SinceKotlin("1.4")
public inline fun CharArray.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(index, accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * 
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * 
 * @param samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> Array<out T>.reduceOrNull(operation: (acc: S, T) -> S): S? {
    if (isEmpty())
        return null
    var accumulator: S = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * 
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * 
 * @param samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun ByteArray.reduceOrNull(operation: (acc: Byte, Byte) -> Byte): Byte? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * 
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * 
 * @param samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun ShortArray.reduceOrNull(operation: (acc: Short, Short) -> Short): Short? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun IntArray.reduceOrNull(operation: (acc: Int, Int) -> Int): Int? {
    var accumulator = this[0] for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

@sample samples.collections.Collections.Aggregates.reduceOrNull

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
    var accumulator = this[0] for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

@sample samples.collections.Collections.Aggregates.reduceOrNull

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun FloatArray.reduceOrNull(operation: (acc: Float, Float) -> Float): Float? {
    var accumulator = this[0] for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

@sample samples.collections.Collections.Aggregates.reduceOrNull

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun DoubleArray.reduceOrNull(operation: (acc: Double, Double) -> Double): Double? {
    var accumulator = this[0] for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

@sample samples.collections.Collections.Aggregates.reduceOrNull

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun BooleanArray.reduceOrNull(operation: (acc: Boolean, Boolean) -> Boolean): Boolean? {
    var accumulator = this[0] for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

@sample samples.collections.Collections.Aggregates.reduceOrNull

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun CharArray.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {
    var accumulator = this[0] for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

@sample samples.collections.Collections.Aggregates.reduceOrNull

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> Array<out
T>.reduceRight(operation: (T, acc: S) -> S): S {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator: S = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
        index--
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value,\n and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun ByteArray.reduceRight(operation: (Byte, acc: Byte) -> Byte): Byte {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value,\n and calculates the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun ShortArray.reduceRight(operation: (Short, acc: Short) -> Short): Short {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value,\n and calculates the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun IntArray.reduceRight(operation: (Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value,\n and calculates the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun LongArray.reduceRight(operation: (Long, acc: Long) -> Long): Long {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value.
 * @param [operation] function that takes an element and current accumulator value,\n and calculates the next accumulator value.\n * @sample samples.collections.Collections.Aggregates.reduceRight
 */
public inline fun FloatArray.reduceRight(operation: (Float, acc: Float) -> Float): Float {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
@param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value,\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\nv

fun DoubleArray.reduceRight(operation: (Double, acc: Double) -> Double): Double {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))\n    }
    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value,\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value,\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */
public inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)\n    }
    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value,\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value,\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */
public inline fun CharArray.reduceRight(operation: (Char, acc: Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)\n    }
    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value,\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */
public inline fun <S, T : S> Array<out T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator: S = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)\n    }
    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value,\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */
public inline fun ByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")\n    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)\n    }
    return accumulator\n}
ShortArray.reduceRightIndexed(operation: (index: Int, Short, acc: Short) -> Short): Short

var index = lastIndex
if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
var accumulator = get(index--)
while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.

* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNonNull] instead. It returns `null` when its receiver is empty.

* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.

* This method is public inline fun

IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int

while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.

* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNonNull] instead. It returns `null` when its receiver is empty.

* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.

* This method is public inline fun


while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.

* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNonNull] instead. It returns `null` when its receiver is empty.

* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.

* This method is public inline fun

FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float

while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.

* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNonNull] instead. It returns `null` when its receiver is empty.

* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.

* This method is public inline fun

DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double

while (index >= 0) {
    accumulator = operation(index, get(index), accumulator)
    return accumulator
}

* Accumulates value starting with the last element and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.

* Throws an exception if this array is empty. If the array can be empty in an expected way,
* please use [reduceRightIndexedOrNonNull] instead. It returns `null` when its receiver is empty.

* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
* and calculates the next accumulator value.

* This method is public inline fun
samples.collections.Collections.Aggregates.reduceRightOrNull *\public inline fun
BooleanArray.reduceRightIndexedOrNull(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// * Throws an exception if this array is empty. If the array can be empty in an expected way,
// * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
// * @param [operation] function that takes the index of an element, the element itself and current accumulator
// * value, and calculates the next accumulator value.

CharArray.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// Returns `null` if the array is empty.
// @param [operation] function that takes the index of an element, the element itself and current accumulator value.
// * Calculates the next accumulator value.
ByteArray.reduceRightIndexedOrNull(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// Returns `null` if the array is empty.
// @param [operation] function that takes the index of an element, the element itself and current accumulator value.
// * Calculates the next accumulator value.
ShortArray.reduceRightIndexedOrNull(operation: (index: Int, Short, acc: Short) -> Short): Short {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// Returns `null` if the array is empty.
// @param [operation] function that takes the index of an element, the element itself and current accumulator value.
// * Calculates the next accumulator value.
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// Returns `null` if the array is empty.
// @param [operation] function that takes the index of an element, the element itself and current accumulator value.
// * Calculates the next accumulator value.
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// Returns `null` if the array is empty.
// @param [operation] function that takes the index of an element, the element itself and current accumulator value.
// * Calculates the next accumulator value.
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// Returns `null` if the array is empty.
// @param [operation] function that takes the index of an element, the element itself and current accumulator value.
// * Calculates the next accumulator value.
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// Returns `null` if the array is empty.
// @param [operation] function that takes the index of an element, the element itself and current accumulator value.
// * Calculates the next accumulator value.
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

// Accumulates value starting with the last element and applying [operation] from right to left
// to each element with its index in the original array and current accumulator value.
// Returns `null` if the array is empty.
// @param [operation] function that takes the index of an element, the element itself and current accumulator value.
// * Calculates the next accumulator value.
IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
[operation] function that takes the index of an element, the element itself and current accumulator value,\n* and calculates the next accumulator value.\n* \n* @sample
samples.collections.Collections.Aggregates.reduceRightOrNullable */@SinceKotlin("1.4")*/

public inline fun
LongArray.reduceRightIndexedOrNullable(operation: (index: Int, Long, acc: Long) -> Long): Long? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n    }\n    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n* \n* @param \[operation\] function that takes the index of an element, the element itself and current accumulator value,\n* \n* @sample
samples.collections.Collections.Aggregates.reduceRightIndexedOrNullable */@SinceKotlin("1.4")*/

public inline fun
FloatArray.reduceRightIndexedOrNullable(operation: (index: Int, Float, acc: Float) -> Float): Float? {
    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n    }\n    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying \[operation\] from right to left\n * to each element with its index in the original array and current accumulator value.\n* \n* @param \[operation\] function that takes the index of an element, the element itself and current accumulator value,\n* \n* @sample
samples.collections.Collections.Aggregates.reduceRightIndexedOrNullable */@SinceKotlin("1.4")*/

public inline fun
DoubleArray.reduceRightIndexedOrNullable(operation: (index: Int, Double, acc: Double) -> Double): Double? {
    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n    }\n    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying \[operation\] from right to left\n * to each element with its index in the original array and current accumulator value.\n* \n* @param \[operation\] function that takes the index of an element, the element itself and current accumulator value,\n* \n* @sample
samples.collections.Collections.Aggregates.reduceRightIndexedOrNullable */@SinceKotlin("1.4")*/

public inline fun
BooleanArray.reduceRightIndexedOrNullable(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean? {
    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n    }\n    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying \[operation\] from right to left\n * to each element and current accumulator value.\n* \n* @param \[operation\] function that takes an element and current accumulator value,\n* \n* @sample
samples.collections.Collections.Aggregates.reduceRightOrNullable */@SinceKotlin("1.4")*/

public inline fun
CharArray.reduceRightOrNullable(operation: (index: Int, Char, acc: Char) -> Char): Char? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return accumulator\n}

/**
 * Accumulates value starting with the last element and applying \[operation\] from right to left\n * to each element and current accumulator value.\n* \n* @param \[operation\] function that takes an element and current accumulator value,\n* \n* @sample
arrays.<T>reduceRightOrNullable */@ExperimentalStdlibApi::class*/

public inline fun
Array<out T>.reduceRightOrNullable(operation: (T, acc: S) -> S): S? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator: S = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return accumulator\n}
and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun ByteArray.reduceRightOrNull(operation: (Byte, acc: Byte) -> Byte): Byte? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}
```

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun ShortArray.reduceRightOrNull(operation: (Short, acc: Short) -> Short): Short? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}
```

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun IntArray.reduceRightOrNull(operation: (Int, acc: Int) -> Int): Int? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}
```

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun LongArray.reduceRightOrNull(operation: (Long, acc: Long) -> Long): Long? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}
```

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun FloatArray.reduceRightOrNull(operation: (Float, acc: Float) -> Float): Float? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}
```

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun DoubleArray.reduceRightOrNull(operation: (Double, acc: Double) -> Double): Double? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}
```
current accumulator value, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun BooleanArray.reduceRightOrNull(operation: (Boolean, acc: Boolean) -> Boolean): Boolean?
{
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left to each element and current accumulator value. Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun CharArray.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char?
{
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
@InlineOnly
public inline fun <T, R> Array<out T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R>
{
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
    {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
@InlineOnly
public inline fun <T, R> CharArray.runningFold(initial: R, operation: (acc: R, Char) -> R): List<R>
{
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
    {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
@InlineOnly
public inline fun <T, R> ShortArray.runningFold(initial: R, operation: (acc: R, Short) -> R): List<R>
{
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
    {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
@InlineOnly
{
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
    {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 */
@SinceKotlin("1.4")
@InlineOnly
public inline fun <T, R> ByteArray.runningFold(initial: R, operation: (acc: R, Byte) -> R): List<R>
{
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this)
    {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}
```

---

Open Source Used In DNAC Platform Antelope 13757
1).apply { add(initial) }
var accumulator = initial
for (element in this) {
    accumulator = operation(accumulator, element)
    result.add(accumulator)
}
return result

* Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.

* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

@sample samples.collections.Collections.Aggregates.runningFold

fun LongArray.runningFold(initial: Long, operation: (acc: Long, Long) -> Long): List<Long> { if (isEmpty()) return listOf(initial)
    val result = ArrayList<Long>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * 
 * @param
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
fun FloatArray.runningFold(initial: Float, operation: (acc: Float, Float) -> Float): List<Float> { if (isEmpty()) return listOf(initial)
    val result = ArrayList<Float>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * 
 * @param
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
fun DoubleArray.runningFold(initial: Double, operation: (acc: Double, Double) -> Double): List<Double> { if (isEmpty()) return listOf(initial)
    val result = ArrayList<Double>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * 
 * @param
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
fun BooleanArray.runningFold(initial: Boolean, operation: (acc: Boolean, Boolean) -> Boolean): List<Boolean> { if (isEmpty()) return listOf(initial)
    val result = ArrayList<Boolean>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * 
 * @param
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
fun CharArray.runningFold(initial: Char, operation: (acc: Char, Char) -> Char): List<Char> { if (isEmpty()) return listOf(initial)
    val result = ArrayList<Char>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}
its index in the original array and current accumulator value that starts with [initial] value.\n\n* \n* Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n* @param [operation] function that takes the index of an element, current accumulator value.\n* and the element itself, and calculates the next accumulator value.\n* \n* @param samples.collections.Collections.Aggregates.runningFold

*\n*SinceKotlin("1.4")\n*public inline fun <T, R>

Array\<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List\<R> \n{ \n    if (isEmpty())
        return listOf(initial) \n    var result = ArrayList\<R\>(size + 1).apply { add(initial) } \n    var accumulator = initial \n    for (index in indices) { \n        accumulator = operation(index, accumulator, this[index]) \n    } \n    return result.add(accumulator) \n} \n
*\n*Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n* @param [operation] function that takes the index of an element, current accumulator value.\n* and the element itself, and calculates the next accumulator value.\n* \n* @param samples.collections.Collections.Aggregates.runningFold

*\n*SinceKotlin("1.4")\n*public inline fun <T, R>

ByteArray\.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List\<R> \n{ \n    if (isEmpty())
        return listOf(initial) \n    var result = ArrayList\<R\>(size + 1).apply { add(initial) } \n    var accumulator = initial \n    for (index in indices) { \n        accumulator = operation(index, accumulator, this[index]) \n    } \n    return result.add(accumulator) \n} \n
*\n*Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n* @param [operation] function that takes the index of an element, current accumulator value.\n* and the element itself, and calculates the next accumulator value.\n* \n* @param samples.collections.Collections.Aggregates.runningFold

*\n*SinceKotlin("1.4")\n*public inline fun <T, R>

ShortArray\.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List\<R> \n{ \n    if (isEmpty())
        return listOf(initial) \n    var result = ArrayList\<R\>(size + 1).apply { add(initial) } \n    var accumulator = initial \n    for (index in indices) { \n        accumulator = operation(index, accumulator, this[index]) \n    } \n    return result.add(accumulator) \n} \n
*\n*Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n* @param [operation] function that takes the index of an element, current accumulator value.\n* and the element itself, and calculates the next accumulator value.\n* \n* @param samples.collections.Collections.Aggregates.runningFold

*\n*SinceKotlin("1.4")\n*public inline fun <T, R>

IntArray\.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List\<R> \n{ \n    if (isEmpty()) return \n    listOf(initial) \n    var result = ArrayList\<R\>(size + 1).apply { add(initial) } \n    var accumulator = initial \n    for (index in indices) { \n        accumulator = operation(index, accumulator, this[index]) \n    } \n    return result.add(accumulator) \n} \n
*\n*Note that `acc` value passed to [operation] function should not be mutated;\n* otherwise it would affect the previous value in resulting list.\n* \n* @param [operation] function that takes the index of an element, current accumulator value.\n* and the element itself, and calculates the next accumulator value.\n* \n* @param samples.collections.Collections.Aggregates.runningFold

*\n*SinceKotlin("1.4")\n*public inline fun <T, R>

LongArray\.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List\<R> \n{ \n    if (isEmpty())
        return listOf(initial) \n    var result = ArrayList\<R\>(size + 1).apply { add(initial) } \n    var accumulator = initial \n    for (index in indices) { \n        accumulator = operation(index, accumulator, this[index]) \n    } \n    return result.add(accumulator) \n}
result.add(accumulator)
}

/**
 * Returns a list containing successive accumulation
 * values generated by applying [operation] from left to right
 * to each element, its index in the original array and
 * current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function
 * should not be mutated; otherwise it would affect the
 * previous value in resulting list.
 * @param [operation] function that takes the index of an element,
 * current accumulator value and the element itself, and calculates
 * the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun FloatArray.runningFoldIndexed(initial: Float, operation: (index: Int, acc: Float, Float) -> Float): List<Float> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<Float>(size + 1).apply {
        add(initial)
    }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation
 * values generated by applying [operation] from left to right
 * to each element, its index in the original array and
 * current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function
 * should not be mutated; otherwise it would affect the
 * previous value in resulting list.
 * @param [operation] function that takes the index of an element,
 * current accumulator value and the element itself, and calculates
 * the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.runningFoldIndexed(initial: Double, operation: (index: Int, acc: Double, Double) -> Double): List<Double> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<Double>(size + 1).apply {
        add(initial)
    }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation
 * values generated by applying [operation] from left to right
 * to each element, its index in the original array and
 * current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function
 * should not be mutated; otherwise it would affect the
 * previous value in resulting list.
 * @param [operation] function that takes the index of an element,
 * current accumulator value and the element itself, and calculates
 * the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun BooleanArray.runningFoldIndexed(initial: Boolean, operation: (index: Int, acc: Boolean, Boolean) -> Boolean): List<Boolean> {
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<Boolean>(size + 1).apply {
        add(initial)
    }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation
 * values generated by applying [operation] from left to right
 * to each element, its index in the original array and
 * current accumulator value that starts with the first element
 * of this array.
 * Note that `acc` value passed to [operation] function
 * should not be mutated; otherwise it would affect the
 * previous value in resulting list.
 * @param [operation] function that takes the index of an element,
 * current accumulator value and the element itself, and calculates
 * the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun Array<out T>.runningReduce(operation: (acc: T, T) -> T): List<T> {
    if (isEmpty()) return emptyList()
    var

Open Source Used In DNAC Platform Antelope 13760
accumulator: S = this[0]
val result = ArrayList<S>(size).apply { add(accumulator) }
for (index in 1 until size) {
    accumulator = operation(accumulator, this[index])
    result.add(accumulator)
}
return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningReduce

*@
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ByteArray.runningReduce(operation: (acc: Byte, Byte) -> Byte): List<Byte> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Byte>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@sample samples.collections.Collections.Aggregates.runningReduce

*@
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun ShortArray.runningReduce(operation: (acc: Short, Short) -> Short): List<Short> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Short>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@sample samples.collections.Collections.Aggregates.runningReduce

*@
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun IntArray.runningReduce(operation: (acc: Int, Int) -> Int): List<Int> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Int>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@sample samples.collections.Collections.Aggregates.runningReduce

*@
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun LongArray.runningReduce(operation: (acc: Long, Long) -> Long): List<Long> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Long>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@sample samples.collections.Collections.Aggregates.runningReduce

*@
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun FloatArray.runningReduce(operation: (acc: Float, Float) -> Float): List<Float> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Float>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with the first element of this array.

@sample samples.collections.Collections.Aggregates.runningReduce

*@
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.runningReduce(operation: (acc: Double, Double) -> Double): List<Double> {
    if (isEmpty()) return emptyList()
    var accumulator =
this[0][n] val result = ArrayList<Double>(size).apply { add(accumulator) } for (index in 1 until size) {
    accumulator = operation(accumulator, this[index])
    result.add(accumulator)
} return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with the first element of this array.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun IntArray.runningReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {
    if (isEmpty()) return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Int>(size).apply { add(accumulator) }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with the first element of this array.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce

```
*\@SinceKotlin("1.4\")*\@kotlin.internal.InlineOnly\npublic inline fun CharArray.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<Char>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <T, R> Array<out T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {
    return runningFold(initial, operation)\n}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.scan(initial: R, operation: (acc: R, Byte) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> ShortArray.scan(initial: R, operation: (acc: R, Short) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> IntArray.scan(initial: R, operation: (acc: R, Int) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right
 * to each element and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> LongArray.scan(initial: R, operation: (acc: R, Long) -> R): List<R> {
    return runningFold(initial, operation)
}

/*\@SinceKotlin("1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n*/

c inline fun <R> FloatArray.scan(initial: R, operation: (acc: R, Float) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * from left to right to each element and current accumulator value that starts with [initial]
 * value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would
 * affect the previous value in resulting list.
 * 
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> DoubleArray.scan(initial: R, operation: (acc: R, Double) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * from left to right to each element and current accumulator value that starts with [initial]
 * value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would
 * affect the previous value in resulting list.
 * 
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> BooleanArray.scan(initial: R, operation: (acc: R, Boolean) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * from left to right to each element and current accumulator value that starts with [initial]
 * value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would
 * affect the previous value in resulting list.
 * 
 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> CharArray.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * from left to right to each element, its index in the original array and current accumulator value that starts with [initial]
 * value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would
 * affect the previous value in resulting list.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <T, R> Array<out T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * from left to right to each element, its index in the original array and current accumulator value that starts with [initial]
 * value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would
 * affect the previous value in resulting list.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * from left to right to each element, its index in the original array and current accumulator value that starts with [initial]
 * value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would
 * affect the previous value in resulting list.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> ByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}
c inline fun <R> ShortArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 */

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 */

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 */

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> FloatArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 */

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> DoubleArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 */

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 */

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 */

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 */

@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}
c inline fun ShortArray.scanReduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): List<Short> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.",
    ReplaceWith("runningReduceIndexed(operation)"), level =
    DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun IntArray.scanReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.",
    ReplaceWith("runningReduceIndexed(operation)"), level =
    DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.",
    ReplaceWith("runningReduceIndexed(operation)"), level =
    DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun FloatArray.scanReduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): List<Float> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.",
    ReplaceWith("runningReduceIndexed(operation)"), level =
    DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun BooleanArray.scanReduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): List<Boolean> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.",
    ReplaceWith("runningReduceIndexed(operation)"), level =
    DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
public inline fun CharArray.scanReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {
    return runningReduceIndexed(operation)
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun <T> Array<out T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ByteArray.sumBy(selector: (Byte) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun ShortArray.sumBy(selector: (Short) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun IntArray.sumBy(selector: (Int) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun LongArray.sumBy(selector: (Long) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun FloatArray.sumBy(selector: (Float) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun DoubleArray.sumBy(selector: (Double) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun BooleanArray.sumBy(selector: (Boolean) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
public inline fun CharArray.sumBy(selector: (Char) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
Returns the sum of all values produced by [selector] function applied to each element in the array.

```
@public inline fun <T> Array<out T>.sumByDouble(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
*public inline fun ByteArray.sumByDouble(selector: (Byte) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
*public inline fun ShortArray.sumByDouble(selector: (Short) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
*public inline fun IntArray.sumByDouble(selector: (Int) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
*public inline fun LongArray.sumByDouble(selector: (Long) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
*public inline fun FloatArray.sumByDouble(selector: (Float) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
*public inline fun DoubleArray.sumByDouble(selector: (Double) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
*public inline fun BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
*public inline fun CharArray.sumByDouble(selector: (Char) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun Array<out T>.sumOf(selector: (T) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun ByteArray.sumOf(selector: (Byte) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun ShortArray.sumOf(selector: (Short) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun IntArray.sumOf(selector: (Int) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun LongArray.sumOf(selector: (Long) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun FloatArray.sumOf(selector: (Float) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun DoubleArray.sumOf(selector: (Double) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun BooleanArray.sumOf(selector: (Boolean) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
*public inline fun CharArray.sumOf(selector: (Char) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```
Open Source Used In DNAC Platform Antelope 13769
```
LongArray.sumOf(selector: (Long) -> Double): Double {
    var sum = selector(element)
    return sum
}

FloatArray.sumOf(selector: (Float) -> Double): Double {
    var sum = selector(element)
    return sum
}

CharArray.sumOf(selector: (Char) -> Double): Double {
    var sum = selector(element)
    return sum
}

ByteArray.sumOf(selector: (Byte) -> Int): Int {
    var sum = selector(element)
    return sum
}

ShortArray.sumOf(selector: (Short) -> Int): Int {
    var sum = selector(element)
    return sum
}

IntArray.sumOf(selector: (Int) -> Int): Int {
    var sum = selector(element)
    return sum
}

LongArray.sumOf(selector: (Long) -> Double): Double {
    var sum = selector(element)
    return sum
}

BooleanArray.sumOf(selector: (Boolean) -> Double): Double {
    var sum = selector(element)
    return sum
}

Array<out T>.sumOf(selector: (T) -> Int): Int {
    var sum = selector(element)
    return sum
}
```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun FloatArray.sumOf(selector: (Float) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun DoubleArray.sumOf(selector: (Double) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfInt")
@kotlin.internal.InlineOnly
public inline fun BooleanArray.sumOf(selector: (Boolean) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfLong")
@kotlin.internal.InlineOnly
public inline fun CharArray.sumOf(selector: (Char) -> Int): Int {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfLong")
@kotlin.internal.InlineOnly
public inline fun Array<out T>.sumOf(selector: (T) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfLong")
@kotlin.internal.InlineOnly
public inline fun ByteArray.sumOf(selector: (Byte) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfLong")
@kotlin.internal.InlineOnly
public inline fun ShortArray.sumOf(selector: (Short) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfLong")
@kotlin.internal.InlineOnly
public inline fun IntArray.sumOf(selector: (Int) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfLong")
@kotlin.internal.InlineOnly
public inline fun LongArray.sumOf(selector: (Long) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
* Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfLong")
@kotlin.internal.InlineOnly
public inline fun FloatArray.sumOf(selector: (Float) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element) /n
    return sum
} /n
```

Open Source Used In DNAC Platform Antelope 13771
sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfLong")@kotlin.internal.InlineOnly
public inline fun DoubleArray.sumOf(selector: (Double) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfLong")@kotlin.internal.InlineOnly
public inline fun BooleanArray.sumOf(selector: (Boolean) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfLong")@kotlin.internal.InlineOnly
public inline fun CharArray.sumOf(selector: (Char) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@kotlin.experimental.UnsignedTypes@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.sumOf(selector: (T) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@kotlin.experimental.UnsignedTypes@kotlin.internal.InlineOnly
public inline fun ByteArray.sumOf(selector: (Byte) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@kotlin.experimental.UnsignedTypes@kotlin.internal.InlineOnly
public inline fun ShortArray.sumOf(selector: (Short) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@kotlin.experimental.UnsignedTypes@kotlin.internal.InlineOnly
public inline fun IntArray.sumOf(selector: (Int) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@kotlin.experimental.UnsignedTypes@kotlin.internal.InlineOnly
public inline fun LongArray.sumOf(selector: (Long) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@kotlin.experimental.UnsignedTypes@kotlin.internal.InlineOnly
public inline fun FloatArray.sumOf(selector: (Float) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun DoubleArray.sumOf(selector: (Double) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun BooleanArray.sumOf(selector: (Boolean) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfUInt")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun CharArray.sumOf(selector: (Char) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun <T> Array<out T>.sumOf(selector: (T) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun ByteArray.sumOf(selector: (Byte) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun ShortArray.sumOf(selector: (Short) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun IntArray.sumOf(selector: (Int) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun LongArray.sumOf(selector: (Long) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun FloatArray.sumOf(selector: (Float) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")@@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@n@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@n@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly@public inline fun DoubleArray.sumOf(selector: (Double) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun BooleanArray.sumOf(selector: (Boolean) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun CharArray.sumOf(selector: (Char) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns an original collection containing all the non-null elements, throwing an [IllegalArgument exceptionally] if there are any `null` elements.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public fun <T : Any> Array<T>?.requireNoNulls(): Array<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as Array<T>
}

/**
 * Splits the original array into pair of lists, *first* list contains elements for which [predicate] yielded `true`, *second* list contains elements for which [predicate] yielded `false`.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <T> Array<out T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {
    val first = ArrayList<T>()
    val second = ArrayList<T>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original array into pair of lists, *first* list contains elements for which [predicate] yielded `true`, *second* list contains elements for which [predicate] yielded `false`.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <T> ByteArray.partition(predicate: (Byte) -> Boolean): Pair<List<Byte>, List<Byte>> {
    val first = ArrayList<Byte>()
    val second = ArrayList<Byte>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original array into pair of lists, *first* list contains elements for which [predicate] yielded `true`, *second* list contains elements for which [predicate] yielded `false`.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <T> ShortArray.partition(predicate: (Short) -> Boolean): Pair<List<Short>, List<Short>> {
    val first = ArrayList<Short>()
    val second = ArrayList<Short>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original array into pair of lists, *first* list contains elements for which [predicate] yielded `true`, *second* list contains elements for which [predicate] yielded `false`.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <T> IntArray.partition(predicate: (Int) -> Boolean): Pair<List<Int>, List<Int>> {
    val first = ArrayList<Int>()
    val second = ArrayList<Int>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original array into pair of lists, *first* list contains elements for which [predicate] yielded `true`, *second* list contains elements for which [predicate] yielded `false`.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <T> LongArray.partition(predicate: (Long) -> Boolean): Pair<List<Long>, List<Long>> {
    val first = ArrayList<Long>()
    val second = ArrayList<Long>()
    for (element in this) {
        if (predicate(element)) {
            first.add(element)
        } else {
            second.add(element)
        }
    }
    return Pair(first, second)
}

/**
 * Splits the original array into pair of lists, *first* list contains elements for which [predicate] yielded `true`, *second* list contains elements for which [predicate] yielded `false`.

@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.jvm.JvmName("sumOfULong")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <T> FloatArray.partition(predicate: (Float) -> Boolean): Pair<List<Float>, List<Float>> {
    val first =
ArrayList<Float>()
val first = ArrayList<Float>()
val second = ArrayList<Float>()
for (element in this) {
    if (predicate(element)) {
        first.add(element)
    } else {
        second.add(element)
    }
}
return Pair(first, second)

/**
 * Splits the original array into pair of lists, \( n \) * where \( *first* \) list contains elements for which
 * \( [predicate] \) yielded \('true', \( n \) * while \( *second* \) list contains elements for which \( [predicate] \)
 * yielded \('false', \( n \) * \( @sample \) \ samples.collections.Arrays.Transformations.partitionArrayOfPrimitives
 */

ArrayList<Double>()
val first = ArrayList<Double>()
val second = ArrayList<Double>()
for (element in this) {
    if (predicate(element)) {
        first.add(element)
    } else {
        second.add(element)
    }
}
return Pair(first, second)

/**
 * Splits the original array into pair of lists, \( n \) * where \( *first* \) list contains elements for which
 * \( [predicate] \) yielded \('true', \( n \) * while \( *second* \) list contains elements for which \( [predicate] 
 * yielded \('false', \( n \) * \( @sample \) \ samples.collections.Arrays.Transformations.partitionArrayOfPrimitives
 */

CharArray().partition(predicate: (Char) -> Boolean) = Pair<List<Char>, List<Char>>

CharArray().partition(predicate: (Char) -> Boolean) = Pair<List<Char>, List<Char>>

*/

IntArray().partition(predicate: (Int) -> Boolean) = Pair<List<Int>, List<Int>>

IntArray().partition(predicate: (Int) -> Boolean) = Pair<List<Int>, List<Int>>

LongArray().partition(predicate: (Long) -> Boolean) = Pair<List<Long>, List<Long>>

LongArray().partition(predicate: (Long) -> Boolean) = Pair<List<Long>, List<Long>>

FloatArray().partition(predicate: (Float) -> Boolean) = Pair<List<Float>, List<Float>>

FloatArray().partition(predicate: (Float) -> Boolean) = Pair<List<Float>, List<Float>>

DoubleArray().partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>>

DoubleArray().partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>>

ByteArray().partition(predicate: (Byte) -> Boolean) = Pair<List<Byte>, List<Byte>>

ByteArray().partition(predicate: (Byte) -> Boolean) = Pair<List<Byte>, List<Byte>>

ShortArray().partition(predicate: (Short) -> Boolean) = Pair<List<Short>, List<Short>>

ShortArray().partition(predicate: (Short) -> Boolean) = Pair<List<Short>, List<Short>>

IntArray().partition(predicate: (Int) -> Boolean) = Pair<List<Int>, List<Int>>

IntArray().partition(predicate: (Int) -> Boolean) = Pair<List<Int>, List<Int>>
Returns a list of pairs built from the elements of 'this' array and the [other] array with the same index.

* The returned list has length of the shortest collection.

```kotlin
/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public infix fun <R> CharArray.zip(other: Array<out R>): List<Pair<Char, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public inline fun <T, R> ShortArray.zip(other: Array<out R>, transform: (a: Short, b: R) -> V): List<V> {
    return zip(other) { t1, t2 -> transform(t1, t2) }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public inline fun <T, R> IntArray.zip(other: Array<out R>, transform: (a: Int, b: R) -> V): List<V> {
    return zip(other) { t1, t2 -> transform(t1, t2) }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public inline fun <T, R> LongArray.zip(other: Array<out R>, transform: (a: Long, b: R) -> V): List<V> {
    return zip(other) { t1, t2 -> transform(t1, t2) }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public inline fun <T, R> FloatArray.zip(other: Array<out R>, transform: (a: Float, b: R) -> V): List<V> {
    return zip(other) { t1, t2 -> transform(t1, t2) }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public inline fun <T, R> DoubleArray.zip(other: Array<out R>, transform: (a: Double, b: R) -> V): List<V> {
    return zip(other) { t1, t2 -> transform(t1, t2) }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Iterables.Operations.zipIterable
 */
public inline fun <T, R> BooleanArray.zip(other: Array<out R>, transform: (a: Boolean, b: R) -> V): List<V> {
    return zip(other) { t1, t2 -> transform(t1, t2) }
}
```

Open Source Used In DNAC Platform Antelope 13776
other.size)\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}

/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> CharArray.zip(other: Array<out R>, transform: (a: Char, b: R) -> V): List<V> {\n    val list = ArrayList<V>(size)\n    for (i in 0 until size) {\n        list.add(transform(this[i], other[i]))\n    }\n    return list\n}

/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}

/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> ShortArray.zip(other: Iterable<R>): List<Pair<Short, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}

/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> IntArray.zip(other: Iterable<R>): List<Pair<Int, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}

/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> FloatArray.zip(other: Iterable<R>): List<Pair<Float, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}

/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> DoubleArray.zip(other: Iterable<R>): List<Pair<Double, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}

/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> BooleanArray.zip(other: Iterable<R>): List<Pair<Boolean, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}

/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> CharArray.zip(other: Iterable<R>): List<Pair<Char, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}

/**\n * Returns a list of pairs built from the elements of `this` array and [other] array with the same index.\n * Using the provided [transform] function applied to each pair of elements.\n * @sample samples.collections.Operations.zip\n */\npublic inline fun <R, V> Array<out T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n    val arraySize = size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other) {
        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}
fun <R, V> ByteArray.zip(other: Iterable<R>, transform: (a: Byte, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterableWithTransform
 */
public inline fun <R, V> ShortArray.zip(other: Iterable<R>, transform: (a: Short, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterableWithTransform
 */
public inline fun <R, V> IntArray.zip(other: Iterable<R>, transform: (a: Int, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterableWithTransform
 */
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterableWithTransform
 */
public inline fun <R, V> FloatArray.zip(other: Iterable<R>, transform: (a: Float, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterableWithTransform
 */
public inline fun <R, V> DoubleArray.zip(other: Iterable<R>, transform: (a: Double, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterableWithTransform
 */
public inline fun <R, V> BooleanArray.zip(other: Iterable<R>, transform: (a: Boolean, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest collection.
 * @sample samples.collections.Operations.zipIterableWithTransform
 */
public inline fun <R, V> CharArray.zip(other: Iterable<R>, transform: (a: Char, b: R) -> V): List<V> {
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}
Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.

The returned list has length of the shortest collection.

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.

The returned list has length of the shortest collection.

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.

The returned list has length of the shortest collection.

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.

The returned list has length of the shortest collection.

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.

The returned list has length of the shortest collection.

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.

The returned list has length of the shortest collection.

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.

The returned list has length of the shortest collection.

Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.

The returned list has length of the shortest collection.

Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest array.

Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest array.

Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest array.

Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest array.

Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest array.

Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest array.

Returns a list of values built from the elements of `this` array and the [other] array using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest array.
*/
public inline fun <V> LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {
    val
    size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] array with the same index *
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest array.
 */
public inline fun <V> FloatArray.zip(other: FloatArray, transform: (a: Float, b: Float) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] array with the same index *
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest array.
 */
public inline fun <V> DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] array with the same index *
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest array.
 */
public inline fun <V> BooleanArray.zip(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of values built from the elements of 'this' array and the [other] array with the same index *
 * using the provided [transform] function applied to each pair of elements.
 * The returned list has length of the shortest array.
 */
public inline fun <V> CharArray.zip(other: CharArray, transform: (a: Char, b: Char) -> V): List<V> {
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). *
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <T, A : Appendable> Array<out T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). *
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> ByteArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Byte) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). *
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> CharArray.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Char) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). *
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> CharSequence.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((CharSequence) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). *
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> CharSequence.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((CharSequence) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). *
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable>CharSequence.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((CharSequence) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). *
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable>CharSequence.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((CharSequence) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied. *
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). *
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable>CharSequence.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((CharSequence) -> CharSequence)? = null): A {
    buffer.append(prefix)
    var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            if (transform != null)
                buffer.append(transform(element))
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}
@sample samples.collections.Collections.Transformations.joinTo

public fun <A : Appendable> ShortArray.joinTo(buffer: A, separator: CharSequence = "", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Short) -> CharSequence)? = null): A {
    buffer.append(prefix) var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            var transformed = transform?.invoke(element)
            if (transformed != null)
                buffer.append(transformed)
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given
 * [prefix] and [postfix] if supplied.
 * 
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> IntArray.joinTo(buffer: A, separator: CharSequence = "", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Int) -> CharSequence)? = null): A {
    buffer.append(prefix) var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            var transformed = transform?.invoke(element)
            if (transformed != null)
                buffer.append(transformed)
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given
 * [prefix] and [postfix] if supplied.
 * 
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> LongArray.joinTo(buffer: A, separator: CharSequence = "", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Long) -> CharSequence)? = null): A {
    buffer.append(prefix) var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            var transformed = transform?.invoke(element)
            if (transformed != null)
                buffer.append(transformed)
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given
 * [prefix] and [postfix] if supplied.
 * 
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> FloatArray.joinTo(buffer: A, separator: CharSequence = "", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Float) -> CharSequence)? = null): A {
    buffer.append(prefix) var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            var transformed = transform?.invoke(element)
            if (transformed != null)
                buffer.append(transformed)
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}

/**
 * Appends the string from all the elements separated using [separator] and using the given
 * [prefix] and [postfix] if supplied.
 * 
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> DoubleArray.joinTo(buffer: A, separator: CharSequence = "", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((Double) -> CharSequence)? = null): A {
    buffer.append(prefix) var count = 0
    for (element in this) {
        if (++count > 1)
            buffer.append(separator)
        if (limit < 0 || count <= limit) {
            var transformed = transform?.invoke(element)
            if (transformed != null)
                buffer.append(transformed)
            else
                buffer.append(element.toString())
        } else break
    }
    if (limit >= 0 && count > limit) buffer.append(truncated)
    buffer.append(postfix)
    return buffer
}
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "..."). If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").

```java
/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<out CharSequence>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<CharSequence>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((Char) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<Boolean>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((Boolean) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<Byte>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((Byte) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<Long>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((Long) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<Double>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((Double) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<String>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((String) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<Throwable>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((Throwable) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * 
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <A : Appendable> Array<Array<ByteArray>>.joinTo(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = ", ", limit: Int = -1, truncated: CharSequence = "...", transform: ((ByteArray) -> CharSequence)? = null): String {
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}
```
huge, you can specify a non-negative value of [limit], in which case only the first [limit]n * elements will be appended, followed by the [truncated] string (which defaults to "\ldots")n * n * @sample
samples.collections.Collections.Transformations.joinToString

/*
 * @sample
 * samples.collections.Collections.Transformations.joinToString
 */
public fun LongArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = "\ldots", transform: ((Long) -> CharSequence)? = null): String { return joinToString(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString().n*"n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.n * n *
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]n * elements will be appended, followed by the [truncated] string (which defaults to "\ldots")n * n * @sample
samples.collections.Collections.Transformations.joinToString

/*
 * @sample
 * samples.collections.Collections.Transformations.joinToString
 */
public fun FloatArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = "\ldots", transform: ((Float) -> CharSequence)? = null): String { return joinToString(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString().n*"n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.n * n *
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]n * elements will be appended, followed by the [truncated] string (which defaults to "\ldots")n * n * @sample
samples.collections.Collections.Transformations.joinToString

/*
 * @sample
 * samples.collections.Collections.Transformations.joinToString
 */
public fun DoubleArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = "\ldots", transform: ((Double) -> CharSequence)? = null): String { return joinToString(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString().n*"n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.n * n *
If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]n * elements will be appended, followed by the [truncated] string (which defaults to "\ldots")n * n * @sample
samples.collections.Collections.Transformations.joinToString

/*
 * @sample
 * samples.collections.Collections.Transformations.joinToString
 */
public fun BooleanArray.joinToString(separator: CharSequence = ", ", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = "\ldots", transform: ((Boolean) -> CharSequence)? = null): String { return joinToString(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString().n*"n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.n * An public fun <T> Array<out T>.asIterable(): Iterable<T> { if (isEmpty()) return emptyList().n * return Iterable { this.iterator() }n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.n * An public fun ByteArray.asIterable(): Iterable<Byte> { if (isEmpty()) return emptyList().n * return Iterable { this.iterator() }n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.n * An public fun ShortArray.asIterable(): Iterable<Short> { if (isEmpty()) return emptyList().n * return Iterable { this.iterator() }n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.n * An public fun IntArray.asIterable(): Iterable<Int> { if (isEmpty()) return emptyList().n * return Iterable { this.iterator() }n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.n * An public fun LongArray.asIterable(): Iterable<Long> { if (isEmpty()) return emptyList().n * return Iterable { this.iterator() }n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.n * An public fun FloatArray.asIterable(): Iterable<Float> { if (isEmpty()) return emptyList().n * return Iterable { this.iterator() }n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.n * An public fun
DoubleArray.asIterable(): Iterable<Double> {
    if (isEmpty()) return emptyList()
    return Iterable {
        this.iterator()
    }
}

/**
 * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
 */
public fun BooleanArray.asIterable(): Iterable<Boolean> {
    if (isEmpty()) return emptyList()
    return Iterable {
        this.iterator()
    }
}

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 */
public fun CharArray.asIterable(): Iterable<Char> {
    if (isEmpty()) return emptyList()
    return Iterable {
        this.iterator()
    }
}

/**
 * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.
 */
public fun <T> Array<out T>.asSequence(): Sequence<T> {
    if (isEmpty()) return emptySequence()
    return Sequence {
        this.iterator()
    }
}

/**
 * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.
 */
public fun ByteArray.asSequence(): Sequence<Byte> {
    if (isEmpty()) return emptySequence()
    return Sequence {
        this.iterator()
    }
}

/**
 * Calculates the average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfByte")
public fun Array<out Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Calculates the average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfShort")
public fun Array<out Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Calculates the average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfInt")
public fun Array<out Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Calculates the average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Array<out Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Calculates the average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Array<out Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}
if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfDouble")
public fun Array<out Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfInt")
public fun Array<out Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Array<out Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Array<out Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfByte")
public fun ByteArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfShort")
public fun Array<out Short>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the array.
 */
@kotlin.jvm.JvmName("averageOfChar")
public fun CharArray.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        ++count
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Array<out Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Array<out Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Array<out Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Array<out Float>.sum(): Float {
    var sum: Float = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Array<out Long>.sum(): Long {
    var sum: Long = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Array<out Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun ByteArray.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun ShortArray.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun IntArray.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun LongArray.sum(): Long {
    var sum: Long = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun FloatArray.sum(): Float {
    var sum: Float = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun DoubleArray.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}
random element from this range.

* * * 

* @SinceKotlin("1.3") inline fun IntRange.random(): Int {

    return random(Random())
}

* @SinceKotlin("1.3") inline fun LongRange.random(): Long {

    return random(Random())
}

* @SinceKotlin("1.3") inline fun CharRange.random(): Char {

    return random(Random())
}

* @SinceKotlin("1.3") inline fun CharRange.random(random: Random): Char {

    return random.nextInt(first.toInt(), last.toInt() + 1).toChar()
}

* @SinceKotlin("1.3") inline fun IntRange.randomOrNull(): Int? {

    return randomOrNull(Random())
}

* @SinceKotlin("1.3") inline fun LongRange.randomOrNull(): Long? {

    return randomOrNull(Random())
}

* @SinceKotlin("1.3") inline fun CharRange.randomOrNull(): Char? {

    return randomOrNull(Random())
}

* @SinceKotlin("1.3") inline operator fun IntRange.contains(element: Int?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun LongRange.contains(element: Long?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun CharRange.contains(element: Char?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun IntRange.contains(element: Int): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun LongRange.contains(element: Long): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun CharRange.contains(element: Char): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun IntRange.contains(element: Int?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun LongRange.contains(element: Long?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun CharRange.contains(element: Char?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun IntRange.contains(element: Int): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun LongRange.contains(element: Long): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun CharRange.contains(element: Char): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun IntRange.contains(element: Int?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun LongRange.contains(element: Long?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun CharRange.contains(element: Char?): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun IntRange.contains(element: Int): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun LongRange.contains(element: Long): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun CharRange.contains(element: Char): Boolean {

    return element != null && contains(element)
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: IntRange): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: LongRange): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: CharRange): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun CharRange.equals(other: Char): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun IntRange.equals(other: Int?): Boolean {

    return this == other
}

* @SinceKotlin("1.3") inline operator fun LongRange.equals(other: Long?): Boolean {
public inline operator fun CharRange.contains(element: Char): Boolean {
    return element != null && contains(element) * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Byte): Boolean {
    return contains(value.toInt()) * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Byte): Boolean {
    return contains(value.toLong()) * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Byte): Boolean {
    return contains(value.toShort()) * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("doubleRangeContains")
public operator fun ClosedRange<Double>.contains(value: Byte): Boolean {
    return contains(value.toDouble()) * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Byte): Boolean {
    return contains(value.toFloat()) * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Double): Boolean {
    return value.toIntOrNull().let { if (it != null) contains(it) else false } * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Double): Boolean {
    return value.toLongOrNull().let { if (it != null) contains(it) else false } * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("byteRangeContains")
public operator fun ClosedRange<Byte>.contains(value: Double): Boolean {
    return value.toByteOrNull().let { if (it != null) contains(it) else false } * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Double): Boolean {
    return value.toShortOrNull().let { if (it != null) contains(it) else false } * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Double): Boolean {
    return contains(value.toFloat()) * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("intRangeContains")
public operator fun ClosedRange<Int>.contains(value: Float): Boolean {
    return value.toIntOrNull().let { if (it != null) contains(it) else false } * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("longRangeContains")
public operator fun ClosedRange<Long>.contains(value: Float): Boolean {
    return value.toLongOrNull().let { if (it != null) contains(it) else false } * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("byteRangeContains")
public operator fun ClosedRange<Byte>.contains(value: Float): Boolean {
    return value.toByteOrNull().let { if (it != null) contains(it) else false } * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("shortRangeContains")
public operator fun ClosedRange<Short>.contains(value: Float): Boolean {
    return value.toShortOrNull().let { if (it != null) contains(it) else false } * Checks if the specified [value] belongs to this range.
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@kotlin.jvm.JvmName("floatRangeContains")
public operator fun ClosedRange<Float>.contains(value: Float): Boolean {
    return contains(value.toFloat()) * Checks if the specified [value] belongs to this range.
}
`Open Source Used In DNAC Platform Antelope` 13789

```kotlin
range
public operator fun ClosedRange<Int>.contains(value: Short): Boolean {
    return contains(value.toInt())
}

public operator fun ClosedRange<Long>.contains(value: Short): Boolean {
    return contains(value.toLong())
}

public operator fun ClosedRange<Byte>.contains(value: Short): Boolean {
    return value.toByteExactOrNull().let { if (it != null) contains(it) else false }
}

public operator fun ClosedRange<Double>.contains(value: Short): Boolean {
    return contains(value.toDouble())
}

public operator fun ClosedRange<Float>.contains(value: Short): Boolean {
    return contains(value.toFloat())
}

public infix fun Int.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this, to.toInt(), -1)
}

public infix fun Long.downTo(to: Byte): LongProgression {
    return LongProgression.fromClosedRange(this, to.toLong(), -1L)
}

public infix fun Byte.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

public infix fun Short.downTo(to: Byte): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

public infix fun Char.downTo(to: Char): CharProgression {
    return CharProgression.fromClosedRange(this, to, -1)
}

public infix fun Int.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this, to, -1)
}

public infix fun Long.downTo(to: Int): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1L)
}

public infix fun Byte.downTo(to: Int): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to, -1)
}
```

---

Open Source Used In DNAC Platform Antelope 13789
fun Int.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this.toLong(), to, -1L)
}

fun Long.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1L)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to `this` value.
 * If the [to] value is greater than `this` value the returned progression is empty.
 */
fun Int.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this.toLong(), to, -1L)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to `this` value.
 * If the [to] value is greater than `this` value the returned progression is empty.
 */
fun Long.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this, to, -1L)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to `this` value.
 * If the [to] value is greater than `this` value the returned progression is empty.
 */
fun Short.downTo(to: Long): ShortProgression {
    return ShortProgression.fromClosedRange(this.toDouble(), to, -1L)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to `this` value.
 * If the [to] value is greater than `this` value the returned progression is empty.
 */
fun Int.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this, to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to `this` value.
 * If the [to] value is greater than `this` value the returned progression is empty.
 */
fun Long.downTo(to: Short): LongProgression {
    return LongProgression.fromClosedRange(this, to.toLong(), -1L)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to `this` value.
 * If the [to] value is greater than `this` value the returned progression is empty.
 */
fun Byte.downTo(to: Long): LongProgression {
    return LongProgression.fromClosedRange(this.toLong(), to, -1L)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to `this` value.
 * If the [to] value is greater than `this` value the returned progression is empty.
 */
fun Byte.downTo(to: Short): IntProgression {
    return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * The [to] value should be less than or equal to `this` value.
 * If the [to] value is greater than `this` value the returned progression is empty.
 */
fun Short.downTo(to: Int): ShortProgression {
    return ShortProgression.fromClosedRange(this.toInt(), to.toInt(), -1)
}

/**
 * Returns a progression that goes over the same range in the opposite direction with the same step.
 */
fun IntProgression.reversed(): IntProgression {
    return IntProgression.fromClosedRange(last, first, -step)
}

/**
 * Returns a progression that goes over the same range in the opposite direction with the same step.
 */
fun LongProgression.reversed(): LongProgression {
    return LongProgression.fromClosedRange(last, first, -step)
}

/**
 * Returns a progression that goes over the same range in the opposite direction with the same step.
 */
fun CharProgression.reversed(): CharProgression {
    return CharProgression.fromClosedRange(last, first, -step)
}

/**
 * Returns a progression that goes over the same range with the given step.
 */
fun IntProgression.step(step: Int): IntProgression {
    checkStepIsPositive(step > 0, step)
    return IntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)
}

/**
 * Returns a progression that goes over the same range with the given step.
 */
fun LongProgression.step(step: Long): LongProgression {
    checkStepIsPositive(step > 0, step)
    return LongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)
}

/**
 * Returns a progression that goes over the same range with the given step.
 */
fun CharProgression.step(step: Int): CharProgression {
    checkStepIsPositive(step > 0, step)
    return CharProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)
}

fun Int.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toInt()..Byte.MAX_VALUE.toInt()) this.toByte() else null
}

fun Long.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toLong()..Byte.MAX_VALUE.toLong()) this.toByte() else null
}

fun Short.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toShort()..Byte.MAX_VALUE.toShort()) this.toByte() else null
}

fun Double.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toDouble()..Byte.MAX_VALUE.toDouble()) this.toInt().toByte() else null
}

fun Float.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toFloat()..Byte.MAX_VALUE.toFloat()) this.toFloat().toByte() else null
}

internal fun Int.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toInt()..Byte.MAX_VALUE.toInt()) this.toByte() else null
}

internal fun Long.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toLong()..Byte.MAX_VALUE.toLong()) this.toByte() else null
}

internal fun Short.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toShort()..Byte.MAX_VALUE.toShort()) this.toByte() else null
}

internal fun Double.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toDouble()..Byte.MAX_VALUE.toDouble()) this.toDouble().toByte() else null
}

internal fun Float.toByteExactOrNull(): Byte? {
    return if (this in Byte.MIN_VALUE.toFloat()..Byte.MAX_VALUE.toFloat()) this.toFloat().toByte() else null
}
Byte.MIN_VALUE.toFloat().Byte.MAX_VALUE.toFloat()) this.toInt().toByte() else null
\n\nuninteral fun Long.toIntExactOrNull(): Int? {\n    if (this in Int.MIN_VALUE.toLong()..Int.MAX_VALUE.toLong())\n        this.toInt() else null\n\nuninteral fun Double.toIntExactOrNull(): Int? {\n    if (this in Int.MIN_VALUE.toDouble()..Int.MAX_VALUE.toDouble())\n        this.toInt() else null\n\nuninteral fun Float.toIntExactOrNull(): Int? {\n    if (this in Int.MIN_VALUE.toFloat()..Int.MAX_VALUE.toFloat())\n        this.toInt() else null\n\nuninteral fun Short.toShortOrNull(): Short? {\n    if (this in Short.MIN_VALUE.toInt()..Short.MAX_VALUE.toInt())\n        this.toShort() else null\n\nuninteral fun Char.toCharOrNull(): Char? {\n    if (to <= Char.MIN_VALUE)\n        CharRange.EMPTY else\n            this.toChar()}

/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */
public infix fun Int.until(to: Byte): IntRange {
    return this .. (to.toInt() - 1).toInt()}

/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */
public infix fun Int.until(to: Int): IntRange {
    return this .. (to.toInt() - 1).toInt()}

/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */
public infix fun Int.until(to: Long): LongRange {
    return this .. (to.toLong() - 1).toLong()}

/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */
public infix fun Int.until(to: Double): DoubleRange {
    return this .. (to.toDouble() - 1).toDouble()}

/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */
public infix fun Int.until(to: Float): FloatRange {
    return this .. (to.toFloat() - 1).toFloat()
Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()

Long): LongRange
    if (to <= Long.MIN_VALUE) return LongRange.EMPTY
    return this.toLong() .. (to - 1).toLong()
greater than the specified \[\text{maximumValue}\].

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{maximumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceAtMost\n*\@public fun Short.coerceAtMost(maximumValue: Short):
Short \{\n    return if (this > maximumValue) maximumValue else this\n\n*\@return this value if it’s less than or equal to the \[\text{maximumValue}\] or the \[\text{maximumValue}\] otherwise.
```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{maximumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceAtMost\n*\@public fun Int.coerceAtMost(maximumValue: Int): Int
\{\n    return if (this > maximumValue) maximumValue else this\n\n*\@return this value if it’s less than or equal to the \[\text{maximumValue}\] or the \[\text{maximumValue}\] otherwise.
```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{maximumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceAtMost\n*\@public fun Double.coerceAtMost(maximumValue: Double): Double
\{\n    return if (this > maximumValue) maximumValue else this\n```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{maximumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceInComparable\n*\@public fun \<T : Comparable<T>> T.coerceIn(minimumValue: T?, maximumValue: T?): T
```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{minimumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceIn\n*\@public fun Byte.coerceIn(minimumValue: Byte, maximumValue: Byte): Byte
```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{minimumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceIn\n*\@public fun Short.coerceIn(minimumValue: Short, maximumValue: Short): Short
```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{minimumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceIn\n*\@public fun Double.coerceIn(minimumValue: Double, maximumValue: Double): Double
```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{minimumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceIn\n*\@public fun Float.coerceIn(minimumValue: Float, maximumValue: Float): Float
```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{minimumValue}\] or the \[\text{maximumValue}\] otherwise.

```java
*\@sample
samples.comparisons.ComparableOps.coerceIn\n*\@public fun Long.coerceIn(minimumValue: Long, maximumValue: Long): Long
```

\[\text{\textbf{return}}\] this value if it’s less than or equal to the \[\text{minimumValue}\] or the \[\text{maximumValue}\] otherwise.
return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}

/**
 * Ensures this value lies in the specified range [minimumValue].
 * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Long.coerceIn(minimumValue: Long, maximumValue: Long): Long {
    if (minimumValue > maximumValue)
        throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}

/**
 * Ensures that this value lies in the specified range [minimumValue]..
 * 
 * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].
 * 
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Float.coerceIn(minimumValue: Float, maximumValue: Float): Float {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}

/**
 * Ensures that this value lies in the specified range [minimumValue]..
 * 
 * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].
 * 
 * @sample samples.comparisons.ComparableOps.coerceIn
 */
public fun Double.coerceIn(minimumValue: Double, maximumValue: Double): Double {
    if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
    if (this < minimumValue) return minimumValue
    if (this > maximumValue) return maximumValue
    return this
}
(range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
    return when {
        this < range.start -> range.start
        this > range.endInclusive -> range.endInclusive
        else ->
            this
    }
}

package kotlin

import kotlin.experimental.*

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UByte @PublishedApi internal constructor(@PublishedApi internal val data: Byte) : Comparable<UByte> {

    companion object {
        /**
         * A constant holding the minimum value an instance of UByte can have.
         */
        public const val MIN_VALUE: UByte = UByte(0)

        /**
         * A constant holding the maximum value an instance of
         * UByte can have.
         */
        public const val MAX_VALUE: UByte = UByte(-1)

        /**
         * The number of bytes used to represent an instance of UByte in a binary form.
         */
        public const val SIZE_BYTES: Int = 1

        /**
         * The number of bits used to represent an instance of UByte
         * in a binary form.
         */
        public const val SIZE_BITS: Int = 8
    }

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    @Suppress("OVERRIDE_BY_INLINE")
    public override inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UShort): UInt = this.toUInt().minus(other.toUInt())

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UShort): UInt = this.toUInt().times(other.toUInt())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: ULong): ULong = this.toULong().times(other)

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UByte): UInt =
it's less than other,
* or a positive number if it's greater than other.

```kotlin
public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UByte): UInt = this.plus(other.toUInt())
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UShort): UInt = this.plus(other.toUInt())
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: UInt): UInt = UInt(this.data.plus(other.data))
/** Adds the other value to this value. */
@kotlin.internal.InlineOnly
public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)
```
value represents \( n \) the same numerical value as this `UInt`. \n
* The resulting `Byte` value is represented by the least significant 8 bits of this `UInt`. \n
* Note that the resulting `Byte` value may be negative. \n
* \@kotlin.internal.InlineOnly \n
* public inline fun toByte(): Byte = data.toByte() \n
* \/**\n
* Converts this [UInt] value to [Short]. \n
* If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value represents \( n \) the same numerical value as this `UInt`. \n
* The resulting `Short` value is represented by the least significant 16 bits of this `UInt` value. \n
* Note that the resulting `Short` value may be negative. \n
* \@kotlin.internal.InlineOnly \n
* public inline fun toShort(): Short = data.toShort() \n
* \/**\n
* Converts this [UInt] value to [Int]. \n
* If this value is less than or equals to [Int.MAX_VALUE], the resulting `Int` value represents \( n \) the same numerical value as this `UInt`. Otherwise the result is negative. \n
* \@kotlin.internal.InlineOnly \n
* public inline fun toInt(): Int = data \n
* \/**\n
* Converts this [UInt] value to [Long]. \n
* The least significant 32 bits of the resulting `Long` value are the same as the bits of this `UInt`. whereas the most significant 32 bits are filled with zeros. \n
* \@kotlin.internal.InlineOnly \n
* public inline fun toLong(): Long = data.toLong() \n
* \/**\n
* Converts this [UInt] value to [UByte]. \n
* \/**\n
* Converts this [UInt] value to [UShort]. \n
* \/**\n
* Returns this value. \n
* \@kotlin.internal.InlineOnly \n
* public inline fun toUInt(): UInt = this \n
* \/**\n
* Converts this [UInt] value to [ULong]. \n
* The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `UInt` value. whereas the most significant 32 bits are filled with zeros. \n
* \@kotlin.internal.InlineOnly \n
* public inline fun toULong(): ULong = ULong(data.toLong() and 0xFFFF_FFFF) \n
* \/**\n
* Converts this [UInt] value to [Float]. \n
* The resulting value is the closest `Float` to this `UInt` value. In case when this `UInt` value is exactly between two `Float` s, the one with zero at least significant bit of mantissa is selected. \n
* \@kotlin.internal.InlineOnly \n
* public inline fun toFloat(): Float = this.toDouble().toFloat() \n
* \/**\n
* Converts this [UInt] value to [Double]. \n
* The resulting `Double` value represents the same numerical value as this `UInt`. \n
* \@kotlin.internal.InlineOnly \n
* public inline fun toDouble(): Double = uintToDouble(data) \n
* \/**\n
* Converts this [UInt] value to [Byte]. \n
* Converts this [Byte] value to [UByte]. \n
* Converts this [UByte] value to [UInt]. \n
* The resulting `Byte` value represents the same numerical value as this `Byte`. whereas the most significant 24 bits are filled with the sign bit of this `Byte` value. whereas the most significant 24 bits are filled with the sign bit of this `Byte` value. 

* \@SinceKotlin("1.3") \@ExperimentalUnsignedTypes \n
* public inline fun Byte.toUInt(): UInt = UInt(this.toInt()) \n
* Converts this [Short] value to [UInt]. \n
* If this value is positive, the resulting `UInt` value represents the same numerical value as this `Short`. 

* The least significant 16 bits of the resulting `UInt` value are the same as the bits of this `Short` value. whereas the most significant 16 bits are filled with the sign bit of this value. 

* \@SinceKotlin("1.3") \@ExperimentalUnsignedTypes \n
* public inline fun Short.toUInt(): UInt = UInt(this.toInt()) \n
* Converts this [Int] value to [UInt]. 

* If this value is positive, the resulting `UInt` value represents the same numerical value as this `Int`. 

* The resulting `UInt` value has the same binary representation as this `Int` value. 

* \@SinceKotlin("1.3") \@ExperimentalUnsignedTypes \n
* public inline fun Int.toUInt(): UInt = UInt(this) \n
* Converts this [Long] value to [UInt]. 

* If this value is positive and less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents \( n \) the same numerical value as this

---

**Open Source Used In DNAC Platform Antelope 13799**
The resulting `UInt` value is represented by the least significant 32 bits of this `Long` value.

Converts this [Float] value to [UInt].

The fractional part, if any, is rounded down towards zero.

Returns zero if this `Double` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.

Converts this [Double] value to [UInt].

The fractional part, if any, is rounded down towards zero.

Returns zero if this `Double` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.

Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* A constant holding the minimum value an instance of UShort can have.

* A constant holding the maximum value an instance of UShort can have.

* The number of bytes used to represent an instance of UShort in a binary form.

* The number of bits used to represent an instance of UShort in a binary form.

* A constant holding the minimum value an instance of UShort can have.

* A constant holding the maximum value an instance of UShort can have.

* The number of bytes used to represent an instance of UShort in a binary form.

* The number of bits used to represent an instance of UShort in a binary form.

* A constant holding the minimum value an instance of UShort can have.

* A constant holding the maximum value an instance of UShort can have.

* A constant holding the minimum value an instance of UShort can have.

* A constant holding the maximum value an instance of UShort can have.
public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt()) /** Multiplies this value by the other value. */
public inline operator fun times(other: UShort): UInt = this.toUInt().times(other.toUInt()) /** Multiplies this value by the other value. */
public inline operator fun times(other: UInt): UInt = this.toUInt().times(other) /** Multiplies this value by the other value. */
public inline operator fun times(other: ULong): ULong = this.toULong().times(other.toUInt()) /** Divides this value by the other value. */
public inline operator fun div(other: UByte): UInt = this.toUInt().div(other.toUInt()) /** Divides this value by the other value. */
public inline operator fun div(other: UShort): UInt = this.toUInt().div(other.toUInt()) /** Divides this value by the other value. */
public inline operator fun div(other: UInt): UInt = this.toUInt().div(other) /** Divides this value by the other value. */
public inline operator fun div(other: ULong): ULong = this.toULong().div(other.toUInt()) /** Divides this value by the other value. */
public inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other.toUInt()) /** Calculates the remainder of dividing this value by the other value. */
public inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt()) /** Calculates the remainder of dividing this value by the other value. */
public inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other) /** Calculates the remainder of dividing this value by the other value. */
public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other.toUInt()) /** Calculates the remainder of dividing this value by the other value. */
public inline operator fun inc(): UShort = UShort(data.inc()) /** Increments this value. */
public inline operator fun dec(): UShort = UShort(data.dec()) /** Decrements this value. */
operator fun rangeTo(other: UShort): UIntRange = UIntRange(this.toUInt(), other.toUInt()) /** Creates a range from this value to the specified [other] value. */
infix fun and(other: UShort): UShort = UShort(this.data and other.data) /** Performs a bitwise AND operation between the two values. */
infix fun or(other: UShort): UShort = UShort(this.data or other.data) /** Performs a bitwise OR operation between the two values. */
infix fun xor(other: UShort): UShort = UShort(this.data xor other.data) /** Performs a bitwise XOR operation between the two values. */
public inline operator fun xor(other: UShort): UInt = this.toUInt().xor(other.toUInt()) /** Inverts the bits in this value. */
public inline operator fun inv(): UShort = data.inv() /** Inverts the bits in this value. */
public inline operator fun toByte(): Byte = data.toByte() /** Converts this [UShort] value to [Byte]. */
public inline operator fun toShort(): Short = data.toShort() /** Converts this [UShort] value to [Short]. */
public inline operator fun toInt(): Int = data.toInt() and 0xFFFF /** Converts this [UShort] value to [Int]. */
public inline operator fun toLong(): Long = data.toLong() and 0xFFFF /** Converts this [UShort] value to [Long]. */
public inline operator fun toUByte(): UByte = data.toUByte() /** Converts this [UShort] value to [UByte]. */

'UByte' value is represented by the least significant 8 bits of this 'UShort' value.
*/
@kotlin.internal.InlineOnly
public inline fun toUByte(): UByte = data.toUByte() /* Returns this value. */
@kotlin.internal.InlineOnly
public inline fun toUShort(): UShort = this /* Converts this [UShort] value to [UInt]. */
*in* * The resulting 'UInt' value represents the same numerical value as this 'UShort'.
*/
*in* * The least significant 16 bits of the resulting 'UInt' value are the same as the bits of this 'UShort' value,
* whereas the most significant 16 bits are filled with zeros. */
@kotlin.internal.InlineOnly
public inline fun toUInt(): UInt = UInt(data.toInt() and 0xFFFF) /* Converts this [UInt] value to [ULong]. */
*in* * The resulting 'ULong' value represents the same numerical value as this 'UShort'.
*in* * The least significant 16 bits of the resulting 'ULong' value are the same as the bits of this 'UShort' value,
* whereas the most significant 48 bits are filled with zeros. */
@kotlin.internal.InlineOnly
public inline fun toULong(): ULong = ULong(data.toLong() and 0xFFFF) /* Converts this [ULong] value to [Float]. */
*in* * The resulting 'Float' value represents the same numerical value as this 'UShort'. */

@kotlin.internal.InlineOnly
public inline fun toInt(): Int = this.toInt().toShort() /* Converts this [Byte] value to [Int]. */
*in* * The resulting 'Int' value represents the same numerical value as this 'UShort'.
*/
*in* * The least significant 16 bits of the resulting 'Int' value are the same as the bits of this 'Byte' value,
* whereas the most significant 8 bits are filled with the sign bit of this value. */
*in* * Since Kotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun Byte.toUShort(): UShort = UShort(this.toShort()) /* Converts this [Byte] value to [UShort]. */
*in* * If this value is positive, the resulting 'UShort' value represents the same numerical value as this Byte.
*in* * The least significant 8 bits of the resulting 'UShort' value are the same as the bits of this Byte.value,
* whereas the most significant 8 bits are filled with the sign bit of this value. */
*in* * Since Kotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun Short.toUShort(): UShort = UShort(this)
*/
*in* * This value is positive, the resulting 'UShort' value represents the same numerical value as this 'Short'.
*/
*in* * The resulting 'UShort' value has the same binary representation as this 'Short'.
*/
*in* * Since Kotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun Short.toUShort(): UShort = UShort(this.toShort()) /* Converts this [Int] value to [UShort]. */
*in* * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting 'UShort' value represents
* the same numerical value as this 'Int'.
*in* * The resulting 'UShort' value is represented by the least significant 16 bits of this 'Int' value,
* whereas the most significant 8 bits are filled with the sign bit of this value. */
*in* * Since Kotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun Int.toUShort(): UShort = UShort(this.toShort()) /* Converts this [Long] value to [UShort]. */
*in* * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting 'UShort' value represents
* the same numerical value as this 'Long'.
*in* * The resulting 'UShort' value is represented by the least significant 16 bits of this 'Long'.
*/
*in* * Since Kotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun Long.toUShort(): UShort = UShort(this.toShort()) /* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors. */
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file. */

package kotlin.ranges

/**
 * A range of values of type `Char`.
 */
public class CharRange(start: Char, endInclusive: Char) : CharProgression(start, endInclusive, 1), ClosedRange<Char> {
    override val start: Char get() = first
    override val endInclusive: Char get() = last

    override fun contains(value: Char): Boolean = first <= value && value <= last

    override fun isEmpty(): Boolean = first > last

    override fun equals(other: Any?): Boolean = other is CharRange && (isEmpty() && other.isEmpty() || first == other.first && last == other.last)

    override fun override fun contains(value: Char): Boolean = !isEmpty() && last >= value && first <= value

    companion object {
        /** An empty range of values of type `Char`. */
        public val EMPTY: CharRange = CharRange(CharRange(start = CharRange(start = CharRange(start = Char), endInclusive = true), endInclusive = true), endInclusive = true)
    }
}

/**
 * A range of values of type `Int`.
 */
public class IntRange(start: Int, endInclusive: Int) : IntProgression(start, endInclusive, 1), ClosedRange<Int> {
    override val start: Int get() = first
    override val endInclusive: Int get() = last

    override fun contains(value: Int): Boolean = first <= value && value <= last

    companion object {
        /** An empty range of values of type `Int`. */
        public val EMPTY: IntRange = IntRange(IntRange(start = IntRange(start = IntRange(start = Int), endInclusive = true), endInclusive = true), endInclusive = true)
    }
}
other.isEmpty() ||
        first == other.first && last == other.last)

    override fun hashCode(): Int =
        if

    override fun toString(): String = "$first..$last"

    companion object

    /**
     * A range of values of type `Long`.
     * 
     */
    public class LongRange(start: Long, endInclusive: Long) :
        LongProgression(start, endInclusive, 1), ClosedRange<Long> {
        override val start: Long get() = first
        override
        val endInclusive: Long get() = last

        override fun contains(value: Long): Boolean =
            first <= value && value <=
        last

        override fun isEmpty(): Boolean =
            first > last

        override fun equals(other: Any?): Boolean =
            other is LongRange && (isEmpty() && other.isEmpty() ||
            first == other.first && last == other.last)

        override fun hashCode(): Int =
            if

        override fun toString(): String = "$first..$last"

    companion object {
        /** An empty
         * range of values of type Long. 
         */
        public val EMPTY: LongRange = LongRange(1, 0)

    }

    /*
     * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
     * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
     */

    @file:kotlin.jvm.JvmMultifileClass
    @file:kotlin.jvm.JvmName("CollectionsKt")
    @file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)

    package kotlin.collections

    import kotlin.contracts.*
    import 

    internal object EmptyIterator : ListIterator<Nothing> {
        override fun hasNext(): Boolean = false
        override fun hasPrevious(): Boolean = false
        override fun nextIndex(): Int = 0
        override
        fun previousIndex(): Int = -1

        override fun next(): Nothing = throw NoSuchElementException()
        override

    override fun listIterator(index: Int): ListIterator<Nothing> {
        if (index != 0) throw

        override fun listIterator(): ListIterator<Nothing> = EmptyIterator

        override fun subList(fromIndex: Int, toIndex: Int): List<Nothing> {
            if (fromIndex == 0 && toIndex == 0) return this
            throw IndexOutOfBoundsException("fromIndex: $fromIndex, toIndex: $toIndex")
        }

        private fun

        public fun toArray(): Array<out Any?> = values.copyToArrayOfAny(isVarargs)

    /**
     * Returns an empty
     * read-only list.  The returned list is serializable (JVM).
     * * @sample
     * samples.collections.CollectionsLists.emptyList()
     * * Returns a new read-only list of given elements.  The returned list is serializable (JVM).
     * * @sample
     * samples.collections.CollectionsLists.readOnlyList("
     */
    @SinceKotlin("1.1")
    @kotlin.internal.InlineOnly
    public inline fun <T> mutableListOf(): MutableList<T> =
        ArrayList()

    /**
     * Returns an empty new 

    */


samples.collections.Collections.Lists.emptyList

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public inline fun <T> ArrayList<T>: return new [MutableList] with the given elements.

@sample samples.collections.Collections.Lists.mutableList

@SinceKotlin("1.1")
@kotlin.internal.InlineOnly

@sample samples.collections.Collections.Lists.arrayList

/**
 * Returns a new [MutableList] with the given elements.
 * @sample samples.collections.Collections.Lists.mutableListFromInitializer
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly

/**
 * Builds a new read-only list by populating a [MutableList] using the given 
 * [builderAction] and returning a read-only list with the same elements.
 * The list passed as a receiver to the [builderAction] is 
 * valid only inside that function.
 * Using it outside of the function produces an unspecified behavior.
 * @throws IllegalArgumentException if the given [capacity] is negative.
 * @sample samples.collections.Builders.Lists.buildListSample
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal expect inline fun <E> buildList(capacity: Int, @BuilderInference builderAction: MutableList<E>.() -> Unit): return new [MutableList] with the given elements.

/**
 * Returns an [IntRange] of the valid indices for this collection.
 * @sample samples.collections.Collections.Collections.indicesOfCollection
 */
public val Collection<*>.indices: IntRange
get() = 0..size - 1
this.size - 1)\n
/**
 * Returns `true` if the collection is not empty.
 * @sample samples.collections.Collections.Collections.isNotEmpty
 */
@sample samples.collections.Collections.Collections.isNotEmpty

/**
 * Returns `true` if this nullable collection is
 * either null or empty.
 * @sample samples.collections.Collections.Collections.isNullOrEmpty
 */
@SinceKotlin("1.3")
@sample samples.collections.Collections.Collections.isNullOrEmpty

/**
 * Returns this Collection if it's not `null` and the empty list otherwise.
 * @sample samples.collections.Collections.Collections.emptyListOrEmpty
 */
@SinceKotlin("1.3")
@sample samples.collections.Collections.Collections.emptyListOrEmpty

/**
 * Returns this collection if it's not empty
 * or the result of calling [defaultValue] function if the collection is
 * empty.
 * @sample samples.collections.Collections.Collections.emptyListOrEmpty
 */
@SinceKotlin("1.3")
@sample samples.collections.Collections.Collections.emptyListOrEmpty

/**
 * Searches this list or its range for the provided
 * [element] using the binary search algorithm.
 * The list is expected to be sorted into ascending order according to the
 * Comparable natural ordering of its elements, otherwise the result is undefined.
 * If the list contains multiple elements equal to the specified [element], there is no guarantee which one will be found.
 * `null` value is considered to be less than any non-null value.
 * `null` @return the index of the element, if it is contained in the
 * list within the specified range; otherwise, the inverted insertion point `(-insertion point - 1)`.
 * The insertion point is defined as the index at which the element should be inserted;
 * so that the list (or the specified subrange of list) still remains sorted.
 * @sample samples.collections.Collections.Collections.emptyListOrEmpty
 */
@SinceKotlin("1.3")
@sample samples.collections.Collections.Collections.emptyListOrEmpty

/**
 * Searches this list or its range for the provided
 * [element] using the binary search algorithm.
 * The list is expected to be sorted into ascending order according to the
 * specified [comparator], otherwise the result is undefined.
 * Searches this list or its range for the provided [element] using the binary search algorithm.
 * The list is expected to be sorted into ascending order according to the specified [comparator].
 * @sample samples.collections.Collections.Collections.emptyListOrEmpty
 */
@sample samples.collections.Collections.Collections.emptyListOrEmpty
rangeCheck(size, fromIndex, toIndex)\n
var low = fromIndex\nvar high = toIndex - 1\nwhile (low <= high) \nval mid = (low + high).ushr(1) // safe from overflows\nval midVal = get(mid)\nval cmp = comparator.compare(midVal, element)\nif (cmp < 0) \n    low = mid + 1\nelse if (cmp > 0) \n    high = mid - 1\nelse return mid // key found\n\nreturn -(low + 1) // key not found\n
* The list is expected to be sorted into ascending order according to the Comparable natural ordering of keys of its elements.\n* Otherwise the result is undefined.\n* If the list contains multiple elements with the specified [key], there is no guarantee which one will be found.\n* 'null' value is considered to be less than any non-null value.\n* @return the index of the element with the specified [key], if it is contained in the list within the specified range; otherwise, the inverted insertion point `-(insertion point - 1)`.\n* The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted.\n* @sample samples.collections.Collections.Lists.binarySearchByKey

/**
 * Searches this list or its range for an element having the key returned by the specified [selector] function equal to the provided [key] value using the binary search algorithm.\n* The list is expected to be sorted into ascending order according to the Comparable natural ordering of keys of its elements.\n* Otherwise the result is undefined.\n* If the list contains multiple elements with the specified [key], there is no guarantee which one will be found.\n* The list contains multiple elements for which [comparison] returns zero, there is no guarantee which one will be found.\n* @return the index of the element with the specified [key], if it is contained in the list within the specified range; otherwise, the inverted insertion point `-(insertion point - 1)`.\n* The insertion point is defined as the index at which the element should be inserted, so that the list (or the specified subrange of list) still remains sorted.\n* @sample samples.collections.Collections.Lists.binarySearchWithComparisonFunction

private fun rangeCheck(size: Int, fromIndex: Int, toIndex: Int) {
    when {
        fromIndex > toIndex -> throw IllegalArgumentException("fromIndex ($fromIndex) is greater than toIndex ($toIndex).")
        fromIndex < 0 -> throw IndexOutOfBoundsException("fromIndex ($fromIndex) is less than zero.")
        toIndex > size -> throw IndexOutOfBoundsException("toIndex ($toIndex) is greater than size ($size).")
    }
}

internal expect fun checkIndexOverflow(index: Int): Int

internal expect fun checkCountOverflow(count: Int): Int

internal fun throwIndexOverflow() { throw ArithmeticException("Index overflow has happened.") }

internal fun throwCountOverflow() { throw ArithmeticException("Count overflow has happened.") }

"MapsKit"

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. All * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
Open Source Used In DNAC Platform Antelope 13806
EmptyMap : Map<Any?, Nothing>, Serializable {
    private const val serialVersionUID: Long = 
    override fun equals(other: Any?): Boolean = other is Map<*?, *> &&
    other.isEmpty() override fun hashCode(): Int = 
    override fun toString(): String = 
    override val size: Int get() = 0
    override fun isEmpty(): Boolean = true
    override fun containsValue(value: Nothing): Boolean = false
    override fun containsKey(key: Any?): Boolean = false
    override fun get(key: Any?): Nothing? = null
    override val entries: Set<Map.Entry<Any?, Nothing>> = Set() 
}

/**
 * Returns an empty read-only map of specified type.
 * The returned map is serializable.
 */
public fun <K, V> emptyMap(): Map<K, V> = @Suppress("UNCHECKED_CAST") (EmptyMap as Map<K, V>)

/**
 * Returns a new read-only map with the specified contents, given as a list of pairs
 * where the first value is the key and the second is the value.
 * The returned map is serializable.
 */
public fun <K, V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> =
    ConcurrentHashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) }

/**
 * Returns an empty new [MutableMap].
 */
@SinceKotlin("1.1")
public inline fun <K, V> mutableMapOf(): MutableMap<K, V> =
    ConcurrentHashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) }

/**
 * Builds a new [Map] by populating a [MutableMap] using the given [builderAction]
 * and returning a read-only map with the same key-value pairs.
 */
@SinceKotlin("1.3")
public inline fun <K, V> linkedMapOf(): Map<K, V> =
    ConcurrentHashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) }

/*
 * Open Source Used In DNAC Platform Antelope 13807
 */
buildMap(@BuilderInference builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> { 
    contract { 
        callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) 
    } 
    return buildMapInternal(builderAction) 
}

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal expect inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> 
    * @throws IllegalArgument Exception if the given [capacity] is negative.

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal expect inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> 
    * @throws IllegalArgument Exception if the given [capacity] is negative.

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal expect fun <K> mapCapacity(expectedSize: Int): Int

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <K, V> Map<out K, V>.isNotEmpty(): Boolean = !isEmpty()

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <K, V> Map<out K, V>?.isNullOrEmpty(): Boolean {
    contract {
        returns(false) implies (this@isNullOrEmpty != null)
    }
    return this == null || isEmpty()
}

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <K, V> Map<out K, V>?.orEmpty(): Map<K, V> = this ?: emptyMap()

@PublishedApi
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <M, R> M.ifEmpty(defaultValue: () -> R): R where M : Map<*, *>, M : R =
    if (isEmpty())
        defaultValue() else this

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.contains(key: K): Boolean = containsKey(key)

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.get(key: K): V? =
    this.get(key)

@kotlin.internal.InlineOnly
    put(key, value)
}

@kotlin.internal.InlineOnly
public inline fun <K> Map<out K, *>.containsValue(value: V): Boolean =
    this.containsValue(value)

@kotlin.internal.InlineOnly
    return put(key, value)
}

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.get(key: K): V {
    return get(key)
}

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.remove(key: K): V {
    return remove(key)
}

@kotlin.internal.InlineOnly
    this.containsKey(key)

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.containsKey(key: K): Boolean =
    this.containsKey(key)

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.isEmpty(): Boolean =
    this.isEmpty()

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.containsKey(key: K): Boolean =
    this.containsKey(key)

@kotlin.internal.InlineOnly
    this.containsKey(value)

@kotlin.internal.InlineOnly
public inline operator fun <K, V> Map<out K, V>.size(): Int =
    this.size()
* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.

* @return the previous value associated with the key, or `null` if the key was not present in the map.

* Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.
/**
 * Puts all the elements of the given sequence into this [MutableMap] with the first component in the pair being the key and the second the value.
 * @funct public fun <K, V> MutableMap<in K, in V>.putAll(pairs: Sequence<Pair<K, V>>): Unit {
 * Returns a new map with entries having the keys of this map and the values obtained by applying the [transform] function to each entry in this [Map].
 * @sample samples.collections.Maps.Transformations.mapValues
 * @public inline fun <K, V, R> Map<out K, V>.mapValues(transform: (Map.Entry<K, V>) -> R): Map<K, R> {
 * Returns a new Map with entries having the keys obtained by applying the [transform] function to each entry in this [Map] and the values of this map.
 * @sample samples.collections.Maps.Transformations.mapValues
 */

/**
 * Returns a new map containing all key-value pairs matching the given [predicate].
 * @sample samples.collections.Maps.Filtering.filter
 */
    return filterNotTo(LinkedHashMap<K, V>(), predicate)
}

/**
 * Returns a new map containing all key-value pairs matching the given [predicate].
 * @sample samples.collections.Maps.Filtering.filterNot
 */
    return filterNotTo(LinkedHashMap<K, V>(), predicate)
}

/**
 * Returns a map containing all key-value pairs with keys matching the given [predicate].
 * @sample samples.collections.Maps.Transformations.filterKeys
 */
    return filterKeysTo(LinkedHashMap<R, V>(mapCapacity(size)), predicate) // .optimizeReadOnlyMap()
}

/**
 * Returns a map containing all key-value pairs with values matching the given [predicate].
 * @sample samples.collections.Maps.Transformations.filterValues
 */
    return filterValuesTo(LinkedHashMap<K, R>(mapCapacity(size)), predicate) // .optimizeReadOnlyMap()
}

/**
 * Returns a new map containing all key-value pairs matching the given [predicate].
 * @sample samples.collections.Maps.Filtering.filterTo
 */
    for (element in this) {
        if (predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Returns a new map containing all key-value pairs not matching the given [predicate].
 * @sample samples.collections.Maps.Filtering.filterNot
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Appends all entries matching the given [predicate] into the mutable map given as [destination] parameter.
 * @sample samples.collections.Maps.Transformations.mapValues
 */
    val result = LinkedHashMap<K, V>(mapCapacity(size))
    for (entry in this) {
        if (transform(entry) != null) {
            result.put(entry.key, entry.value)
        }
    }
    return result
}

/**
 * Returns a new map containing all key-value pairs not matching the given [predicate].
 * @sample samples.collections.Maps.Filtering.filterNotTo
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Returns a new map containing all key-value pairs from the given collection of pairs.
 * @sample samples.collections.Maps.Transformations.toMap
 */
public fun <K, V> Iterable<Pair<K, V>>.toMap(): Map<K, V> {
    val result = LinkedHashMap<K, V>(mapCapacity(size))
    for (pair in this) {
        result.put(pair.first, pair.second)
    }
    return result
}

/**
 * Returns a new map containing all key-value pairs from the given collection of pairs.
 * @sample samples.collections.Maps.Transformations.toMap
 */
public fun <K, V> Collection<Pair<K, V>>.toMap(): Map<K, V> {
    return toMap(LinkedHashMap<K, V>(mapCapacity(size))}

/**
 * Appends all entries not matching the given [predicate] into the given [destination] mutable map.
 * @sample samples.collections.Maps.Transformations.filterNotTo
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Returns a new map containing all key-value pairs not matching the given [predicate].
 * @sample samples.collections.Maps.Transformations.filterNotTo
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Appends all entries not matching the given [predicate] into the given [destination] mutable map.
 * @sample samples.collections.Maps.Transformations.filterNotTo
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Appends all entries matching the given [predicate] into the mutable map given as [destination] parameter.
 * @sample samples.collections.Maps.Transformations.filterTo
 */
    for (element in this) {
        if (predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Returns a new map containing all key-value pairs not matching the given [predicate].
 * @sample samples.collections.Maps.Transformations.filterNotTo
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Returns a new map containing all key-value pairs from the given collection of pairs.
 * @sample samples.collections.Maps.Transformations.toMap
 */
public fun <K, V> Iterable<Pair<K, V>>.toMap(): Map<K, V> {
    val result = LinkedHashMap<K, V>(mapCapacity(size))
    for (pair in this) {
        result.put(pair.first, pair.second)
    }
    return result
}

/**
 * Returns a new map containing all key-value pairs not matching the given [predicate].
 * @sample samples.collections.Maps.Transformations.filterNotTo
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Appends all entries not matching the given [predicate] into the given [destination] mutable map.
 * @sample samples.collections.Maps.Transformations.filterNotTo
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Appends all entries matching the given [predicate] into the mutable map given as [destination] parameter.
 * @sample samples.collections.Maps.Transformations.filterTo
 */
    for (element in this) {
        if (predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}

/**
 * Returns a new map containing all key-value pairs not matching the given [predicate].
 * @sample samples.collections.Maps.Transformations.filterNotTo
 */
    for (element in this) {
        if (!predicate(element)) {
            destination.put(element.key, element.value)
        }
    }
    return destination
}
```kotlin
putAll(this@toMap) }\n
/**
 * Returns a new map containing all key-value pairs from the given array of pairs.\n * The returned map preserves the entry iteration order of the original array.\n * If any of two pairs would have the same key the last one gets added to the map.\n */
public fun <K, V> Array<out Pair<K, V>>.toMap(): Map<K, V> = when (size) {
    0 -> emptyMap()\n    1 -> mapOf(this[0])\n    else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))\n}\n
/**
 * Returns a new map containing all key-value pairs from the given sequence of pairs.\n * The returned map preserves the entry iteration order of the original sequence.\n */
@SinceKotlin("1.1")
public fun <K, V> Sequence<Pair<K, V>>.toMap(): Map<K, V> = toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()

/**
 * Returns a new map containing all key-value pairs from the given map.\n * The returned map preserves the entry iteration order of the original map.\n */
@SinceKotlin("1.1")
public fun <K, V> Map<out K, V>.toMap(): Map<K, V> = when (size) {
    0 -> emptyMap()\n    1 -> toSingletonMap()\n    else -> toMutableMap()\n}\n
/**
 * Returns a new map containing all key-value pairs from the given iterable collection of pairs.\n */
@kotlin.internal.InlineOnly
    LinkedHashMap(this).apply { putAll(pairs) }\n
/**
 * Returns a new map containing all key-value pairs from the given array of key-value pairs.\n */
@kotlin.internal.InlineOnly
    LinkedHashMap(this).apply { putAll(pairs) }\n
/**
 * Returns a new map containing all key-value pairs from the given sequence of key-value pairs.\n */
@kotlin.internal.InlineOnly
    LinkedHashMap(this).apply { putAll(pairs) }\n
/*...
* The returned map preserves the entry iteration order of the original array.
* If any of two pairs would have the same key the last one gets added to the map.
* Returns a new map containing all key-value pairs from the given array of pairs.
* The returned map preserves the entry iteration order of the original sequence.
* If any of two pairs would have the same key the last one gets added to the map.
*/
```

**Open Source Used In DNAC Platform Antelope 13811**
Array<out Pair<K, V>> { /* putAllpairs */ * Append or replaces all pairs from the given sequence of pairs in this mutable map. */ * Returns a map containing all entries of the original map except the entry with the given [key]. */ * The returned map preserves the entry iteration order of the original map. */ * Returns a map containing all entries of the original map except those entries whose keys are contained in the given [keys] collection. */ * The returned map preserves the entry iteration order of the original map. */ * Returns a map containing all entries of the original map except those entries whose keys are contained in the given [keys] array. */ * The returned map preserves the entry iteration order of the original map. */ * Returns a map containing all entries of the original map except those entries whose keys are contained in the given [keys] sequence. */ * The returned map preserves the entry iteration order of the original map. */ * Removes the entry with the given [key] from this mutable map. */ * Removes all entries the keys of which are contained in the given [keys] collection from this mutable map. */ * Removes all entries the keys of which are contained in the given [keys] array from this mutable map. */ * Removes all entries the keys of which are contained in the given [keys] sequence from this mutable map. */ * @PublishedApi */ internal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) { 0 -> EmptySet 1 -> toSingletonMapOrSelf() else -> this
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SetsKt")@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)

package kotlin.collections

import kotlin.contracts.*

internal object EmptySet : Set<Nothing>, Serializable {
    private const val serialVersionUID: Long = 3406603774387020532

    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()
    override fun hashCode(): Int = 0
    override fun toString(): String = "[]"
    override val size: Int get() = 0
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun iterator(): Iterator<Nothing> = EmptyIterator

    private fun readResolve(): Any = EmptySet
}

/**
 * Returns an empty read-only set. The returned set is serializable (JVM).
 * @sample samples.collections.Collections.SetSetKt
 */
public fun <T> emptySet(): Set<T> = EmptySet

/**
 * Returns a new read-only set with the given elements. Elements of the set are iterated in the order they were specified. The returned set is serializable (JVM).
 */
public fun <T> emptySet(vararg elements: T): Set<T> = EmptySet

// do not expose for now
@PublishedApi
internal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) { 0 -> EmptySet 1 -> toSingletonMapOrSelf() else -> this
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SetsKt")@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)

package kotlin.collections

import kotlin.contracts.*

internal object EmptySet : Set<Nothing>, Serializable {
    private const val serialVersionUID: Long = 3406603774387020532

    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()
    override fun hashCode(): Int = 0
    override fun toString(): String = "[]"
    override val size: Int get() = 0
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun iterator(): Iterator<Nothing> = EmptyIterator

    private fun readResolve(): Any = EmptySet
}

/**
 * Returns an empty read-only set. The returned set is serializable (JVM).
 * @sample samples.collections.Collections.SetSetKt
 */
public fun <T> emptySet(): Set<T> = EmptySet

/**
 * Returns a new read-only set with the given elements. Elements of the set are iterated in the order they were specified. The returned set is serializable (JVM).
 */
public fun <T> emptySet(vararg elements: T): Set<T> = EmptySet

// do not expose for now
@PublishedApi
internal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) { 0 -> EmptySet 1 -> toSingletonMapOrSelf() else -> this
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SetsKt")@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)

package kotlin.collections

import kotlin.contracts.*

internal object EmptySet : Set<Nothing>, Serializable {
    private const val serialVersionUID: Long = 3406603774387020532

    override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()
    override fun hashCode(): Int = 0
    override fun toString(): String = "[]"
    override val size: Int get() = 0
    override fun contains(element: Nothing): Boolean = false
    override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()

    override fun iterator(): Iterator<Nothing> = EmptyIterator

    private fun readResolve(): Any = EmptySet
}

/**
 * Returns an empty read-only set. The returned set is serializable (JVM).
 * @sample samples.collections.Collections.SetSetKt
 */
public fun <T> emptySet(): Set<T> = EmptySet

/**
 * Returns a new read-only set with the given elements. Elements of the set are iterated in the order they were specified. The returned set is serializable (JVM).
 */
public fun <T> emptySet(vararg elements: T): Set<T> = EmptySet

// do not expose for now
@PublishedApi
internal fun <K, V> Map<K, V>.optimizeReadOnlyMap() = when (size) { 0 -> EmptySet 1 -> toSingletonMapOrSelf() else -> this
}
```kotlin
@SinceKotlin("1.1")
public fun String.toLongOrNull(radix: Int): Long? {
    checkRadix(radix)

    val length = this.length
    if (length == 0) return null

    val start: Int
    val isNegative: Boolean
    val limit: Long

    val firstChar = this[0]
    if (firstChar < '0') {  // Possible leading sign
        if (length == 1) return null  // non-digit (possible sign) only, no digits after

        start = 1

        if (firstChar == '-') {
            isNegative = true

            limit = Int.MIN_VALUE
        } else if (firstChar == '+') {
            isNegative = false
            limit = -Int.MAX_VALUE
        } else -> this
    }

    var limitBeforeMul = limit / radix
        } else
            return null
    } else {
        start = 0
        isNegative = false
        limit = Long.MAX_VALUE
    }

    var result = 0
    for (i in start until length) {
        val digit = digitOf(this[i], radix)

        if (digit < 0) return null
        if (result < limitBeforeMul) {
            if (limitBeforeMul == limitForMaxRadix) {

                result *= radix

                if (result < limitBeforeMul) {
                    return null
                }
            } else {
                result -= digit
            }
        }
    }

    return if (isNegative) result else -result
}
```

```kotlin
/**
 * Parses the string as a [Long] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgument onException when [radix] is not a valid radix for string to number
 * conversion.
 */
@SinceKotlin("1.1")
public fun String.toLongOrNull(radix: Int?): return null? if (int < Byte.MIN_VALUE || int > Byte.MAX_VALUE) return null
return int.toLongOrNull(radix = 10)@ann@Inkotlin.internal.InlineOnly
public inline fun String.toLongOrNull(radix: Int): Long? = toLongOrNull(radix)

/**
 * Parses the string as a signed [Byte] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgumentExcept onion when [radix] is not a valid radix for string to number
 * conversion.
 */
@SinceKotlin("1.1")
public fun String.toByteOrNull(radix: Int): Byte? = toByteOrNull(radix = 10)

/**
 * Parses the string as a [Int] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgument onException when [radix] is not a valid radix for string to number
 * conversion.
 */
@SinceKotlin("1.1")
public fun String.toIntOrNull(radix: Int): Int? = toIntOrNull(radix)

/**
 * Parses the string as a [Short] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgument onException when [radix] is not a valid radix for string to number
 * conversion.
 */
@SinceKotlin("1.1")
public fun String.toShortOrNull(radix: Int): Short? = toShortOrNull(radix = 10)

/**
 * Parses the string as an [Int] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgument onException when [radix] is not a valid radix for string to number
 * conversion.
 */
@SinceKotlin("1.1")
public fun String.toByteOrNull(radix: Int): Byte? = toByteOrNull(radix = 10)

/**
 * Parses the string as a signed [Byte] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgument onException when [radix] is not a valid radix for string to number
 * conversion.
 */
@SinceKotlin("1.1")
public fun String.toShortOrNull(radix: Int): Short? = toShortOrNull(radix = 10)
```

---

Open Source Used In DNAC Platform Antelope 13814

- Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
- Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

---

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

- Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
- Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
length) {
  val digit = digitOf(this[i], radix)
  if (digit < 0) return null
  if (result < limitBeforeMul)
    result *= radix
  result += digit
}

if (result < limitBeforeMul) {
  return if (isNegative) result else -result
}

internal fun numberFormatError(input: String): Nothing = throw

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
@file:kotlin.jvm.JvmName("UnsignedKt")
@file:OptIn(ExperimentalUnsignedTypes::class)
package kotlin

@PublishedApi
internal fun uintDivide(v1: Int, v2: Int): Int = (v1 xor Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)

@PublishedApi
internal fun ulongDivide(v1: ULong, v2: ULong): ULong = when {
  v1.isNaN() -> 0u
  v1 > (v1 - Long.MAX_VALUE).toDouble() / v2.toDouble() -> ULong.MAX_VALUE
  else -> (v1.toDouble() / v2.toDouble()).toULong()
}

@PublishedApi
internal fun doubleToUInt(v: Double): UInt = when {
  v.isNaN() -> 0u
  v <= UInt.MIN_VALUE.toDouble() -> UInt.MIN_VALUE
  v >= UInt.MAX_VALUE.toDouble() -> UInt.MAX_VALUE
  v <= Int.MAX_VALUE.toDouble() -> v.toInt().toUInt()
  else -> v.toDouble().toUInt()
}

@PublishedApi
internal fun uintToString(v: ULong): String = uintToString(v, 10)

@PublishedApi
internal fun longToString(v: Long, base: Int): String {
  if (v >= 0) return v.toString(base)
  var quotient = ((v ushr 1) / base) shl 1
  var rem = v - quotient * base
  if (rem >= base) {
    rem -= base
    quotient += 1
  }
  return quotient.toString(base) + rem.toString(base)
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CollectionsKt")
package kotlin.collections

/**
 * Given an [iterator] function constructs an [Iterable] instance that returns values through
the [Iterator] of the given collection if it is known, or `null` otherwise. *

```kotlin
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SequencesKt")

package kotlin.sequences

import kotlin.random.Random

/**
 * Given an [iterator] function constructs a [Sequence]
 * that returns values through the [Iterator]
 * provided by that function. *
 * The values are evaluated lazily, and the sequence is potentially infinite. *
 * @sample samples.collections.Sequences.Building.sequenceFromIterator
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence(crossinline iterator: () -> Iterator<T>): Sequence<T> = object : Sequence<T> {
    override fun iterator(): Iterator<T> = iterator()
}

/**
 * Creates a sequence that returns all elements from this iterator. The sequence is constrained to be iterated only once. *
 * @sample samples.collections.Sequences.Usage.sequenceOrEmpty
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>?.orEmpty(): Sequence<T> = if (this != null) this?.iterator() else EmptySequence

private object EmptySequence : Sequence<Nothing>, DropTakeSequence<Nothing> {
    override fun iterator(): Iterator<Nothing> = EmptyIterator
    override fun drop(n: Int) = EmptySequence
    override fun take(n: Int) = EmptySequence
}
```

*Open Source Used In DNAC Platform Antelope 13816*
Sequence<T> = this ?: emptySequence()

/**
 * Returns a sequence that iterates through the elements either of
 * this sequence or, if this sequence turns out to be empty, of the sequence returned by [defaultValue] function.
 * @sample samples.collections.Sequences.Use.sequenceIfEmpty
 */
public fun <T> Sequence<T>.ifEmpty(defaultValue: () -> Sequence<T>): Sequence<T> = sequence {
    yieldAll(iterator())
}

Sequence<T>.flatten(): Sequence<T> = flatten { it.iterator() } * Returns a sequence of all elements from all iterables in this sequence.

Sequence<T>.shuffled(): Sequence<T> = shuffled(Random) * Returns a sequence that yields elements of this sequence randomly shuffled.

Sequence<T>.shuffled(random: Random): Sequence<T> = sequence<T> {
    val buffer = toMutableList()
    while (buffer.isNotEmpty()) {
        val j = random.nextInt(buffer.size)
        val last = buffer.removeLast()
        val value = if (j < buffer.size) buffer.set(j, last)
    }
}

FilteringSequence<T>(sequence: Sequence<T>, sendWhen: Boolean, predicate: (T) -> Boolean) * Returns a sequence that yields elements of this sequence randomly shuffled * Returns a sequence that yields elements from the underlying sequence that either match or do not match the specified predicate.

TransformingSequence<T, R>(sequence: Sequence<T>, transformer: (T) -> R) * Returns a sequence that yields elements from the underlying sequence that yields elements of this sequence randomly shuffled * Returns a sequence that yields elements of this sequence randomly shuffled * Returns a sequence that yields elements from the underlying sequence that either match or do not match the specified predicate.

@OptIn(ExperimentalStdlibApi::class) buffer.removeLast() * Returns a sequence that yields elements of this sequence randomly shuffled * Returns a sequence that yields elements from the underlying sequence that either match or do not match the specified predicate.

internal class FilteringSequence<T>(sequence: Sequence<T>, sendWhen: Boolean, predicate: (T) -> Boolean) : Sequence<T> {
    override fun iterator(): Iterator<T> = object : Iterator<T> {
        val iterator = sequence.iterator()
        var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue
        var nextItem: T? = null

        private fun calcNext() {
            while (iterator.hasNext()) {
                val item = iterator.next()
                if (predicate(item) == sendWhen) {
                    nextItem = item
                    nextState = 1
                    return
                }
            }
            nextState = 0
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext()
            if (nextState == 0)
                throw NoSuchElementException()
            val result = nextItem
            nextItem = null
            nextState = -1
            @Suppress("UNCHECKED_CAST")
            return result as T
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext()
            return nextState == 1
        }
    }
}

interface FilteringSequence<T, R> {
    fun filter(predicate: (T) -> Boolean): Sequence<T>
}

internal class TransformingSequence<T, R>(sequence: Sequence<T>, transformer: (T) -> R) : Sequence<R> {
    override fun iterator(): Iterator<R> = object : Iterator<R> {
        val iterator = sequence.iterator()
        override fun next(): R {
            return transformer(iterator.next())
        }

        override fun hasNext(): Boolean {
            return iterator.hasNext()
        }
    }
}

interface TransformingSequence<T, R> {
    fun transform(transformer: (T) -> R): Sequence<R>
}

internal class TransformingSequence<T, R>(sequence: Sequence<T>, transformer: (T) -> R) : Sequence<R> {
    override fun iterator(): Iterator<R> = object : Iterator<R> {
        val iterator = sequence.iterator()
        override fun next(): R {
            return transformer(iterator.next())
        }
    }

    override fun hasNext(): Boolean {
        return iterator.hasNext()
    }
}

internal class TransformingSequence<T, R>(sequence: Sequence<T>, transformer: (T) -> R) : Sequence<R> {
    override fun iterator(): Iterator<R> = object : Iterator<R> {
        val iterator = sequence.iterator()
        override fun next(): R {
            return transformer(iterator.next())
        }
    }

    override fun hasNext(): Boolean {
        return iterator.hasNext()
    }
}

interface TransformingSequence<T, R> {
    fun transform(transformer: (T) -> R): Sequence<R>
}

Open Source Used In DNAC Platform Antelope 13817
FlatteningSequence<T, R, E>(sequence, transformer, iterator) A sequence which returns the results of applying the given [transformer] function to the values in the underlying [sequence], where the transformer function takes the index of the value in the underlying sequence along with the value itself. An internal class TransformingIndexedSequence<T, R> unconstructor(private val sequence: Sequence<T>, private val transformer: (T) -> R) : Sequence<R> { override fun iterator(): Iterator<R> = sequence.iterator().let { it var index = 0 it override fun hasNext() = index < sequence.size it override fun next() = transformer(index++, it.next()) } } internal class TransformingIndexedSequence<T, R> constructor(private val sequence: Sequence<T>, private val transformer: (Int, T) -> R) : Sequence<R> { override fun iterator(): Iterator<R> = sequence.iterator().let { it var index = 0 it override fun hasNext() = index < sequence.size it override fun next() = transformer(checkIndexOverflow(index++), it.next()) } } internal class TransformingIndexedSequence<T, R, E> constructor(private val sequence: Sequence<T>, private val transformer: (T) -> R, private val iterator: (R) -> Iterator<E>) : Sequence<E> { override fun iterator(): Iterator<E> = sequence.iterator().let { it var index = 0 it var itemIterator: Iterator<E>? = null it override fun hasNext() = ensureItemIterator() it override fun next() = if (itemIterator == null) throw NoSuchElementException() else itemIterator.next() } private fun ensureItemIterator(): Boolean = if (itemIterator?.hasNext() == false) itemIterator = null else { val element = it.next() val nextItemIterator = iterator(transformer(element)) if (nextItemIterator.hasNext()) itemIterator = nextItemIterator return true } internal class FlatteningSequence<T, R, E>(sequence, transformer, iterator: (R) -> Iterator<E>): Sequence<E> { override fun iterator(): Iterator<E> = sequence.iterator().let { it var index = 0 it var itemIterator: Iterator<E>? = null it override fun hasNext() = ensureItemIterator() it override fun next() = if (itemIterator == null) throw NoSuchElementException() else if (!itemIterator.hasNext()) return false itemIterator.next() } private fun ensureItemIterator(): Boolean = if (itemIterator?.hasNext() == false) itemIterator = null else { val element = it.next() val nextItemIterator = iterator(transformer(element)) if (nextItemIterator.hasNext()) itemIterator = nextItemIterator return true } internal fun <T, C, R> flatMapIndexed(source: Sequence<T>, transform: (Int, T) -> C, iterator: (C) -> Iterator<R>): Sequence<R> = sequence { var index = 0 for (element in source) yieldAll(transform(index++), element, iterator) }
override fun hasNext(): Boolean {
    drop() // return (position < endIndex) && iterator.hasNext()
    drop() // if (position == endIndex) throw NoSuchElementException()
    position++
    return iterator.next() || throw NoSuchElementException()
}

override fun next(): T {
    drop()
    if (position == endIndex) throw NoSuchElementException()
    return iterator.next() || throw NoSuchElementException()
}

@Suppress("UNCHECKED_CAST")
    val result = nextItem as T

    // Clean next to avoid keeping reference on yielded instance
    nextItem = null
    nextState = -1
    return result
}

override fun hasNext(): Boolean {
    if (nextState == -1)
        calcNext() // will change nextState
    return nextState == 1
}

override fun next(): T {
    if (nextState == -1)
        calcNext() // will change nextState
    return iterator.next()
}

override fun hasNext(): Boolean {
    if (nextState == -1)
        calcNext() // will change nextState
    return (position < endIndex) && iterator.hasNext()
    if (position == endIndex) throw NoSuchElementException()
    position++
    return iterator.next() || throw NoSuchElementException()
}

@Suppress("UNCHECKED_CAST")
            val result = nextItem as T

            // Clean next to avoid keeping reference on yielded instance
            nextItem = null
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return nextState == 1
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext() // will change nextState
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return (position < endIndex) && iterator.hasNext()
            if (position == endIndex) throw NoSuchElementException()
            position++
            return iterator.next() || throw NoSuchElementException()
        }

        @Suppress("UNCHECKED_CAST")
            val result = nextItem as T

            // Clean next to avoid keeping reference on yielded instance
            nextItem = null
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return nextState == 1
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext() // will change nextState
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return (position < endIndex) && iterator.hasNext()
            if (position == endIndex) throw NoSuchElementException()
            position++
            return iterator.next() || throw NoSuchElementException()
        }

        @Suppress("UNCHECKED_CAST")
            val result = nextItem as T

            // Clean next to avoid keeping reference on yielded instance
            nextItem = null
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return nextState == 1
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext() // will change nextState
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return (position < endIndex) && iterator.hasNext()
            if (position == endIndex) throw NoSuchElementException()
            position++
            return iterator.next() || throw NoSuchElementException()
        }

        @Suppress("UNCHECKED_CAST")
            val result = nextItem as T

            // Clean next to avoid keeping reference on yielded instance
            nextItem = null
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return nextState == 1
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext() // will change nextState
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return (position < endIndex) && iterator.hasNext()
            if (position == endIndex) throw NoSuchElementException()
            position++
            return iterator.next() || throw NoSuchElementException()
        }

        @Suppress("UNCHECKED_CAST")
            val result = nextItem as T

            // Clean next to avoid keeping reference on yielded instance
            nextItem = null
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return nextState == 1
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext() // will change nextState
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return (position < endIndex) && iterator.hasNext()
            if (position == endIndex) throw NoSuchElementException()
            position++
            return iterator.next() || throw NoSuchElementException()
        }

        @Suppress("UNCHECKED_CAST")
            val result = nextItem as T

            // Clean next to avoid keeping reference on yielded instance
            nextItem = null
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return nextState == 1
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext() // will change nextState
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return (position < endIndex) && iterator.hasNext()
            if (position == endIndex) throw NoSuchElementException()
            position++
            return iterator.next() || throw NoSuchElementException()
        }

        @Suppress("UNCHECKED_CAST")
            val result = nextItem as T

            // Clean next to avoid keeping reference on yielded instance
            nextItem = null
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return nextState == 1
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext() // will change nextState
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return (position < endIndex) && iterator.hasNext()
            if (position == endIndex) throw NoSuchElementException()
            position++
            return iterator.next() || throw NoSuchElementException()
        }

        @Suppress("UNCHECKED_CAST")
            val result = nextItem as T

            // Clean next to avoid keeping reference on yielded instance
            nextItem = null
            nextState = -1
            return result
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return nextState == 1
        }

        override fun next(): T {
            if (nextState == -1)
                calcNext() // will change nextState
            return iterator.next()
        }

        override fun hasNext(): Boolean {
            if (nextState == -1)
                calcNext() // will change nextState
            return (position < endIndex) && iterator.hasNext()
            if (position == endIndex) throw NoSuchElementException()
            position++
            return iterator.next() || throw NoSuchElementException()
*\@kotlin.internal.InlineOnly\@public inline fun require(value: Boolean, lazyMessage: () -> Any): Unit {\n    contract { \n        returns() implies value \n    } \n    if (!value) \n        throw IllegalArgumentException(lazyMessage().toString())
}

/**
 * Throws an [IllegalArgumentException] if the [value] is null.
 * Otherwise returns the not null value.
 * @sample samples.misc.Preconditions.failRequireWithLazyMessage
 */
@kotlin.internal.InlineOnly
    contract { \n        returns() implies (value != null) \n    } \n    if (value == null) \n        throw IllegalArgumentException(lazyMessage().toString())
    return value
}

/**
 * Throws an [IllegalStateException] if the [value] is false.
 * @sample samples.misc.Preconditions.failCheckWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun check(value: Boolean, lazyMessage: () -> Any): Unit {
    contract { \n        returns() implies value \n    } \n    if (!value) {
        val message = lazyMessage()
        throw IllegalStateException(message.toString())
    }
}

/**
 * Throws an [IllegalStateException] if the [value] is null.
 * Otherwise returns the not null value.
 * @sample samples.misc.Preconditions.failCheckWithLazyMessage
 */
@kotlin.internal.InlineOnly
public inline fun checkNotNull(value: T?): T {
    contract { \n        returns() implies (value != null) \n    } \n    return value
}

/**
 * Throws an [IllegalStateException] if the [value] is null. Otherwise returns the not null value.
 * @sample samples.misc.Preconditions.failCheckWithLazyMessage
 */
@kotlin.internal.InlineOnly
    contract { \n        returns() implies (value != null) \n    } \n    if (value == null) {
        val message = lazyMessage()
        throw IllegalStateException(message.toString())
    } else {
        return value
    }
}

/**
 * Throws an [IllegalStateException] with the given [message].
 * @sample samples.misc.Preconditions.failWithError
 */
@kotlin.internal.InlineOnly
public inline fun error(message: Any): Nothing = throw IllegalStateException(message.toString())

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.js.*
import primitiveArrayConcat
import withType
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun <T> Array<out T>.elementAt(index: Int): T {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun ByteArray.elementAt(index: Int): Byte {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun ShortArray.elementAt(index: Int): Short {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public actual fun IntArray.elementAt(index: Int): Int {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

private fun Array<out T>.elementAtOrElse(index: Int, message: () -> Any): T {
    return when {
        index < 0 -> throw IndexOutOfBoundsException("index: $index, size: $size")
        index >= size -> throw IndexOutOfBoundsException("index: $index, size: $size")
        else -> { message() } // Use type safety to make sure message() is a function
    }
}
IndexOutOfBoundsException("index: Sindex, size: $size\}) ;n\n/**n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * @return elementAtOrElse(index) \n */\n
/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun Array<out T>.asList(): List<T> {
    return ArrayList<T>(this.unsafeCast<Array<Any?>>())
}

/**\n * Returns a [List] that wraps the original array.\n * @sample samples.collections.Collections.ElementArray\n */\npublic actual fun <T> List<T>.contentDeepEquals(other: List<T>): Boolean {
    return
}
this.contentDeepEquals(other)|null/*null * Returns `true` if the two specified arrays are *deeply* equal to one another,\n* i.e. contain the same number of the same elements in the same order.\n* The specified arrays are also considered deeply equal if both are `null`.\n* If two corresponding elements are nested arrays, they are also compared deeply.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n* The elements of other types are compared for equality with the `equals<Any.equals>` function.\n* For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepEquals")\npublic actual infix fun <T> Array<T>.contentDeepEquals(other: Array<T>): Boolean \n*\nreturns a hash code based on the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepHashCode")\npublic actual fun <T> Array<T>.contentDeepHashCode(): Int \n*\nreturns a hash code based on the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepToString")\npublic actual fun <T> Array<T>.contentDeepToString(): String \n*\nreturns a string representation of the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepToString")\npublic actual fun <T> Array<T>.contentDeepToString(): String \n*\nreturns a string representation of the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepToString")\npublic actual fun <T> Array<T>.contentDeepToString(): String \n*\nreturns a string representation of the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepToString")\npublic actual fun <T> Array<T>.contentDeepToString(): String \n*\nreturns a string representation of the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepToString")\npublic actual fun <T> Array<T>.contentDeepToString(): String \n*\nreturns a string representation of the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepToString")\npublic actual fun <T> Array<T>.contentDeepToString(): String \n*\nreturns a string representation of the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.\n
*\n@SinceKotlin("1.4.1")\n@library("arrayDeepToString")\npublic actual fun <T> Array<T>.contentDeepToString(): String \n*\nreturns a string representation of the contents of this array as if it is a [List].\n* Nested arrays are treated as lists too.\n* If any of arrays contains itself on any nesting level the behavior is undefined.
IntArray.contentEquals(other: IntArray): Boolean { return this.contentEquals(other) } * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the [equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. *

@ Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@ Since Kotlin("1.1")
public actual infix fun LongArray.contentEquals(other: LongArray): Boolean { return this.contentEquals(other) } * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the [equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. *

@ Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@ Since Kotlin("1.1")
public actual infix fun FloatArray.contentEquals(other: FloatArray): Boolean { return this.contentEquals(other) } * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the [equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. *

@ Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@ Since Kotlin("1.1")
public actual infix fun DoubleArray.contentEquals(other: DoubleArray): Boolean { return this.contentEquals(other) } * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the [equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. *

@ Since Kotlin("1.4")
@ library("arrayEquals")
public actual infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean { definedExternally } * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the [equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. *

@ Since Kotlin("1.4")
@ library("arrayEquals")
public actual infix fun ByteArray?.contentEquals(other: ByteArray?): Boolean { definedExternally } * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the [equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. *

@ Since Kotlin("1.4")
@ library("arrayEquals")
public actual infix fun ShortArray?.contentEquals(other: ShortArray?): Boolean { definedExternally } * Returns `true` if the two specified arrays are *structurally* equal to one another, i.e. contain the same number of the same elements in the same order. * The elements are compared for equality with the [equals] function. * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`. *
means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.

*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual infix fun IntArray?.contentEquals(other: IntArray?): Boolean {\n    definedExternally\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual infix fun LongArray?.contentEquals(other: LongArray?): Boolean {\n    definedExternally\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual infix fun FloatArray?.contentEquals(other: FloatArray?): Boolean {\n    definedExternally\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual infix fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean {\n    definedExternally\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean {\n    definedExternally\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual infix fun CharArray?.contentEquals(other: CharArray?): Boolean {\n    definedExternally\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual fun IntArray?.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual fun LongArray?.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual fun FloatArray?.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual fun DoubleArray?.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual fun BooleanArray?.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual fun CharArray?.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\n*fn@Since Kotlin(`1.4`)*\n@library(`arrayEquals`)\npublic actual fun CharArray?.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n
\nOpen Source Used In DNAC Platform Antelope 13825
DoubleArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\nBooleanArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n\nByteArray.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n
ByteArray?.contentHashCode(): Int {\n    return this.contentHashCode()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {\n    return this.contentToString()\n}\n
ByteArray?.contentToString(): String {
actual fun FloatArray.contentToString(): String { return this.contentToString() } Returns a string representation of the contents of the specified array as if it is [List].

samples.collections.Arrays.ContentOperations.contentToString

@SinceKotlin("1.4")
public fun DoubleArray.contentToString(): String { return this.contentToString() } Returns a string representation of the contents of the specified array as if it is [List].

samples.collections.Arrays.ContentOperations.contentToString

@SinceKotlin("1.4")
public fun BooleanArray.contentToString(): String { return this.contentToString() } Returns a string representation of the contents of the specified array as if it is [List].

samples.collections.Arrays.ContentOperations.contentToString

@SinceKotlin("1.4")
public fun CharArray.contentToString(): String {
    return this.contentToString()
}

/**
 * Returns a string representation of the contents of the specified array as if it is [List].
 * 
 * @sample samples.collections.Arrays.ContentOperations.contentToString
 */
@SinceKotlin("1.4")
@library("arrayToString")
public actual fun <T> Array<out T>?.contentToString(): String {
    definedExternally
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it
 * overlaps with the destination range.
 * 
 * @param destination the array to copy to.
 * @param destinationOffset the position in the [destination] array to copy to.
 * @param param start and endIndex the beginning (inclusive) of the subrange to copy, size of this
 * array by default.
 * 
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.
 * 
 * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified

---

Open Source Used In DNAC Platform Antelope 13827
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T> {
    arrayCopy(this, destination, destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It’s allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.\n * @return the [destination] array.\n */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray {
    arrayCopy(this.unsafeCast<Array<Byte>>(), destination.unsafeCast<Array<Byte>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It’s allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.\n * @return the [destination] array.\n */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray {
    arrayCopy(this.unsafeCast<Array<Short>>(), destination.unsafeCast<Array<Short>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It’s allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.\n * @return the [destination] array.\n */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray {
    arrayCopy(this.unsafeCast<Array<Int>>(), destination.unsafeCast<Array<Int>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It’s allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.\n * @return the [destination] array.\n */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray {
    arrayCopy(this.unsafeCast<Array<Int>>(), destination.unsafeCast<Array<Int>>(), destinationOffset, startIndex, endIndex)
    return destination
}
copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

@throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*param destination the array to copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

@throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*param destination the array to copy to, 0 by default.

*param destination the array to copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

*throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*param destination the array to copy to, 0 by default.

*param destination the array to copy to, 0 by default.

*param destination the array to copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

@throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*param destination the array to copy to, 0 by default.

*param destination the array to copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

@throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*param destination the array to copy to, 0 by default.

*param destination the array to copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

@throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*param destination the array to copy to, 0 by default.

*param destination the array to copy to, 0 by default.
arrayCopy(this.unsafeCast<Array<Char>>(), destination.unsafeCast<Array<Char>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.
 * @param destination the array to copy to.
 * @param destinationOffset the position in the [destination] array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.
 * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.
 * @throws IndexOutOfBoundsException or IllegalArgumentException when [startIndex] or [endIndex] is out of range of this array indices or when [startIndex] > [endIndex].
 * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset], or when that index is out of the [destination] array indices range.
 * @return the [destination] array.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual inline fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray {
    arrayCopy(this.unsafeCast<Array<Char>>(), destination.unsafeCast<Array<Char>>(), destinationOffset, startIndex, endIndex)
    return destination
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline fun <T> Array<out T>.copyOf(): Array<T> {
    return this.asDynamic().slice()
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline fun ByteArray.copyOf(): ByteArray {
    return this.asDynamic().slice()
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline fun ShortArray.copyOf(): ShortArray {
    return this.asDynamic().slice()
}

/**
 * Returns new array which is a copy of the original array.
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline fun IntArray.copyOf(): IntArray {
    return this.asDynamic().slice()
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public actual fun LongArray.copyOf(newSize: Int): LongArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, ByteArray(newSize))
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public actual fun FloatArray.copyOf(newSize: Int): FloatArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, ByteArray(newSize))
}

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize]. The copy is either truncated or padded at the end with zero values if necessary.
 * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf
 */
public actual fun DoubleArray.copyOf(newSize: Int): DoubleArray {
    require(newSize >= 0) { "Invalid new array size: $newSize." }
    return fillFrom(this, ByteArray(newSize))
}
necessary. If [newSize] is less than the size of the original array, the copy array is truncated to the
[newSize]. If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with zero values.

* - If [newSize] is less than the size of the original array, the copy array is resized to the given [newSize].
* The copy is either truncated or paddted at the end with zero values if necessary.
* - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with zero values.

samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf

* - If [newSize] is less than the size of the original array, the copy array is resized to the given [newSize].
* The copy is either truncated or paddted at the end with zero values if necessary. If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with zero values.

samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf

IntArray.copyOf(newSize: Int): IntArray

* - If [newSize] is less than the size of the original array, the copy array is resized to the given [newSize].
* The copy is either truncated or paddted at the end with zero values if necessary.

LongArray.copyOf(newSize: Int): LongArray

* - If [newSize] is less than the size of the original array, the copy array is resized to the given [newSize].
* The copy is either truncated or paddted at the end with null char (` `) values if necessary. If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with null char (` `) values.

samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf

BooleanArray.copyOf(newSize: Int): BooleanArray

* - If [newSize] is less than the size of the original array, the copy array is resized to the given [newSize].
* The copy is either truncated or paddted at the end with `null` values if necessary. If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original array, the extra elements in the copy array are
filled with `null` values.
copy array are filled with `null` values.

```kotlin
*\n* @sample
samples.Collections.Arrays.CopyOfOperationsresizingCopyOf\n/*\n@Suppress("ACTUAL_WITHOUT_EXPECT")\npublic actual fun <T> Array<out T>.copyOf(newSize: Int):
Array<T?> { \n   require(newSize >= 0) { "Invalid new array size: $newSize." }\n   return arrayCopyResize(this, newSize, null)\n}\n\n* @param fromIndex the start of the range (inclusive) to copy.\n* @param toIndex the end of the range (exclusive) to copy.\n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n*\n@Suppress("ACTUAL_WITHOUT_EXPECT")\npublic actual fun <T> Array<out T>.copyOf(newSize: Int):
Array<T?> { \n    require(newSize >= 0) { "Invalid new array size: $newSize." }\n    return arrayCopyResize(this, newSize, null)\n}\n```

```kotlin
/**
 * Returns a new array which is a copy of the specified range of the original array.\n* @param fromIndex the start of the range (inclusive) to copy.\n* @param toIndex the end of the range (exclusive) to copy.\n* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n*/
```
@param fromIndex the start of the range (inclusive) to copy.
* @param toIndex the end of the range (exclusive) to copy.
* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
* @throws IllegallegalArgumentException if [fromIndex] is greater than [toIndex].

/**
 * Returns a new array which is a copy of the specified range of the original array.
* @param fromIndex the start of the range (inclusive) to copy.
* @param toIndex the end of the range (exclusive) to copy.
* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
* @throws IllegalArgumentsException if [fromIndex] is greater than [toIndex].
 */
public actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return withType("BooleanArray", this.asDynamic().slice(fromIndex, toIndex))
}

/**
 * Returns a new array which is a copy of the specified range of the original array.
 * @param fromIndex the start of the range (inclusive) to copy.
 * @param toIndex the end of the range (exclusive) to copy.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
actual fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    return withType("CharArray", this.asDynamic().slice(fromIndex, toIndex))
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentsException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentsException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
actual fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentsException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentsException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * 
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    this.asDynamic().fill(element, fromIndex, toIndex)
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("ACTUAL_WITHOUT_EXPECT", "NOTHING_TO_INLINE")
public actual inline operator fun <T> Array<out T>.plus(element: T): Array<T> {
    return this.asDynamic().concat(arrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ByteArray.plus(element: Byte): ByteArray {
    return plus(byteArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun ShortArray.plus(element: Short): ShortArray {
    return plus(shortArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun IntArray.plus(element: Int): IntArray {
    return plus(intArrayOf(element))
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun LongArray.plus(element: Long):
LongArray {\n    return plus(longArrayOf(element))\n} /*/n/**/n * Returns an array containing all elements of the original array and then the given [element].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun FloatArray.plus(element: Float): FloatArray {\n    return plus(floatArrayOf(element))\n} /*/n/**/n * Returns an array containing all elements of the original array and then the given [element].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(element: Double): DoubleArray {\n    return plus(doubleArrayOf(element))\n} /*/n/**/n * Returns an array containing all elements of the original array and then the given [element].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun BooleanArray.plus(element: Boolean): BooleanArray {\n    return plus(booleanArrayOf(element))\n} /*/n/**/n * Returns an array containing all elements of the original array and then the given [element].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun CharArray.plus(element: Char): CharArray {\n    return plus(charArrayOf(element))\n} /*/n/**/n * Returns an array containing all elements of the original array and then the given [elements].\n} */n@Suppress("ACTUALWITHOUTEXPECTCTY")\npublic actual inline operator fun <T> Array<T>.plus(elements: Collection<T>): Array<T> {\n    return arrayPlusCollection(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray {\n    return fillFromCollection(this.copyOf(size + elements.size), size, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("ACTUALWITHOUTEXPECTCTY")\npublic actual inline operator fun IntArray.plus(elements: Collection<Int>): IntArray {\n    return fillFromCollection(this.copyOf(size + elements.size), size, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(elements: Collection<Long>): LongArray {\n    return arrayPlusCollection(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun FloatArray.plus(elements: Collection<Float>): FloatArray {\n    return fillFromCollection(this.copyOf(size + elements.size), size, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray {\n    return fillFromCollection(this.copyOf(size + elements.size), size, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray {\n    return fillFromCollection(this.copyOf(size + elements.size), size, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun CharArray.plus(elements: Collection<Char>): CharArray {\n    return fillFromCollection(this.copyOf(size + elements.size), size, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun <T> Array<T>.plus(elements: Array<T>): Array<T> {\n    return this.asDynamic().concat(elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ByteArray.plus(elements: ByteArray): ByteArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun IntArray.plus(elements: IntArray): IntArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(elements: LongArray): LongArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun CharArray.plus(elements: CharArray): CharArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun <T> Array<T>.plus(elements: Array<T>): Array<T> {\n    return this.asDynamic().concat(elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ByteArray.plus(elements: ByteArray): ByteArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun IntArray.plus(elements: IntArray): IntArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(elements: LongArray): LongArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun CharArray.plus(elements: CharArray): CharArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun <T> Array<T>.plus(elements: Array<T>): Array<T> {\n    return this.asDynamic().concat(elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun ByteArray.plus(elements: ByteArray): ByteArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun IntArray.plus(elements: IntArray): IntArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")\npublic actual inline operator fun LongArray.plus(elements: LongArray): LongArray {\n    return primitiveArrayConcat(this, elements)\n} /*/n/**/n * Returns an array containing all elements of the original array and then all elements of the given [elements].\n} */n@Suppress("NOTHING_TO_INLINE")

Open Source Used In DNAC Platform Antelope 13835
IntArray { return primitiveArrayConcat(this, elements); } * Returns an array containing all elements of the original array and then all elements of the given [elements] array.

*\@Suppress("NOTHING_TO_INLINE")\*public actual inline operator fun LongArray.plus(elements: LongArray): LongArray {
    return primitiveArrayConcat(this, elements);
}

/**
 * Returns an array containing all elements of the original array and then all elements of the given [elements] array.
 */
@Suppress("NOTHING_TO_INLINE")
public actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {
    return primitiveArrayConcat(this, elements);
}

*\@Suppress("NOTHING_TO_INLINE")\*public actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {
    return primitiveArrayConcat(this, elements);
}

*\@Suppress("NOTHING_TO_INLINE")\*public actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {
    return primitiveArrayConcat(this, elements);
}

*\@Suppress("NOTHING_TO_INLINE")\*public actual inline operator fun CharArray.plus(elements: CharArray): CharArray {
    return primitiveArrayConcat(this, elements);
}

public actual inline fun <T> Array<out T>.plusElement(element: T): Array<T> {
    return this.asDynamic().concat(arrayOf(element))
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun IntArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
public actual fun LongArray.sort(): Unit {
    if (size > 1) sort { a: Long, b: Long -> a.compareTo(b) }
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun ByteArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun ShortArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun DoubleArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@library("primitiveArraySort")
public actual fun FloatArray.sort(): Unit {
    definedExternally
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
public actual fun CharArray.sort(): Unit {
    definedExternally
}

public actual fun <T : Comparable<T>> Array<out T>.sort(): Unit {
    if (size > 1) sortArray(this)
}

/**
 * Sorts a range in the array in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * 
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * 
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * 
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 * 
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable
 */
@SinceKotlin("1.4")
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    definedExternally
}
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n\nsortArrayWith(this, fromIndex, toIndex, naturalOrder())\n\n**\n
Sorts a range in the array in-place.\n\n@param fromIndex the start of the range (inclusive) to sort, size of this array by default.\n\n@throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n\n@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n\n@sample samples.collections.Arrays.Sorting.sortRangeOfArray

*\n
@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n
public actual fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ByteArray>()\n    subarray.sort()\n}

/**
 * Sorts a range in the array in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * 
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray

*\n
@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n
public actual fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ShortArray>()\n    subarray.sort()\n}

/**
 * Sorts a range in the array in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * 
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray

*\n
@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n
public actual fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<IntArray>()\n    subarray.sort()\n}

/**
 * Sorts a range in the array in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * 
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray

*\n
@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n
public actual fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    sortArrayWith(this.unsafeCast<Array<Long>>(), fromIndex, toIndex, naturalOrder())\n    \n\n\n}

/**
 * Sorts a range in the array in-place.
 * 
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * 
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray

*\n
@SinceKotlin("1.4")\n@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")\n
public actual fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<FloatArray>()\n    subarray.sort()\n}
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].

/*@SinceKotlin("1.4")@Suppress("\'ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS'\")*/
actual fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<DoubleArray>()
    subarray.sort()}

/*@SinceKotlin("1.4")@Suppress("\'ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS'\")*/
actual fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<CharArray>()
    subarray.sort()}

@kotlin.internal.InlineOnly
global inline fun ByteArray.sort(noinline comparison: (a: Byte, b: Byte) -> Int): Unit {
    asDynamic().sort(comparison)
}

@kotlin.internal.InlineOnly
global inline fun ShortArray.sort(noinline comparison: (a: Short, b: Short) -> Int): Unit {
    asDynamic().sort(comparison)
}

@kotlin.internal.InlineOnly
global inline fun IntArray.sort(noinline comparison: (a: Int, b: Int) -> Int): Unit {
    asDynamic().sort(comparison)
}

@kotlin.internal.InlineOnly
global inline fun LongArray.sort(noinline comparison: (a: Long, b: Long) -> Int): Unit {
    asDynamic().sort(comparison)
}

@kotlin.internal.InlineOnly
global inline fun FloatArray.sort(noinline comparison: (a: Float, b: Float) -> Int): Unit {
    asDynamic().sort(comparison)
}

@kotlin.internal.InlineOnly
global inline fun DoubleArray.sort(noinline comparison: (a: Double, b: Double) -> Int): Unit {
    asDynamic().sort(comparison)
}

@kotlin.internal.InlineOnly
global inline fun CharArray.sort(noinline comparison: (a: Char, b: Char) -> Int): Unit {
    asDynamic().sort(comparison)
}

/*@SinceKotlin("1.4")@Suppress("\'ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS'\")*/
public actual fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<CharArray>()
    subarray.sort()}

/*@SinceKotlin("1.4")@Suppress("\'ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS'\")*/
public actual fun ByteArray.toTypedArray(): Array<Byte> {
    return js("[
    ]").slice.call(this)
}

/*@SinceKotlin("1.4")@Suppress("\'ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS'\")*/
public actual fun ShortArray.toTypedArray(): Array<Short> {
    return js("[
    ]").slice.call(this)
}
object array containing all of the elements of this primitive array.

*/
public actual fun IntArray.toTypedArray(): Array<Int> {
    return js("[
    ]").slice.call(this)
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive
array.
*/
public actual fun LongArray.toTypedArray(): Array<Long> {
    return js("[
    ]").slice.call(this)
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive
array.
*/
public actual fun FloatArray.toTypedArray(): Array<Float> {
    return js("[
    ]").slice.call(this)
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive
array.
*/
public actual fun DoubleArray.toTypedArray(): Array<Double> {
    return js("[
    ]").slice.call(this)
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive
array.
*/
public actual fun BooleanArray.toTypedArray(): Array<Boolean> {
    return js("[
    ]").slice.call(this)
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive
array.
*/
public actual fun CharArray.toTypedArray(): Array<Char> {
    return Array(size) { index ->
        this[index]
    }
}

// /*
// * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
// * Use of this source code is governed by the Apache 2.0 license that can be found in
the license/LICENSE.txt file.
// */
@file:kotlin.jvm.JvmName("ComparisonsKt")
@file:kotlin.jvm.JvmMultifileClass

package

/**
 * Compares two values using the specified functions [selectors] to calculate the result
of the comparison.
* The functions are called sequentially, receive the given values [a] and [b] and return
[Comparable] objects. As soon as the [Comparable] instances returned by a function for [a] and [b] values do
not compare as equal, the result of that comparison is returned.
*/
public fun <T> compareValuesBy(a: T, b: T, vararg selectors: (T) -> Comparable<*>?): Int {
    require(selectors.size > 0)
    return compareValuesByImpl(a, b, selectors)
}

private fun <T> compareValuesByImpl(a: T, b: T, selectors: Array<out (T) -> Comparable<*>?>): Int {
    for (fn in selectors) {
        val v1 = fn(a)
        val v2 = fn(b)
        val diff = compareValues(v1, v2)
        if (diff != 0) return diff
    }
    return 0
}

/**
 * Compares two values using the specified [selector] function to calculate the result of the comparison.
* The function is applied to the given values [a] and [b] and return [Comparable] objects.
* The result of comparison of these [Comparable] instances is returned.
*/
@kotlin.internal.InlineOnly
public inline fun <T> compareValuesBy(a: T, b: T, selector: (T) -> Comparable<*>?): Int {
    return compareValues(selector(a), selector(b))
}

/**
 * Compares two values using the specified [selector] function to calculate the result of the comparison.
* The function is applied to the given values [a] and [b] and return objects of type K which are then being compared with the given
[comparator].
*/
@kotlin.internal.InlineOnly
public inline fun <T, K> compareValuesBy(a: T, b: T, comparator: Comparator<in K>, selector: (T) -> K): Int {
    return comparator.compare(selector(a), selector(b))
}

// Not so useful without type inference for receiver of expression
//@Suppress("NOTHING_TO_INLINE")
public inline fun <T> compareWith(a: T, b: T, comparator: Comparator<T>): Int = comparator.compare(a, b)

/**
 * Compares two nullable [Comparable] values. Null is considered less than any value.
*/
public fun <T : Comparable<*>> compareValues(a: T?, b: T?): Int {
    if (a === b) return 0
    if (a == null) return -1
    if (b == null) return 1

    @Suppress("UNCHECKED_CAST")
    return (a as Comparable<Any>).compareTo(b)
}

/**
 * Creates a comparator using the sequence of functions to calculate a result of comparison.
* The functions are called sequentially, receive the given values `a` and `b` and return
[Comparable] objects. As soon as the [Comparable] instances returned by a function for `a` and `b` values do
not compare as equal, the result of that comparison is returned from the [Comparator]. The function to transform value
* creates a comparator using the function to transform value
*/

to a [Comparable] instance for comparison.

```kotlin
public inline fun <T> compareBy(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =
    Comparator { a, b ->
        compareValuesBy(a, b, selector) }

public inline fun <T, K> compareBy(compare: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =
    Comparator { a, b ->
        compareValuesBy(a, b, selector) }
```

Creates a descending comparator using the [selector] function to transform values being compared and then applying the specified [comparator] to compare transformed values.

```kotlin
public inline fun <T> compareByDescending(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =
    Comparator { a, b ->
        compareValuesBy(b, a, selector) }
```

Creates a descending comparator using the [selector] function to transform values being compared and then applying the specified [comparator] to compare transformed values.

```kotlin
public inline fun <T, K> compareByDescending(compare: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =
    Comparator { a, b ->
        compareValuesBy(b, a, selector) }
```

Creates a descending comparator using the [selector] function to transform values being compared and then applying the specified [comparator] to compare transformed values.

```kotlin
public inline fun <T> Comparator<T>.thenBy(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenBy.compare(a, b)
        if (previousCompare != 0) previousCompare else
go inside function
```

Creates a descending comparator using the [selector] function to transform values being compared and then applying the specified [comparator] to compare transformed values.

```kotlin
public inline fun <T, K> Comparator<T>.thenBy(compare: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenBy.compare(a, b)
        if (previousCompare != 0) previousCompare else
go inside function
```

Creates a descending comparator using the [selector] function to transform values being compared and then applying the specified [comparator] to compare transformed values.

```kotlin
public inline fun <T> Comparator<T>.thenComparator(crossinline comparison: (a: T, b: T) -> Int): Comparator<T> =
    Comparator { a, b ->
        val previousCompare = this@thenComparator.compare(a, b)
        if (previousCompare != 0) previousCompare else
go inside function
```

Creates a descending comparator using the [selector] function to transform values being compared and then applying the specified [comparator] to compare transformed values.
Comparator<in T>): Comparator<T> =
    Comparator { a, b ->
        val previousCompare =
        this@then.compare(a, b)
        if (previousCompare != 0) previousCompare else comparator.compare(a, b)
    }

/**
 * Combines this comparator and the given [comparator] such that the latter is applied only
 * when the former considered values equal.
 */
public fun <T> Comparator<T>.thenDescending(comparator: Comparator<in T>): Comparator<T> =
    Comparator<T> { a, b ->
        val previousCompare = this@thenDescending.compare(a, b)
        if (previousCompare != 0) previousCompare else comparator.compare(a, b)
    }

/**
 * Extends the given [comparator] of non-nullable values to a comparator of nullable values
 * considering `null` value less than any other value.
 */
public fun <T> Comparator<T>.nullsFirst(comparator: Comparator<in T>): Comparator<T> =
    Comparator { a, b ->
        when {
            a === b -> 0
            a == null -> -1
            b == null -> 1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Extends the given [comparator] of non-nullable values to a comparator of nullable values
 * considering `null` value greater than any other value.
 */
public fun <T> Comparator<T>.nullsLast(comparator: Comparator<in T>): Comparator<T> =
    Comparator { a, b ->
        when {
            a === b -> 0
            a == null -> 1
            b == null -> -1
            else -> comparator.compare(a, b)
        }
    }

/**
 * Provides a comparator of nullable [Comparable] values considering `null` value less than any
 * other value.
 */
public fun <T : Comparable<T>> nullsFirst(): Comparator<T?> =
    nullsFirst(naturalOrder())

/**
 * Provides a comparator of nullable [Comparable] values considering `null` value greater than any
 * other value.
 */
public fun <T : Comparable<T>> nullsLast(): Comparator<T?> =
    nullsLast(naturalOrder())

/**
 * Returns a comparator that compares [Comparable] objects in natural order.
 */
public fun <T : Comparable<T>> naturalOrder(): Comparator<T> =
    @Suppress("UNCHECKED_CAST") (NaturalOrderComparator as Comparator<T>)

/**
 * Returns a comparator that compares [Comparable] objects in reversed natural order.
 */
public fun <T : Comparable<T>> reverseOrder(): Comparator<T> =
    ReverseOrderComparator as Comparator<T>

private class ReversedComparator<T>(public val comparator: Comparator<T>) : Comparator<T> {
    override fun compare(a: T, b: T): Int = comparator.compare(b, a)
    @Suppress("VIRTUAL_MEMBER_HIDDEN")
    fun reversed(): Comparator<T> =
        ReversedComparator(comparator)
}

private object NaturalOrderComparator : Comparator<Comparable<Any>> {
    override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)
    @Suppress("VIRTUAL_MEMBER_HIDDEN")
    fun reversed(): Comparator<Comparable<Any>> =
        NaturalOrderComparator
}

private object ReverseOrderComparator : Comparator<Comparable<Any>> {
    override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)
    @Suppress("VIRTUAL_MEMBER_HIDDEN")
    fun reversed(): Comparator<Comparable<Any>> =
        ReverseOrderComparator
}

@kotlin.internal.InlineOnly
public inline fun <T : Any>
    @kotlin.jvm.JvmName("nullsFirstWithComparator") nullsFirstWithComparator(comparator: Comparator<in T>): Comparator<T?> =
    nullsFirst(comparator)

@kotlin.internal.InlineOnly
public inline fun <T : Any>
    @kotlin.jvm.JvmName("nullsLastWithComparator") nullsLastWithComparator(comparator: Comparator<in T>): Comparator<T?> =
    nullsLast(comparator)

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("StandardKt")
package kotlin

import kotlin.contracts.*
public class NotImplementedError(message: String = "An operation is not implemented.") :
Error(message)

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 */
public inline fun TODO(): Nothing? = throw NotImplementedError()

/**
 * Always throws [NotImplementedError] stating that operation is not implemented.
 */
@kotlin.internal.InlineOnly
public inline fun TODO(reason: String): Nothing? = throw NotImplementedError("An operation is not implemented: $reason")

/**
 * Calls the specified function [block] and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <R> run(block: () -> R): R

/**
 * Calls the specified function [block] with 'this' value as its receiver and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R> T.run(block: T.() -> R): R

/**
 * Calls the specified function [block] with the given [receiver] as its receiver and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <T> with(receiver: T, block: T.() -> Unit): T

/**
 * Calls the specified function [block] with `this` value as its argument and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun <T, R> T.let(block: (T) -> R): R

/**
 * Returns `this` value if it satisfies the given [predicate] or `null`, if it doesn't.
 */
@kotlin.internal.InlineOnly
public inline fun <T> T.takeIf(predicate: (T) -> Boolean): T? { return if (predicate(this)) this else null

/**
 * Returns 'this' value if it does not satisfy the given [predicate] or `null`, if it does.
 */
@kotlin.internal.InlineOnly
public inline fun <T> T.takeUnless(predicate: (T) -> Boolean): T? { return if (!predicate(this)) this else null

/**
 * Calls the specified function [block] with 'this' value as its argument and returns its result.
 */
@kotlin.internal.InlineOnly
public inline fun repeat(times: Int, action: (Int) -> Unit) {

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
public actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {
    return if (a >= b) a

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.comparisons

//
// NOTE: THIS FILE IS AUTO-GENERATED by the
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.js.*
else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Byte, b: Byte): Byte { return Math.max(a.toInt(), b.toInt()).unsafeCast<Byte>()
} * Returns the greater of two values.

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Short, b: Short): Short { return Math.max(a.toInt(), b.toInt()).unsafeCast<Short>()
} * Returns the greater of two values.

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Int, b: Int): Int { return Math.max(a, b)
} * Returns the greater of two values.

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Long, b: Long): Long { return if (a >= b) a else b
} * Returns the greater of two values.

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Float, b: Float): Float { return Math.max(a, b)
} * Returns the greater of two values.

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Double, b: Double): Double { return Math.max(a, b)
} * Returns the greater of two values.

/**
 * Returns the greater of three values.
 * 
 * If there are multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Byte, b: Byte, c: Byte): Byte { return Math.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()
} * Returns the greater of three values.

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Short, b: Short, c: Short): Short { return Math.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()
} * Returns the greater of three values.

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Int, b: Int, c: Int): Int { return Math.max(a, b, c)
} * Returns the greater of three values.

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual inline fun maxOf(a: Long, b: Long, c: Long): Long { return Math.max(a, b, c)
} * Returns the greater of three values.

/**
 * Returns the greater of the given values.
 * 
 * If there are multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual fun maxOf(a: Byte, vararg other: Byte): Byte { return maxOf(maxOf(a, b), other)...
} * Returns the greater of the given values.

/**
 * Returns the greater of the given values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual fun maxOf(a: Short, vararg other: Short): Short { return maxOf(maxOf(a, b), other)...
} * Returns the greater of the given values.

/**
 * Returns the greater of the given values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual fun maxOf(a: Int, vararg other: Int): Int { return maxOf(maxOf(a, b), other)...
} * Returns the greater of the given values.

/**
 * Returns the greater of the given values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual fun maxOf(a: Long, vararg other: Long): Long { return maxOf(maxOf(a, b), other)...
} * Returns the greater of the given values.

/**
 * Returns the greater of the given values.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
@Suppress("DEPRECATION_ERROR")
public actual fun maxOf(a: Float, vararg other: Float): Float:
Float {\n  var max = a\n  for (e in other) max = maxOf(max, e)\n  return max\n}\n
\n/**
 * Returns the greater of the given values.\n * If any value is `NaN`, returns `NaN`.\n */
\npublic actual fun maxOf(a: Double, vararg other: Double): Double {\n  var max = a\n  for (e in other) max = maxOf(max, e)\n  return max\n}\n
\n/**
 * Returns the smaller of two values.\n * If values are equal, returns the first one.\n */
\npublic actual fun <T : Comparable<T>> minOf(a: T, b: T): T {\n  return if (a <= b) a else b\n}\n
\n/**
 * Returns the smaller of two values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Byte, b: Byte): Byte {\n  return Math.min(a.toInt(), b.toInt()).unsafeCast<Byte>()\n}\n
\n/**
 * Returns the smaller of two values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Short, b: Short): Short {\n  return Math.min(a, b)\n}\n
\n/**
 * Returns the smaller of two values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Int, b: Int): Int {\n  return Math.min(a, b)\n}\n
\n/**
 * Returns the smaller of two values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Long, b: Long): Long {\n  return if (a <= b) a else b\n}\n
\n/**
 * Returns the smaller of two values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Float, b: Float): Float {\n  return Math.min(a, b)\n}\n
\n/**
 * Returns the smaller of two values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Double, b: Double): Double {\n  return Math.min(a, b)\n}\n
\n/**
 * Returns the smaller of three values.\n * If there are multiple equal minimal values, returns the first of them.\n */
\npublic actual fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T {\n  return minOf(a, minOf(b, c))\n}\n
\n/**
 * Returns the smaller of three values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Byte, b: Byte, c: Byte): Byte {\n  return Math.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()\n}\n
\n/**
 * Returns the smaller of three values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Short, b: Short, c: Short): Short {\n  return Math.min(a, b, c)\n}\n
\n/**
 * Returns the smaller of three values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Int, b: Int, c: Int): Int {\n  return Math.min(a, b, c)\n}\n
\n/**
 * Returns the smaller of three values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Long, b: Long, c: Long): Long {\n  return minOf(a, minOf(b, c))\n}\n
\n/**
 * Returns the smaller of three values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Float, b: Float, c: Float): Float {\n  return Math.min(a, b, c)\n}\n
\n/**
 * Returns the smaller of three values.\n */
\n@SinceKotlin("1.4")
\npublic actual inline fun minOf(a: Double, b: Double, c: Double): Double {\n  return Math.min(a, b, c)\n}\n
\n/**
 * Returns the smaller of the given values.\n * If there are multiple equal minimal values, returns the first of them.\n */
\npublic actual fun <T : Comparable<T>> minOf(a: T, vararg other: T): T {\n  var min = a\n  for (e in other) min = minOf(min, e)\n  return min\n}\n
\n/**
 * Returns the smaller of the given values.\n */
\n@SinceKotlin("1.4")
\npublic actual fun minOf(a: Byte, vararg other: Byte): Byte {\n  var min = a\n  for (e in other) min = minOf(min, e)\n  return min\n}\n
\n/**
 * Returns the smaller of the given values.\n */
\n@SinceKotlin("1.4")
\npublic actual fun minOf(a: Short, vararg other: Short): Short {\n  var min = a\n  for (e in other) min = minOf(min, e)\n  return min\n}\n
Open Source Used In DNAC Platform Antelope  13844
/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * 
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// Auto-generated file. DO NOT EDIT!

package kotlin

import kotlin.experimental.*

@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class ULong @PublishedApi internal constructor(@PublishedApi internal val data: Long) : Comparable<ULong> {

    companion object {
        /**
         * A constant holding the minimum value an instance of ULong can have.
         */
        public const val MIN_VALUE: ULong = ULong(0)
        /**
         * A constant holding the maximum value an instance of ULong can have.
         */
        public const val MAX_VALUE: ULong = ULong(-1)
        /**
         * The number of bytes used to represent an instance of ULong in a binary form.
         */
        public const val SIZE_BYTES: Int = 8
        /**
         * The number of bits used to represent an instance of ULong in a binary form.
         */
        public const val SIZE_BITS: Int = 64
    }

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UByte): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UShort): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: UInt): Int = this.compareTo(other.toULong())

    /**
     * Compares this value with the specified value for order.
     * Returns zero if this value is equal to the specified other value, a negative number if
     * it's less than other, or a positive number if it's greater than other.
     */
    @kotlin.internal.InlineOnly
    public inline operator fun compareTo(other: ULong): Int = ulongCompare(this.data, other.data)

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UByte): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UShort): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: UInt): ULong = this.plus(other.toULong())

    /** Adds the other value to this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun plus(other: ULong): ULong = ULong(this.data.plus(other.data))

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UByte): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UShort): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: UInt): ULong = this.minus(other.toULong())

    /** Subtracts the other value from this value. */
    @kotlin.internal.InlineOnly
    public inline operator fun minus(other: ULong): ULong = ULong(this.data.minus(other.data))

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UByte): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UShort): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: UInt): ULong = this.times(other.toULong())

    /** Multiplies this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UByte): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UShort): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: UInt): ULong = this.div(other.toULong())

    /** Divides this value by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun div(other: ULong): ULong = ULong(this.data.div(other.data))

    /** Modulo of this value divided by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UByte): ULong = this.rem(other.toULong())

    /** Modulo of this value divided by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UShort): ULong = this.rem(other.toULong())

    /** Modulo of this value divided by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())

    /** Modulo of this value divided by the other value. */
    @kotlin.internal.InlineOnly
    public inline operator fun rem(other: ULong): ULong = ULong(this.data.rem(other.data))

    /**
     * Returns the smaller of the given values.
     */
    @SinceKotlin("1.4")
    public actual fun minOf(a: Int, vararg other: Int): Int {
        var min = a
        for (e in other) min = minOf(min, e)
        return min
    }

    /**
     * Returns the smaller of the given values.
     */
    @SinceKotlin("1.4")
    public actual fun minOf(a: Long, vararg other: Long): Long {
        var min = a
        for (e in other) min = minOf(min, e)
        return min
    }

    /**
     * Returns the smaller of the given values.
     */
    @SinceKotlin("1.4")
    public actual fun minOf(a: Float, vararg other: Float): Float {
        var min = a
        for (e in other) min = minOf(min, e)
        return min
    }

    /**
     * Returns the smaller of the given values.
     */
    @SinceKotlin("1.4")
    public actual fun minOf(a: Double, vararg other: Double): Double {
        var min = a
        for (e in other) min = minOf(min, e)
        return min
    }
public inline operator fun times(other: UShort): ULong = this.times(other.toULong())
/** Multiplies this value by the other value. */

public inline operator fun times(other: UInt): ULong = this.times(other.toULong())
/** Multiplies this value by the other value. */

public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))
/** Divides this value by the other value. */

public inline operator fun div(other: UByte): ULong = this.div(other.toULong())
/** Divides this value by the other value. */

public inline operator fun div(other: UShort): ULong = this.div(other.toULong())
/** Divides this value by the other value. */

public inline operator fun div(other: UInt): ULong = this.div(other.toULong())
/** Divides this value by the other value. */

public inline operator fun div(other: ULong): ULong = ulongDivide(this, other)
/** Divides this value by the other value. */

public inline operator fun rem(other: UByte): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

public inline operator fun rem(other: UShort): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

public inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())
/** Calculates the remainder of dividing this value by the other value. */

public inline operator fun rem(other: ULong): ULong = ulongRemainder(this, other)
/** Calculates the remainder of dividing this value by the other value. */

public inline operator fun inc(): ULong = ULong(data.inc())
/** Increments this value. */

public inline operator fun dec(): ULong = ULong(data.dec())
/** Decrements this value. */

public inline operator fun rangeTo(other: ULong): ULongRange = ULongRange(this, other)
/** Creates a range from this value to the specified [other] value. */

public inline infix fun shl(bitCount: Int): ULong = ULong(data shl bitCount)
/** Shifts this value left by the [bitCount] number of bits. */

public inline infix fun shr(bitCount: Int): ULong = ULong(data ushr bitCount)
/** Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros. */

public inline infix fun and(other: ULong): ULong = ULong(this.data and other.data)
/** Performs a bitwise AND operation between the two values. */

public inline infix fun or(other: ULong): ULong = ULong(this.data or other.data)
/** Performs a bitwise OR operation between the two values. */

public inline infix fun xor(other: ULong): ULong = ULong(this.data xor other.data)
/** Performs a bitwise XOR operation between the two values. */

public inline fun inv(): ULong = ULong(data.inv())
/** Inverts the bits in this value. */

public inline fun toByte(): Byte = data.toByte()
/** Converts this [ULong] value to [Byte]. */

public inline fun toShort(): Short = data.toShort()
/** Converts this [ULong] value to [Short]. */

public inline fun toInt(): Int = data.toInt()
/** Converts this [ULong] value to [Int]. */

public inline fun toLong(): Long = data
/** Converts this [ULong] value to [Long]. */

public inline fun toULong(): ULong = data
/** Converts this [ULong] value to [ULong]. */
value to [UByte].
*\n* If this value is less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents
* the same numerical value as this `ULong`.\n*\n* The resulting `UByte` value is represented by the least significant 8 bits of this `ULong` value.\n*\n* @kotlin.internal.InlineOnly
public inline fun toUByte(): UByte = data.toUByte()\n/**\n* Converts this [ULong] value to [UShort].\n*\n* If this value is less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents
* the same numerical value as this `ULong`.\n*\n* The resulting `UShort` value is represented by the least significant 16 bits of this `ULong` value.\n*\n* @kotlin.internal.InlineOnly
public inline fun toUShort(): UShort = data.toUShort()\n/**\n* Returns this value.\n*\n* @kotlin.internal.InlineOnly
public inline fun toUInt(): UInt = data.toUInt()

/**\n* Converts this [ULong] value to [Float].\n*\n* The resulting value is the closest `Float` to this `ULong` value.\n*\n* In case when this `ULong` value is exactly between two `Float`'s, the one with zero at least significant bit of mantissa is selected.\n*\n* @kotlin.internal.InlineOnly
public inline fun toFloat(): Float = this.toDouble().toFloat()
/**\n* Converts this [ULong] value to [Double].\n*\n* @kotlin.internal.InlineOnly
public inline fun toDouble(): Double = ulongToDouble(data)

/**
* Converts this [Byte] value to [ULong].\n*\n* If this value is positive, the resulting `ULong` value represents the same numerical value as this `Byte`.\n*\n* The least significant 8 bits of the resulting `ULong` value are the same as the bits of this `Byte` value.\n*\n* @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline fun Byte.toULong(): ULong = ULong(this.toLong())
/**\n* Converts this [Short] value to [ULong].\n*\n* If this value is positive, the resulting `ULong` value represents the same numerical value as this `Short`.\n*\n* The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `Short` value.\n*\n* @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline fun Short.toULong(): ULong = ULong(this.toLong())
/**\n* Converts this [Int] value to [ULong].\n*\n* If this value is positive, the resulting `ULong` value represents the same numerical value as this `Int`.\n*\n* The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `Int` value.\n*\n* @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline fun Int.toULong(): ULong = ULong(this.toLong())
/**\n* Converts this [Long] value to [ULong].\n*\n* If this value is positive, the resulting `ULong` value represents the same numerical value as this `Long`.\n*\n* The resulting `ULong` value has the same binary representation as this `Long` value.
*\n* @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline fun Long.toULong(): ULong = ULong(this)
/**\n* Converts this [Float] value to [ULong].\n*\n* The fractional part, if any, is rounded down towards zero.\n*\n* @SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic inline fun Float.toULong(): ULong = doubleToULong(this)
/**\n* Converts this [Double] value to [ULong].\n*\n* The fractional part, if any, is rounded down towards zero.\n*\n* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("\"CollectionsKt\"\")

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stddlib
//

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns 1st *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 1.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component1(): T {
    return get(0)
}

/**
 * Returns 2nd *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 2.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component2(): T {
    return get(1)
}

/**
 * Returns 3rd *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 3.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component3(): T {
    return get(2)
}

/**
 * Returns 4th *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 4.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component4(): T {
    return get(3)
}

/**
 * Returns 5th *element* from the list.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 5.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> List<T>.component5(): T {
    return get(4)
}

/**
 * Returns `true` if [element] is found in the collection.
 * 
 * Throws an [IndexOutOfBoundsException] if the size of this list is less than 0.
 */
public operator fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.contains(element: T): Boolean {
    if (this is Collection)
        return contains(element)
    return indexOf(element) >= 0
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this collection.
 * 
 * @sample samples.collections.Collections.Elements.elementAt
 */
public fun <T> Iterable<T>.elementAt(index: Int): T {
    if (this is List)
        return get(index)
    return atOrElse(index) { throw IndexOutOfBoundsException("Collection doesn't contain element at index $index.") }
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this list.
 * 
 * @sample samples.collections.Collections.Elements.elementAtOrElse
 */
@kotlin.internal.InlineOnly
public inline fun <T> List<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this list.
 * 
 * @sample samples.collections.Collections.Elements.elementAtOrNull
 */
public fun <T> Iterable<T>.elementAtOrNull(index: Int): T? {
    if (this is List)
        return getOrNull(index)
    if (index < 0)
        return null
    val iterator = iterator()
    var count = 0
    while (iterator.hasNext()) {
        val element = iterator.next()
        if (index == count++)
            return element
    }
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample samples.collections.Collections.Elements.find
 */
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.find(predicate: (T) -> Boolean): T? {
    return firstOrNull(predicate)
}

Open Source Used In DNAC Platform Antelope 13848
empty.

```kotlin
@<T> fun Iterable<T>.last(): T {
    @<T> fun <T> List<T>.last(): T {
        when (this) {
            is List -> return this[lastIndex]
            else -> {
                val iterator = this.iterator()
                while (iterator.hasNext())
                    iterator.next()
                throw NoSuchElementException("List is empty.")
            }
        }
    }

    @<T> fun Iterable<T>.lastOrNull(): T? {
        when (this) {
            is List -> return if (isEmpty()) null else this[size - 1]
            else -> {
                val iterator = iterator()
                if (!iterator.hasNext())
                    return null
                var last = iterator.next()
                while (iterator.hasNext())
                    last = iterator.next()
                return last
            }
        }
    }

    @<T> fun List<T>.lastOrNull(): T? {
        return if (isEmpty()) null else this[size - 1]
    }

    @<T> fun <T> Iterable<T>.lastOrNull(predicate: (T) -> Boolean): T? {
        var last: T? = null
        for (element in this) {
            if (predicate(element)) {
                last = element
            }
        }
        return last
    }

    @<T> fun List<T>.lastOrNull(predicate: (T) -> Boolean): T? {
        val iterator = this.listIterator(size)
        while (iterator.hasPrevious()) {
            val element = iterator.previous()
            if (predicate(element)) return element
        }
        return null
    }

    @<T> fun List<T>.single(): T {
        when (this) {
            is List -> return this.single()
            else -> {
                val iterator = iterator()
                if (iterator.hasNext())
                    throw NoSuchElementException("Collection is empty.")
                var last = iterator.next()
                while (iterator.hasNext())
                    last = iterator.next()
                throw NoSuchElementException("List contains no element matching the given predicate.")
            }
        }
    }

    @<T> fun List<T>.random(): T {
        if (isEmpty())
            throw NoSuchElementException("Collection is empty.")
        return elementAt(random.nextInt(size))
    }

    @<T> fun List<T>.randomOrNull(): T? {
        if (isEmpty())
            return null
        return elementAt(random.nextInt(size))
    }

    @<T> fun Collection<T>.random(): T {
        return random(Random)
    }

    @<T> fun Collection<T>.randomOrNull(random: Random): T? {
        if (isEmpty())
            return null
        return elementAt(random.nextInt(size))
    }

    @<T> fun Collection<T>.randomOrNull(): T? {
        return randomOrNull(Random)
    }
```

Open Source Used In DNAC Platform Antelope 13850
element itself and returns the result of predicate evaluation on the element. and returns the result of predicate evaluation on the element.

```kotlin
public inline fun <T> Iterable<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {
    return filterIndexedTo(ArrayList<T>(), predicate)
}

/**
 * Appends all elements matching the given [predicate] to
 * the given [destination].
 * 
 * @param [predicate] function that takes the index of an element and the element itself and returns the result of predicate evaluation on the element.
 */
public inline fun <T, C : MutableCollection<in T>> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}
```

```kotlin
/**
 * Returns a list containing all elements that are instances of specified type parameter R and returns the result of predicate evaluation on the element.
 */
public inline fun <reified R> Iterable<*>.filterIsInstanceTo(destination: Collection<R>): List<R> {
    return filterIsInstanceTo(ArrayList<R>(), predicate)
}

/**
 * Appends all elements that are instances of specified type parameter R and returns the result of predicate evaluation on the element.
 */
public inline fun <reified R, C : MutableCollection<in R>> Iterable<*>.filterIsInstanceTo(destination: C): C {
    for (element in this) if (element is R) destination.add(element)
    return destination
}
```

```kotlin
/**
 * Returns a list containing all elements that are not `null` and returns the result of predicate evaluation on the element.
 */
public fun <T : Any> Iterable<T?>.filterNotNull(): List<T> {
    return filterNotNullTo(ArrayList<T>())
}

/**
 * Appends all elements that are not `null` to the
 * the given [destination].
 */
public fun <C : MutableCollection<in T>> Iterable<T>.filterNotNullTo(destination: C): C {
    for (element in this) if (element != null) destination.add(element)
    return destination
}
```

```kotlin
/**
 * Returns a list containing elements at indices in the specified [indices] range.
 */
public fun <T> List<T>.slice(indices: IntRange): List<T> {
    if (indices.isEmpty()) return listOf()
    return this.subList(indices.start, indices.endInclusive + 1).toList()
}
```

```kotlin
/**
 * Returns a list containing elements at specified [indices] and returns the result of predicate evaluation on the element.
 */
public fun <T> List<T>.slice(indices: Iterable<Int>): List<T> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<T>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}
```
listIterator(size - n))
            list.add(item)
    }
    return list
}

/**
 * Returns a list containing last elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.takeLastWhile
 */
public inline fun <T> List<T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {
    if (isEmpty()) return emptyList()
    val iterator = listIterator(size)
    while (iterator.hasNext()) {
        if (!predicate(iterator.next())) {
            iterator.next()
            val expectedSize = size - iterator.nextIndex()
            if (expectedSize == 0) return emptyList()
            return ArrayList<T>(expectedSize).apply {
                while (iterator.hasNext())
                    add(iterator.next())
            }
        }
    }
    return toList()<@@*/

/**
 * Returns a list containing first elements satisfying the given [predicate].
 * @sample samples.collections.Collections.Transformations.takeWhile
 */
public inline fun <T> Iterable<T>.takeWhile(predicate: (T) -> Boolean): List<T> {
    var list = ArrayList<T>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

/**
 * Reverses elements in the list in-place.
 */
public expect fun <T> MutableList<T>.reverse(): Unit

/**
 * Returns a list with elements in reversed order.
 */
public fun <T> Iterable<T>.reversed(): List<T> {
    if (this is Collection && size <= 1) return toList()<@@*/
    val list = toMutableList()<@@*/
    list.reverse()<@@*/
    return list
}

/**
 * Randomly shuffles elements in this list in-place using the specified [random] instance as the source of randomness.
 */
@SinceKotlin("1.3")
public fun <T> MutableList<T>.shuffle(random: Random): Unit {
    for (i in lastIndex downTo 1) {
        val j = random.nextInt(i + 1)
        this[j] = this.set(i, this[j])
    }
}

/**
 * Sorts elements in the list in-place according to natural sort order of the value returned by specified [selector] function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public inline fun <T, R : Comparable<R>> MutableList<T>.sortBy(crossinline selector: (T) -> R?): Unit {
    if (size > 1)
        sortWith(compareBy(selector))
}

/**
 * Sorts elements in the list in-place descending according to natural sort order of the value returned by specified [selector] function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public inline fun <T, R : Comparable<R>> MutableList<T>.sortByDescending(crossinline selector: (T) -> R?): Unit {
    if (size > 1)
        sortWith(compareByDescending(selector))
}

/**
 * Returns a list of all elements sorted according to natural sort order.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T : Comparable<T>> MutableList<T>.sortDescending(): Unit {
    sortWith(reverseOrder())
}

/**
 * Returns a list of all elements sorted according to the specified [comparator].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T> Iterable<T>.sortedWith(comparator: Comparator<in T>): List<T> {
    if (this is Collection && size <= 1) return toList()<@@*/
    @Suppress("UNCHECKED_CAST")
    return (toTypedArray<Comparable<T>>() as Array<T>).apply { sort() }.asList()<@@*/
}

/**
 * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T, R : Comparable<R>> Iterable<T>.sortedBy(selector: (T) -> R?): List<T> {
    return sortedWith(compareBy(selector))
}

/**
 * Returns a list of all elements sorted descending according to the specified [comparator].
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun <T> Iterable<T>.sortedDescending(comparator: Comparator<in T>): List<T> {
    return sortedWith(reverseOrder())
}
(toTypedArray<Any?>() as Array<T>).apply { sortWith(comparator) }.asList()
    }
    return
toMutableList().apply { sortWith(comparator) }
}

/**
 * Returns an array of Boolean containing all of the elements of this collection.
 */
public fun Collection<Boolean>.toBooleanArray(): BooleanArray {
    val result = BooleanArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Byte containing all of the elements of this collection.
 */
public fun Collection<Byte>.toByteArray(): ByteArray {
    val result = ByteArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Char containing all of the elements of this collection.
 */
public fun Collection<Char>.toCharArray(): CharArray {
    val result = CharArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Double containing all of the elements of this collection.
 */
public fun Collection<Double>.toDoubleArray(): DoubleArray {
    val result = DoubleArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Float containing all of the elements of this collection.
 */
public fun Collection<Float>.toFloatArray(): FloatArray {
    val result = FloatArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Int containing all of the elements of this collection.
 */
public fun Collection<Int>.toIntArray(): IntArray {
    val result = IntArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Long containing all of the elements of this collection.
 */
public fun Collection<Long>.toLongArray(): LongArray {
    val result = LongArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns an array of Short containing all of the elements of this collection.
 */
public fun Collection<Short>.toShortArray(): ShortArray {
    val result = ShortArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns a [Map] containing key-value pairs provided by [transform] function
 * applied to elements of the given collection.
 * If any of two pairs would have the same key the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 */
public inline fun <T, K, V> Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateTo(LinkedHashMap<K, V>(capacity), transform)
}

/**
 * Returns a [Map] containing the elements from the given collection indexed
 * by the key returned from [keySelector] function applied to each element.
 * The returned map preserves the entry iteration order of the original collection.
 */
public inline fun <T, K> Iterable<T>.associateBy(keySelector: (T) -> K): Map<K, T> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)
}

/**
 * Returns a [Map] containing the values provided by [valueTransform] and
 * indexed by [keySelector] functions applied to elements of the given collection.
 * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 */
public inline fun <T, K, V> Iterable<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {
    val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)
}

/**
 * Populates and returns the [destination] mutable map with
 * key-value pairs provided by [keySelector] function applied to each element of the given collection.
 * If any two pairs would have the same key returned by [keySelector] the last one gets added to the map.
 * The returned map preserves the entry iteration order of the original collection.
 */
public inline fun <T, K, M : MutableMap<in K, in T>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this)
        destination.put(keySelector(element), element)
    return destination
}
mutable map with key-value pairs.

If any two elements would have the same key returned by [keySelector] the last one gets added to the map.

If any two elements are equal, the last one gets added to the map. The returned map preserves the entry iteration order of the original collection.

Populates and returns the [destination] mutable map with key-value pairs provided by [transform] function applied to each element of the given collection.

Appends all elements to the given [destination] collection.

Returns a [List] containing all elements.

Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original collection.

Yielded from results of [transform] function being invoked on each element of original collection.
By Lambda ReturnType

public inline fun <T, R>
Iterable<T>.flatMap(transform: (T) -> Sequence<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being
 * invoked on each element in * and its index in the original collection.
 * @sample samples.collections.Collections.Transformations.flatMapIndexed
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
public inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (T, Int) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element in * and its index in the original collection, to the given [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform: (T, Int) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(checkIndexOverflow(index++), element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups elements of the original collection by the key returned by the given [keySelector]
 * function on each element and returns a map where each group key is associated with a list
 * of corresponding elements. The returned map preserves the entry iteration order of the keys produced from the
 * original collection.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <T, K> Iterable<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {
    return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original collection
 * by the key returned by the given [keySelector] function on each element and returns a map where each group key is associated with a list
 * of corresponding values. The returned map preserves the entry iteration order of the keys produced from the
 * original collection.
 * @sample samples.collections.Collections.Transformations.groupBy
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun <T, K, C : MutableCollection<in T>> Iterable<T>.groupByTo(destination: C, keySelector: (T) -> K, valueTransform: (T) -> C): C {
    var index = 0
    for (element in this) {
        val key = keySelector(element)
        val list = valueTransform(element)
        destination.addIfAbsent(key, list)
    }
    return destination
}
original collection.

```
*\n* @sample samples.collections.Collections.Transformations.groupByKeysAndValues

*\npublic inline fun <T, K, V> Iterable<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {
    \n    \n    \n    val key = keySelector(element)
    val list = destination.getOrPut(key) { ArrayList<V>() }
    list.add(element)
    return destination
}

*\n* Groups elements of the original collection by the key returned by the given [keySelector] function\n* applied to each element and puts to the [destination] map each group key associated with\n* a list of corresponding elements.\n*\n* @sample samples.collections.Collections.Transformations.groupByKeysAndValues
```

```
samples.collections.Collections.Transformations.mapNotNull

public inline fun <T, R : Any>
Iterable<T>.mapNotNull(transform: (T) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(), transform)
}

/**
 * Applies the given [transform] function to each element of the original collection
 * and appends only the non-null results to the given [destination].
 */
public */*public inline fun <T, R, C : MutableCollection<C>>
Iterable<T>.mapTo(destination: C, transform: (T) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original collection
 * into an [IndexedValue] containing the index of the element and the element itself.
 */
public fun <T> Iterable<T>.withIndex(): Iterable<IndexedValue<T>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a list containing only distinct elements from the given collection.
 */
public fun <T> Iterable<T>.distinct(): List<T> {
    return this.toMutableSet().toList()
}

/**
 * Returns a list containing only elements from the given collection
 * having distinct keys returned by the given [selector] function.
 */
public inline fun <T, K>
Iterable<T>.distinctBy(selector: (T) -> K): List<T> {
    val set = this.toMutableSet() .set.removeAll(other)
    return set
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
public inline fun <T>
Iterable<T>.all(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return true
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if collection has at least one element.
 */
public fun <T>
Iterable<T>.any(): Boolean {
    if (this is Collection) return !isEmpty()
    return iterator().hasNext()
}

/**
 * Returns `true` if at least one element matches the given [predicate].
 */
public inline fun <T>
Iterable<T>.any(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return false
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns the number of elements in
 * Returns the number of elements in
 */
this collection.

```kotlin
public fun <T> Iterable<T>.count(): Int { if (this is Collection) return size
  var count = 0
  for (element in this) checkCountOverflow(++count)
  return count
}

/**
 * Returns the number of elements matching the given [predicate].
 */
Iterable<T>.count(predicate: (T) -> Boolean): Int { if (this is Collection && !isEmpty()) return 0
  var count = 0
  for (element in this) if (predicate(element)) checkCountOverflow(++count)
  return count
}

/**
 * Returns the number of elements in this collection.
 */
@kotlin.internal.InlineOnly
public inline fun <T> Collection<T>.count(): Int { return size

/**
 * Returns the number of elements in the original collection.
 */
@kotlin.internal.InlineOnly
public inline fun <T> List<T>.count(): Int { return size

/**
 * Returns the number of elements matching the given [predicate].
 */
@kotlin.internal.InlineOnly
public inline fun <T> List<T>.count(predicate: (T) -> Boolean): Int { return size

/**
 * Performs the given [action] on each element.
 */
@kotlin.internal.HidesMembers
public inline fun <T> Iterable<T>.forEach(action: (T) -> Unit): Unit { for (element in this) action(element)

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes the index of an element and the element itself and performs the action on the element.
 */
public inline fun <T> Iterable<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit { var index = 0
  for (item in this) action(checkIndexOverflow(index++), item)

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun <T : Comparable<T>> Iterable<T>.max(): T? { return maxOrNull()

@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public inline fun <T, R : Comparable<R>> Iterable<T>.maxBy(selector: (T) -> R): T? { return maxByOrNull(selector)
```

---

Open Source Used In DNAC Platform Antelope 13859
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Iterable<T>.maxByOrNull(selector: (T) -> R): T? {
    val iterator = iterator() as Iterable<T>.maxOf(selector) {
        var maxElem = iterator.next() as T
        var v = selector(maxElem) as R
        do {
            v = selector(iterator.next()) as R
            if (v > maxElem) {
                maxElem = iterator.next() as T
                v = selector(maxElem) as R
            } else if (v < maxElem) {
                return maxElem
            } else {
                maxElem = maxElement
            }
        } while (iterator.hasNext())
        return maxElem
    } as T? * Returns the largest value among all values produced by [selector] function * applied to each element in the collection. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * @throws NoSuchElementException if the collection is empty.
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.maxOf(selector: (T) -> Double): Double {
    val iterator = iterator() as Iterable<T>.maxOf(selector) {
        var maxValue = iterator.next() as Double
        do {
            val v = iterator.next() as Double
            if (v > maxValue) {
                maxValue = v
            }
        } while (iterator.hasNext())
        return maxValue
    }
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.maxOf(selector: (T) -> Float): Float {
    val iterator = iterator() as Iterable<T>.maxOf(selector) {
        var maxValue = iterator.next() as Float
        do {
            val v = iterator.next() as Float
            if (v > maxValue) {
                maxValue = v
            }
        } while (iterator.hasNext())
        return maxValue
    }
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <T, R : Comparable<R>> Iterable<T>.maxOfOrNull(selector: (T) -> R): R? {
    val iterator = iterator() as Iterable<T>.maxOfOrNull(selector) {
        var maxElem = iterator.next() as T
        var v = selector(maxElem) as R
        do {
            v = selector(iterator.next()) as R
            if (v > maxElem) {
                maxElem = iterator.next() as T
                v = selector(maxElem) as R
            } else if (v < maxElem) {
                return maxElem
            } else {
                maxElem = maxElement
            }
        } while (iterator.hasNext())
        return maxElem
    } as T? * Returns the largest value among all values produced by [selector] function * applied to each element in the collection. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * @throws NoSuchElementException if the collection is empty.
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.maxOfOrNull(selector: (T) -> Double): Double? {
    val iterator = iterator() as Iterable<T>.maxOfOrNull(selector) {
        var maxValue = iterator.next() as Double
        do {
            val v = iterator.next() as Double
            if (v > maxValue) {
                maxValue = v
            }
        } while (iterator.hasNext())
        return maxValue
    } as T? * Returns the largest value among all values produced by [selector] function * applied to each element in the collection. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * @throws NoSuchElementException if the collection is empty.
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.maxOfOrNull(selector: (T) -> Float): Float? {
    val iterator = iterator() as Iterable<T>.maxOfOrNull(selector) {
        var maxValue = iterator.next() as Float
        do {
            val v = iterator.next() as Float
            if (v > maxValue) {
                maxValue = v
            }
        } while (iterator.hasNext())
        return maxValue
    } as T? * Returns the largest value among all values produced by [selector] function * applied to each element in the collection. * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`. * @throws NoSuchElementException if the collection is empty.

Iterable<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R

/**
 * Returns the largest value according to the provided [comparator] among all values
 * produced by [selector] function applied to each element in the collection.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public inline fun <T, R>
    Iterable<T>.maxOfWithNull(comparator: Comparator<in R>, selector: (T) -> R): R? {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var maxValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest element or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public fun <T : Comparable<T>> Iterable<T>.maxOrNull(): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var max = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (max < e) max = e
    }
    return max
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun <T> Iterable<T>.maxWith(comparator: Comparator<in T>): T? {
    return maxWithOrNull(comparator)
}

/**
 * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.internal.InlineOnly
public fun <T, R : Comparable<R>> Iterable<T>.minByOrNull(selector: (T) -> R): T? {
    return minByOrNull(selector)
}
R): T? {\n  var iterator = iterator()\n  if (!iterator.hasNext()) return null \n  var minElem = iterator.next()\n  var minVal = selector(minElem) \n  do { \n    var e = iterator.next()\n    var v = selector(e) \n    if (minValue > v) { \n      minValue = e\n      minElem = e\n    } \n  } while (iterator.hasNext())\n  return minElem\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the collection.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the collection is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) -> Double): Double { \n  var iterator = iterator() \n  if (!iterator.hasNext()) throw NoSuchElementException() \n  var minValue = selector(iterator.next()) \n  while (iterator.hasNext()) { \n    var e = selector(iterator.next()) \n    minValue = minOf(minValue, e) \n  } \n  return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the collection.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the collection is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) -> Float): Float { \n  var iterator = iterator() \n  if (!iterator.hasNext()) throw NoSuchElementException() \n  var minValue = selector(iterator.next()) \n  while (iterator.hasNext()) { \n    var e = selector(iterator.next()) \n    minValue = minOf(minValue, e) \n  } \n  return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the collection.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the collection is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minOf(selector: (T) -> R): R { \n  var iterator = iterator() \n  if (!iterator.hasNext()) throw NoSuchElementException() \n  var minValue = selector(iterator.next()) \n  while (iterator.hasNext()) { \n    var e = selector(iterator.next()) \n    if (minValue > e) { \n      minValue = e\n    } \n  } \n  return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.\n * @throws NoSuchElementException if the collection is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T) -> Double): Double? { \n  var iterator = iterator() \n  if (!iterator.hasNext()) return null \n  var minValue = selector(iterator.next()) \n  while (iterator.hasNext()) { \n    var e = selector(iterator.next()) \n    minValue = minOf(minValue, e) \n  } \n  return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * @throws NoSuchElementException if the collection is empty.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T) -> Float): Float? { \n  var iterator = iterator() \n  if (!iterator.hasNext()) return null \n  var minValue = selector(iterator.next()) \n  while (iterator.hasNext()) { \n    var e = selector(iterator.next()) \n    minValue = minOf(minValue, e) \n  } \n  return minValue\n}\n
/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.\n */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable>R> Iterable<T>.minOfOrNull(selector: (T) -> R): R? { \n  var iterator = iterator() \n  if (!iterator.hasNext()) return null \n  var minValue = selector(iterator.next()) \n  while (iterator.hasNext()) { \n    var e = selector(iterator.next()) \n    if (minValue > e) { \n      minValue = e\n    } \n  } \n  return minValue\n}
func min{v:

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the collection.
 * @return Returns the smallest value.
 */

public inline fun <T, R> Iterable<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Performs the given [action] on each element and returns the collection itself afterwards.
 * @param [predicate] function that takes the index of an element and the element itself as parameters and returns true if the element matches the given predicate.
 * @return Returns true if no elements match the given [predicate].
 */

public inline fun <T> Iterable<T>.noneWithPredicateOrNull(predicate: (T) -> Boolean): Boolean {
    if (this is Collection && isEmpty()) return true
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes an element as parameter and performs the given [action] on it.
 * @param [index] accepting the index of the element.
 * @return Performs the given [action] on each element.
 */

public inline fun <T, C : Iterable<T>> C.onEachIndexed(action: (index: Int, T) -> Unit): C {
    return apply { forEachIndexed(action) }
}

/**
 * Performs the given [action] on each element and returns the collection itself afterwards.
 * @param [element] accepting the element.
 * @param [predicate] function that takes an element as parameter and returns true if the element matches the given [predicate].
 * @param [action] function that takes an element as parameter and performs the given [action] on it.
 * @return Performs the given [action] on each element.
 */

public inline fun <T, C : Iterable<T>> C.onEach(action: (T) -> Unit): C {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [operation] on each element, starting with the first element and accumulating value on each element.
 * @param [operation] function that takes an element as parameter and performs the given [operation] on it.
 * @param [currentIndex] accepting the index of the element.
 * @return Accumulates value starting with the first element and applies [operation] from left to right to current accumulator value and each element.
 * @throws Exception if this collection is empty.
 */

content_forⒼ
collection can be empty in an expected way.

* please use [reduceOrNull] instead. It returns `null` when its receiver is empty.

* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.reduce

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element with its index in the original collection.

* @param [operation] function that takes the index of an element, current accumulator value and the element itself,
 * and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.reduce

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> Iterable<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return null
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(accumulator, iterator.next())
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.

* @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.reduceRight

*/
@SinceKotlin("1.4")
public inline fun <S, T : S> List<T>.reduceRight(operation: (T, acc: S) -> S): S {
    val iterator = listIterator(size)
    if (!iterator.hasPrevious()) throw UnsupportedOperationException("Empty list can't be reduced.")
    var accumulator: S = iterator.previous()
    while (iterator.hasPrevious()) {
        accumulator = operation(iterator.previous(), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original list and current accumulator value.

* @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.

* @sample samples.collections.Collections.Aggregates.reduceRightIndexed

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
    val iterator = listIterator(size)
    if (!iterator.hasPrevious()) throw UnsupportedOperationException("Empty list can't be reduced.")
    var accumulator = operation(accumulator, iterator.next())
    while (iterator.hasPrevious()) {
        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.previous())
    }
    return accumulator
}
Accumulates value starting with the last element and applying [operation] from right to left. 
* to each element with its index in the original list and current accumulator value.

* and calculates the next accumulator value.

@samples.collections.Collections.Aggregates.reduceRightOrNull

@sinceKotlin("1.4")
public inline fun <S, T : S> List<T>.reduceRightIndexedOrNull(operation: (Int, T, acc: S) -> S): S? {
    val iterator = listIterator(size)
    if (!iterator.hasNext())
        return null
    var accumulator: S = iterator.previous()
    while (iterator.hasNext()) {
        val index = iterator.previousIndex()
        accumulator = operation(index, iterator.previous(), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left.
 * 
 * Returns `null` if the list is empty.
 * 
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value.
 * 
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull
 */
@sinceKotlin("1.4")
@wasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> List<T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {
    val iterator = listIterator(size)
    if (!iterator.hasNext())
        return null
    var accumulator: S = iterator.previous()
    while (iterator.hasNext()) {
        accumulator = operation(iterator.previous(), accumulator)
    }
    return accumulator
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right. 
 * 
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the
 * previous value in resulting list.
 * 
 * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.
 * 
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@sinceKotlin("1.4")
public inline fun <T, R> Iterable<T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {
    val estimatedSize = collectionSizeOrDefault(9)
    if (estimatedSize == 0) return listOf(initial)
    val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the
 * previous value in resulting list.
 * 
 * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.
 * 
 * @sample samples.collections.Collections.Aggregates.runningFold
 */
@sinceKotlin("1.4")
@wasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> Iterable<T>.runningReduce(operation: (acc: S, T) -> S): List<S> {
    val iterator = this.iterator()
    if (!iterator.hasNext())
        throw UnsupportedOperationException("Empty list can't be reduced.")
    var accumulator = operation(iterator.previous(), accumulator)
    return accumulator
var accumulator: S = iterator.next()
result.add(accumulator)
while (iterator.hasNext())
    accumulator = operation(accumulator, iterator.next())
    result.add(accumulator)
return result

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from
 * left to right to each element, its index in the original collection and current accumulator value that starts with
 * the first element of this collection.
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the
 * index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @return a list containing successive accumulation values generated by applying [operation] from
 * left to right to each element, its index in the original collection and current accumulator value that starts with
 * the first element of this collection.
 */
@SinceKotlin("1.4")
public inline fun <S, T : S> Iterable<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return emptyList()
    var accumulator: S = iterator.next()
    val result = ArrayList<S>(collectionSizeOrDefault(10)).apply {
        add(accumulator)
    }
    var index = 1
    while (iterator.hasNext()) {
        accumulator = operation(index++,
            accumulator, iterator.next())
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from
 * left to right to each element, its index in the original collection and current accumulator value that starts with
 * the first element of this collection.
 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes the
 * index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
 * @return a list containing successive accumulation values generated by applying [operation] from
 * left to right to each element, its index in the original collection and current accumulator value that starts with
 * the first element of this collection.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <T, R> Iterable<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

@Deprecated("Use runningReduce instead.",
    ReplaceWith("runningReduce(operation)"), level =
    DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public inline fun <S, T : S> Iterable<T>.sumBy(selector: (T) -> Int): Int {
    var sum: Int = 0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function
 * applied to each element in the collection.
 * @param [selector] function that takes
 * an element in the collection and returns its sum.
 * @return the sum of all values produced by [selector] function
 * applied to each element in the collection.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("sumOfDouble")
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.sumOf(selector: (T) -> Double): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/*@
@OpenSourceUsedInDNACPlatformAntelope 13866
*/
[selector] function applied to each element in the collection.

```kotlin
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOf")
@kotlin.internal.InlineOnly
public inline fun <T>
Iterable<T>.sumOf(selector: (T) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the collection.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOf")
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.sumOf(selector: (T) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the collection.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOf")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <T> Iterable<T>.sumOf(selector: (T) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
/**
 * Returns the sum of all values produced by [selector]
 * function applied to each element in the collection.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.jvm.JvmName("sumOf")
@ExperimentalUnsignedTypes
@kotlin.inter
al.InlineOnly
public inline fun <T> Iterable<T>.sumOf(selector: (T) -> ULong): ULong {
    var sum: ULong = 0.toULong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
```

```kotlin
/**
 * Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are
 * any `null` elements.
 */
public fun <T : Any> Iterable<T?>.requireNoNulls(): Iterable<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as Iterable<T>
}
```

```kotlin
/**
 * Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null`
 * elements.
 */
public fun <T : Any> List<T?>.requireNoNulls(): List<T> {
    for (element in this) {
        if (element == null) {
            throw IllegalArgumentException("null element found in $this.")
        }
    }
    @Suppress("UNCHECKED_CAST")
    return this as List<T>
}
```

```kotlin
/**
 * Splits this collection into a list of lists each not exceeding the given 
 * [size].
 * 
 * The last list in the resulting list may have less elements than the 
 * given [size].
 * 
 * @sample samples.collections.Collections.Transformations.chunked
 */
@SinceKotlin("1.2")
public fun <T>
Iterable<T>.chunked(size: Int): List<List<T>> {
    return windowed(size, size, partialWindows = true)
}
```

```kotlin
/**
 * Splits this collection into several lists each not exceeding the given 
 * [size] and applies the given [transform] function to an each.
 * 
 * The last list may have less elements than the 
 * given [size].
 * 
 * @param size the number of elements to take in each list, must be positive and can be greater than 
 * the number of elements in this collection.
 * @param * @sample 
 * samples.collections.Collections.Transformations.chunked
 */
@SinceKotlin("1.2")
public fun <T, R> Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}
```

```kotlin
/**
 * Returns a list containing all elements of the original collection without the first occurrence of the given 
 * [element].
 */
public 
operator fun <T> Iterable<T>.minus(element: T): List<T> {
    val result = 
    ArrayList<T>(collectionSizeOrDefault(10))
    var removed = false
    return this.filterTo(result) { if (!removed
    && it == element) { removed = true; false } else true }
}
```

```kotlin
/**
 * Returns a list containing all elements of the original collection without the first occurrence of the given 
 * [element].
 */
public operator fun <T> Iterable<T>.minus(element: T): List<T> {
    val result = 
    ArrayList<T>(collectionSizeOrDefault(10))
    var removed = false
    return this.filterTo(result) { if (!removed
    && it == element) { removed = true; false } else true }
}
```

```kotlin
/**
 * Returns a list containing all elements of the original collection without the first occurrence of the given 
 * [element].
 */
public operator fun <T> Iterable<T>.minus(element: T): List<T> {
    val result = 
    ArrayList<T>(collectionSizeOrDefault(10))
    var removed = false
    return this.filterTo(result) { if (!removed
    && it == element) { removed = true; false } else true }
}
```

```kotlin
/**
 * Returns a list containing all elements of the original collection without the first occurrence of the given 
 * [element].
 */
public operator fun <T> Iterable<T>.minus(element: T): List<T> {
    val result = 
    ArrayList<T>(collectionSizeOrDefault(10))
    var removed = false
    return this.filterTo(result) { if (!removed
    && it == element) { removed = true; false } else true }
}
```
Iterable<T>.minus(elements: Array<out T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Iterable<T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Element): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: List<T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Sequence<T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Array<out T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Collection<T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Sequence<T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Array<out T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Collection<T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.minus(elements: Sequence<T>): List<T> { return this.filterNot { it in elements.toList() } }

Iterable<T>.plus(element: T): List<T> { return this.plus(element) }

Iterable<T>.plus(elements: Array<out T>): List<T> { return this.plus(elements) }

Iterable<T>.plus(elements: Collection<T>): List<T> { return this.plus(elements) }

Iterable<T>.plus(elements: Sequence<T>): List<T> { return this.plus(elements) }

Open Source Used In DNAC Platform Antelope 13868
Iterable<T>.plusElement(element: T): List<T> {
    return plus(element)
}

/**
 * Returns a list containing all elements of the original collection and then the given [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Collection<T>.plusElement(element: T): List<T> {
    return plus(element)
}

/**
 * Returns a list of snapshots of the window of the given [size] sliding along this collection with the given [step], where each snapshot is a list.
 */
    checkWindowSizeStep(size, step)
    if (this is RandomAccess && this is List) {
        val thisSize = this.size
        val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1
        val result = ArrayList<List<T>>(resultCapacity)
        var index = 0
        while (index in 0 until thisSize) {
            val windowSize = size.coerceAtMost(thisSize - index)
            if (windowSize < size && !partialWindows) break
            result.add(List(windowSize) { this[it + index] })
            index += step
        }
        return result
    }
    val result = ArrayList<List<T>>()
    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = false).forEach {
        result.add(it)
    }
    return result
}

/**
 * Returns a list of results of applying the given [transform] function to an each list representing a view over the window of the given [size] sliding along this collection with the given [step].
 */
    checkWindowSizeStep(size, step)
    if (this is RandomAccess && this is List) {
        val thisSize = this.size
        val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1
        val result = ArrayList<R>(resultCapacity)
        val window = MovingSubList(this)
        var index = 0
        while (index in 0 until thisSize) {
            val windowSize = size.coerceAtMost(thisSize - index)
            if (!partialWindows && windowSize < size) break
            window.move(index, index + windowSize)
            result.add(transform(window))
            index += step
        }
        return result
    }
    val result = ArrayList<R>()
    windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = true).forEach {
        result.add(transform(it))
    }
    return result
}

/**
 * Returns a list of pairs built from the elements of `this` collection and the [other] array with the same index.
 * The returned list has length of the shortest collection.
 */
public infix fun <T, R> Iterable<T>.zip(other: Array<out R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of values built from the elements of `this` collection and the [other] array with the same index using the provided [transform] function applied to each pair of elements.
 */
public inline fun <T, R, V> Iterable<T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {
    val arraySize = other.size
    val list = ArrayList<V>(minOf(collectionSizeOrDefault(10), arraySize))
    var i = 0
    forEach { element in this } {
        if (i >= arraySize) break
        list.add(transform(element, other[i++]))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] collection with the same index.
 */
public infix fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> {
    return zip(other) { t1, t2 ->
t1 to t2

Returns a list of values built from the elements of `this` collection and the [other] collection with the same index * * using the provided [transform] function applied to each pair of elements.

The returned list has length of the shortest collection.

@sample samples.collections.Iterables.Operations.zipIterableWithTransform

Iterable<T>.zip(other: Iterable<R>, transform: (a: T; b: R) -> V): List<V> |

val first = iterator().val second = other.iterator().val list = ArrayList<V>(minOf(collectionSizeOrDefault(10), other.collectionSizeOrDefault(10))).while (first.hasNext() && second.hasNext()) |

list.add(transform(first.next(), second.next())) return list

* * *ln * The returned list is empty if this collection contains less than two elements.

@sample samples.collections.Collections.Transformations.zipWithNextUn

*An@SinceKotlin("1.2")unpublic inline fun <T, R> Iterable<T>.zipWithNext(transform: (a: T, b: R) -> V): List<Pair<T, R>> |

{ a, b -> a to b } return zipWithNext { a, b -> a to b } 

* * *ln * * * un * Returns a list containing the results of applying the given [transform] function to an each pair of two adjacent elements in this collection.

* * *ln * * * The returned list is empty if this collection contains less than two elements.

@sample samples.collections.Collections.Transformations.joinToFindDeltas

*An@SinceKotlin("1.2")unpublic inline fun <T, R> Iterable<T>.zipWithNext(transform: (a: T, b: R) -> V): List<R> |

val iterator = iterator()

if (!iterator.hasNext()) return emptyList()

val result = mutableListOf<R>()

var current = iterator.next()

while (iterator.hasNext()) {

    val next = iterator.next()
    result.add(transform(current, next))

    current = next
}

return result

* * *ln * * * Append the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.

* * *ln * * * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to \"...\"").

* * *ln * * * @sample samples.collections.Collections.Transformations.joinToString

* * *ln * * * @public

Iterable<T>.joinToString(separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "...", transform: ((T) -> CharSequence)? = null): String |

return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()

* * *ln * * *

Iterable<T>.asIterable(): Iterable<T> |

return this

* * *ln * * *

Iterable<T>.asSequence(): Sequence<T> |

return Sequence { this.iterator() }

* * *ln * * *

Iterable<T>.average(): Double |

val sum: Double = 0.0

var count = 0

for (element in this) {

    sum += element

    checkCountOverflow(++count)
}

return if (count == 0) Double.NaN else sum / count

* * *ln * * *

Iterable<T>.average(): Double |

val sum: Double = 0.0

var count = 0

for (element in this) {

    sum += element

    checkCountOverflow(++count)
}

return if (count == 0) Double.NaN else sum / count

* * *ln * * *

Iterable<T>.average(): Double |

val sum: Double = 0.0

var count = 0

for (element in this) {

    sum += element

    checkCountOverflow(++count)
}

return if (count == 0) Double.NaN else sum / count

* * *ln * * *

Iterable<T>.average(): Double |

val sum: Double = 0.0

var count = 0

for (element in this) {

    sum += element

    checkCountOverflow(++count)
}

return if (count == 0) Double.NaN else sum / count

* * *ln * * *

Iterable<T>.average(): Double |

val sum: Double = 0.0

var count = 0

for (element in this) {

    sum += element

    checkCountOverflow(++count)
}

return if (count == 0) Double.NaN else sum / count

* * *ln * * *

Iterable<T>.average(): Double |

val sum: Double = 0.0

var count = 0

for (element in this) {

    sum += element

    checkCountOverflow(++count)
}

return if (count == 0) Double.NaN else sum / count

* * *ln * * *
var count: Int = 0
for (element in this) {
    sum += element
    checkCountOverflow(++count)
}

return if (count == 0) Double.NaN else sum / count

/**
 * Returns an average value of elements in the collection.
 */
@kotlin.jvm.JvmName("averageOfLong")
public fun Iterable<Long>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns
 an average value of elements in the collection.
 */
@kotlin.jvm.JvmName("averageOfFloat")
public fun Iterable<Float>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the collection.
 */
@kotlin.jvm.JvmName("averageOfDouble")
public fun Iterable<Double>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfByte")
public fun Iterable<Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfShort")
public fun Iterable<Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfInt")
public fun Iterable<Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfLong")
public fun Iterable<Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfFloat")
public fun Iterable<Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the collection.
 */
@kotlin.jvm.JvmName("sumOfDouble")
public fun Iterable<Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the
 * Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

import kotlin.comparisons.naturalOrder
import kotlin.random.Random

/** Returns the array if it's not `null`, or an
empty array otherwise. */
@kotlin.internal.InlineOnly
public actual inline fun <T> Array<out T>?.orEmpty(): Array<out T> = this ?: emptyArray<T>()

@kotlin.internal.InlineOnly
public actual inline fun <T> Collection<T>.toTypedArray(): Array<T> =
copyToArray(this)

@JsName("copyToArray")
@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@kotlin.internal.InlineOnly
internal actual inline fun <E> buildListInternal(builderAction: MutableList<E>.() -> Unit): List<E> {
    return ArrayList<E>().apply(builderAction).build()
}

@PublishedApi
@SinceKotlin("1.3")
@ExperimentalStdlib

fun <K, V> Map<out K, V>.toSingletonMap(): Map<K, V> =
this.toMutableMap().toImmutableMap()

@Suppress("NOTHING_TO_INLINE")
internal actual inline fun <T> Array<out T>.copyToAnyArray(isVarargs: Boolean): Array<Any?> =
if (isVarargs)
    // no need to copy vararg array in JS
    this
else
    this.copyOf()

@PublishedApi
internal actual fun checkIndexOverflow(index: Int): Int {
    if (index < 0) {
        throwIndexOverflow()
    }
    return index
}

@PublishedApi
internal actual fun checkCountOverflow(count: Int): Int {
    if (count < 0) {
        throwCountOverflow()
    }
    return count
}

/**
 * JS map and set implementations do not make use of capacities or load factors.
 */
@PublishedApi
internal actual fun mapCapacity(expectedSize: Int) =
    expectedSize

/**
 * Checks a collection builder function capacity argument.
 */
@PublishedApi
internal fun checkBuilderCapacity(capacity: Int) {
    require(capacity >= 0) { "capacity must be non-negative." }
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CollectionsKt")

package kotlin.collections

/**
 * Returns the given iterator itself. This allows to use an instance of iterator in a `for`
 * loop.
 */
@kotlin.internal.InlineOnly
public inline operator fun <T> Iterator<T>.iterator(): Iterator<T> = this

/**
 * Returns an [Iterator] that wraps each element produced
 * by the original iterator into an [IndexedValue],
 * counting index from zero.
 */
public fun <T> Iterator<T>.withIndex():
    Iterator<IndexedValue<T>> = IndexingIterator(this)

/**
 * Performs the given [operation] on each element of
 * this [Iterator].
 */
public inline fun <T>
    Iterator<T>.forEach(operation: (T) -> Unit): Unit {
    for (element in this) operation(element)
}

/**
 * Performs the given [operation] on each element of
 * this [Iterator].
 */
internal class IndexingIterator<out T>(private var iterator: Iterator<T>) : Iterator<IndexedValue<T>> {
    private var index =
        0
    final override fun hasNext(): Boolean = iterator.hasNext()
    final override fun next(): IndexedValue<T> =
        IndexedValue(checkIndexOverflow(index++), iterator.next())
}

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("ComparisonsKt")

package kotlin.comparisons

import kotlin.random.*

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Byte, b: Byte): Byte

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Short, b: Short): Short

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Int, b: Int): Int

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Long, b: Long): Long

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Float, b: Float): Float

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
public expect fun <T : Comparable<T>> maxOf(a: T, b: T, c: T):
    T

/*
 * Copyright 2010-2020 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 * Use of this source code is
 * governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

Open Source Used In DNAC Platform Antelope 13873
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Int, b: Int, c: Int): Int

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Long, b: Long, c: Long): Long

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Float, b: Float, c: Float): Float

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun maxOf(a: Double, b: Double, c: Double): Double

/**
 * Returns the greater of three values according to the order specified by the given [comparator].
 */
@SinceKotlin("1.4")
public expect fun <T> maxOf(a: T, vararg other: T, comparator: Comparator<in T>): T

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Byte, b: Byte): Byte

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Short, b: Short): Short

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Int, b: Int): Int

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Long, b: Long): Long

/**
 * Returns the greater of two values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Float, b: Float): Float

/**
 * Returns the greater of two values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Double, b: Double): Double

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Byte, b: Byte): Byte

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Short, b: Short): Short

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Int, b: Int): Int

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Long, b: Long): Long

/**
 * Returns the smaller of two values.
 * 
 * If either value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Float, b: Float): Float

/**
 * Returns the smaller of two values.
 * 
 * If either value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Double, b: Double): Double

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Byte, b: Byte, c: Byte): Byte

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Short, b: Short, c: Short): Short

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Int, b: Int, c: Int): Int

/**
 * Returns the smaller of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect inline fun minOf(a: Long, b: Long, c: Long): Long

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect function maxOf(a: Int, b: Int, c: Int): Int

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect function maxOf(a: Long, b: Long, c: Long): Long

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect function maxOf(a: Float, b: Float, c: Float): Float

/**
 * Returns the greater of three values.
 * 
 * If any value is `NaN`, returns `NaN`.
 */
@SinceKotlin("1.1")
@kotlin.internal.InlineOnly
public expect function maxOf(a: Double, b: Double, c: Double): Double

/**
 * Returns the greater of three values according to the order specified by the given [comparator].
 * 
 * If there are multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.4")
public expect fun <T> maxOf(a: T, b: T, c: T, comparator: Comparator<in T>): T

/**
 * Returns the greater of three values.
 * 
 * If there are multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.4")
public expect fun <T> maxOf(a: T, b: T, comparator: Comparator<in T>): T

/**
 * Returns the greater of three values.
 * 
 * If there are multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.4")
public expect fun <T> maxOf(a: T, vararg other: T): T

/**
 * Returns the greater of three values.
 * 
 * If there are multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.4")
public expect fun <T> maxOf(a: T, vararg other: T, comparator: Comparator<in T>): T

/**
 * Returns the greater of three values.
 * 
 * If there are multiple equal maximal values, returns the first of them.
 */
@SinceKotlin("1.4")
public expect fun <T> maxOf(a: T, vararg other: T, comparator: Comparator<in T>): T
any value is `NaN`, returns `NaN`.

fun minOf(a: Float, b: Float, c: Float): Float

fun minOf(a: Double, b: Double, c: Double): Double

fun minOf(a: Float, b: Float, c: Float): Float

fun minOf(a: Double, b: Double, c: Double): Double

fun <T> minOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {
    return minOf(a, minOf(b, c), comparator)
}

fun <T> minOf(a: T, vararg other: T): T {
    val min = a
    for (e in other) if (comparator.compare(min, e) > 0) min = e
    return min
}

fun <K, V> Map<out K, V>.toList(): List<Pair<K, V>>


fun <K, V> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> Sequence<R>): List<R>

fun <K, V> Map<out K, V>.flatMapTo(destination: MutableList<R>, transform: (Map.Entry<K, V>) -> Iterable<R>)

fun <K, V> Map<out K, V>.flatMapTo(destination: MutableList<R>, transform: (Map.Entry<K, V>) -> Iterable<R>)

fun <K, V> Map<out K, V>.flatMapTo(destination: MutableList<R>, transform: (Map.Entry<K, V>) -> Iterable<R>)
the given [destination].

*\@SinceKotlin("1.4.4")*\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@kotlin.jvm.JvmName(\"flatMapSequenceTo\")\@in public inline fun \<K, V, R, C : MutableCollection\<in R\>> Map\<out K, V\>.flatMapTo(destination: C, transform: (Map.Entry\<K, V\>) -> Sequence\<R\>): C { \n \n for (element in this) { \n \n list = transform(element) \n \n destination.addAll(list) \n \n return destination(n) } \n }\n\n\* Returns a list containing the results of applying the given [transform] function\n* to each entry in the original map.\n\* @sample
samples.collections.Maps.Transformations.mapTo
*\@in public inline fun \<K, V, R\> Map\<out K, V\>.map(transform: (Map.Entry\<K, V\>) -> R): List\<R\> { \n \n return mapToArray(ArrayList\<R\>(size), transform) \n }\n\* Returns a list containing only the non-null results of applying the given [transform] function\n* to each entry in the original map.\n\* @sample
samples.collections.Maps.Transformations.mapNotNull
*\@in public inline fun \<K, V, R\> Map\<out K, V\>.mapNotNull(transform: (Map.Entry\<K, V\>) -> R?): List\<R\>: \n \n return mapNotNullToArray(ArrayList\<R\>(), transform) \n }\n\* Applies the given [transform] function to each entry in the original map\n* and appends only the non-null results to the given [destination].\n\* @sample
samples.collections.Maps.Transformations.mapToNot Null
*\@in public inline fun \<K, V, R, C : MutableCollection\<in R\>> Map\<out K, V\>.mapTo(destination: C, transform: (Map.Entry\<K, V\>) -> R?): C { \n \n \n for (item in this) \n \n \n \n destination.add(transform(item)) \n \n return destination(n) } \n }\n\* Returns the number of entries matching the given [predicate].\n\* @sample
samples.collections.Collections.Aggregates.all
*\@in public inline fun \<K, V\> Map\<out K, V\>.all(predicate: (Map.Entry\<K, V\>) -> Boolean): Boolean { \n \n \n if (isEmpty()) return true \n \n \n return true(n) }\n\* Returns `true` if all entries match the given [predicate].\n\* @sample
samples.collections.Collections.Aggregates.all
*\@in public inline fun \<K, V\> Map\<out K, V\>.any(predicate: (Map.Entry\<K, V\>) -> Boolean): Boolean { \n \n \n if (isEmpty()) return false \n \n \n return false(n) }\n\* Returns `true` if at least one entry matches the given [predicate].\n\* @sample
samples.collections.Collections.Aggregates.any
*\@in public inline fun \<K, V\> Map\<out K, V\>.count(): Int { \n \n return size(n) }\n\* Returns the number of entries in this map.\n\* @sample
samples.collections.Collections.Aggregates.count
*\@in public inline fun \<K, V\> Map\<out K, V\>.count(predicate: (Map.Entry\<K, V\>) -> Boolean): Int { \n \n \n \n return size(n) }\n\* Returns the number of entries matching the given [predicate].\n\* @sample
samples.collections.Collections.Aggregates.count
*\@in public inline fun \<K, V\> Map\<out K, V\>.count(predicate: (Map.Entry\<K, V\>) -> Boolean): Int { \n \n return size(n) }\n\* Returns the number of entries matching the given [predicate].\n\* @sample
samples.collections.Collections.Aggregates.count
*\@in public inline fun \<K, V\> Map\<out K, V\>.count(predicate: (Map.Entry\<K, V\>) -> Boolean): Int { \n \n \n \n return size(n) }\n\* Performs the given [action] on each entry.\n\* @sample
samples.collections.Collections.Aggregates.count
*\@in public inline fun \<K, V\> Map\<out K, V\>.forEach(action: (Map.Entry\<K, V\>) -> Unit): Unit { \n \n \n for (element in this) \n \n \n \n action(element) \n \n }\n\* Returns the largest value among all values produced by [selector] function\n* applied to each entry in the map.\n\* @sample
samples.collections.Collections.Aggregates.forEach
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxBy(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* Returns the first entry yielding the largest value of the given function or `null` if there are no entries.\n\* @sample
samples.collections.Collections.Aggregates.maxBy
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxByNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* Returns the largest value among all values produced by [selector] function\n* applied to each entry in the map.\n\* @sample
samples.collections.Collections.Aggregates.maxByNullNull
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOf(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
samples.collections.Collections.Aggregates.maxOf
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOfNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
samples.collections.Collections.Aggregates.maxOfNullNull
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOfNullNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
samples.collections.Collections.Aggregates.maxOfNullNullNull
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOfNullNullNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
samples.collections.Collections.Aggregates.maxOfNullNullNullNull
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOfNullNullNullNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullNullNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
samples.collections.Collections.Aggregates.maxOfNullNullNullNullNull
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOfNullNullNullNullNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullNullNullNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOfNullNullNullNullNullNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullNullNullNullNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOfNullNullNullNullNullNullNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullNullNullNullNullNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
*\@in public inline fun \<K, V\> Map\<out K, V\>.maxOfNullNullNullNullNullNullNullNullNull(selector: (Map.Entry\<K, V\>) -> R): Map.Entry\<K, V\>: \n \n return maxByNullNullNullNullNullNullNullNullNullToArray(ArrayList\<R\>(), selector) \n }\n\* @sample
among all values produced by [selector] function
applied to each entry in the map.
* If any of values produced by [selector] function
is ‘NaN’, the returned result is ‘NaN’.
* @throws NoSuchElementException if the map is empty.
* * * @SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <K, V> Map<out K, V>.maxOf(selector: (Map.Entry<K, V>) -> Float): Float {
    return entries.maxOf(selector)
}

/**
 * Returns the largest value among all values produced by [selector] function
applied to each entry in the map.
* @throws NoSuchElementException if the map is empty.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <K, V, R : Comparable<R>> Map<out K, V>.maxOf(selector: (Map.Entry<K, V>) -> R): R {
    return entries.maxOf(selector)
}

/**
 * Returns the largest value among all values produced by [selector] function
applied to each entry in the map or null if there are no entries.
* If any of values produced by [selector] function is ‘NaN’, the returned result is ‘NaN’.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
public inline fun <K, V> Map<out K, V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {
    return entries.maxOfOrNull(selector)
}

/**
 * Returns the largest value according to the provided [comparator]
* among all values produced by [selector] function applied to each entry in the map.
* @throws NoSuchElementException if the map is empty.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
    return entries.maxOfWith(comparator, selector)
}

/**
 * Returns the largest value according to the provided [comparator]
* among all values produced by [selector] function applied to each entry in the map or null if there are no entries.
*/
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.internal.InlineOnly
    return entries.maxOfWithOrNull(comparator, selector)
}

@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
    return

Open Source Used In DNAC Platform Antelope 13877

return minByOrNull(selector)\n
```kotlin
/*
 * Returns the first entry yielding the smallest value of the given function
 * or `null` if there are no entries.
 */
fun minByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {
    return entries.minByOrNull(selector)\n}
```

```kotlin
/**
 * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN\n *;\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
public inline fun <K, V> Map<K, V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {
    return entries.minOfOrNull(selector)\n}
```

```kotlin
/**
 * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if there are no entries.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN\n *;\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
public inline fun <K, V> Map<K, V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {
    return entries.minOfOrNull(selector)\n}
```

```kotlin
/**
 * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if there are no entries.\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN\n *;\n */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
public inline fun <K, V, R : Comparable<R>> Map<K, V>.minOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {
    return entries.minOfOrNull(selector)\n}
```
    return minWithOrNull(comparator)
}

/**
 * Returns the first entry having the smallest value according to the provided 
 * comparator or `null` if there are no entries.
 */
@SinceKotlin("1.1")
    return entries.minWithOrNull(comparator)
}

/**
 * Performs the given [action] on each entry and returns the map itself afterwards.
 * @sample samples.collections.Collections.Aggregates.minWith
 */
public fun <K, V> Map<out K, V>.minWith(action: (Map.Entry<K, V>) -> Unit): Map<K, V> {
    return apply { for (element in this) action(element) }
}

/**
 * Performs the given [action], providing sequential index with the entry, and returns 
 * the map itself afterwards.
 * @sample samples.collections.Collections.Aggregates.minWithIndexed
 */
public inline fun <K, V> Map<out K, V>.minWithIndexed(action: (index: Int, Map.Entry<K, V>) -> Unit): Map<K, V> {
    return apply {
        entries.forEachIndexed(action)
    }
}

/**
 * Creates an [Iterable] instance that wraps the original map returning 
 * its entries when being iterated.
 * @sample samples.collections.Collections.Aggregates.asIterable
 */
public inline fun <K, V> Map<out K, V>.asIterable(): Iterable<Map.Entry<K, V>> {
    return entries
}

/**
 * Creates a [Sequence] instance that 
 * wraps the original map returning its entries when being iterated.
 * @sample samples.collections.Collections.Aggregates.asSequence
 */
public fun <K, V> Map<out K, V>.asSequence(): Sequence<Map.Entry<K, V>> {
    return entries.asSequence()
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the
 * Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SequencesKt")

package kotlin.sequences

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
// https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*

/**
 * Returns 
 * `true` if [element] is found in the sequence.
 * The operation is _terminal_.
 */
public operator fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.contains(element: T): Boolean {
    return indexOf(element) >= 0
}

/**
 * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the 
 * [index] is out of bounds of this sequence.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public fun <T> Sequence<T>.elementAt(index: Int): T {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("Sequence doesn't contain element at 
index $index.") }
}

/**
 * Returns an element at the given [index] or the result of calling the [defaultValue] 
 * function if the [index] is out of bounds of this sequence.
 * @sample samples.collections.Collections.Elements.elementAtOrElse
 */
public fun <T> Sequence<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {
    if (index < 0)
        return defaultValue(index)
    val iterator = iterator()
    var count = 0
    while (iterator.hasNext()) {
        val element = iterator.next()
        if (index == count++)
            return element
    }
    return defaultValue(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this 
 * sequence.
 * @sample samples.collections.Collections.Elements.elementAtOrNull
 */
public fun <T> Sequence<T>.elementAtOrNull(index: Int): T? {
    if (index < 0)
        return null
    val iterator = iterator() var count = 0
    while (iterator.hasNext()) {
        val element = iterator.next() if (index == count++)
            return element
    }
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {
    if (index < 0)
        return defaultValue(index)
    val iterator = iterator() val var count = 0
    while (iterator.hasNext()) {
        val element = iterator.next() if (index == count++)
            return element
    }
    return null
}
samples.collections.Collections.Elements.find

@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.find(predicate: (T) -> Boolean): T? {
    return firstOrNull(predicate)
}

/**
 * Returns first element.
 * @throws NoSuchElementException if the sequence is empty.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.first(): T {
    val iterator = iterator()
    if (!iterator.hasNext())
        throw NoSuchElementException("Sequence is empty.")
    return iterator.next()
}

/**
 * Returns first element matching the given [predicate], or `null` if no such element was found.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.first(predicate: (T) -> Boolean): T? {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Sequence contains no element matching the predicate.")
}

/**
 * Returns first index of [element], or -1 if the sequence does not contain element.
 * The operation is _terminal_.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (element == item)
            return index
        index++
    }
    return -1
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the sequence does not contain such element.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.indexOfFirst(predicate: (T) -> Boolean): Int {
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (predicate(item))
            return index
        index++
    }
    return -1
}

/**
 * Returns last index of [element], or -1 if the sequence does not contain element.
 * The operation is _terminal_.
 */
public fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.lastIndexOf(element: T): Int {
    var lastIndex = -1
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (element == item)
            lastIndex = index
        index++
    }
    return lastIndex
}

/**
 * Returns last element.
 * @throws NoSuchElementException if the sequence is empty.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.last(): T {
    val iterator = iterator()
    if (!iterator.hasNext())
        throw NoSuchElementException("Sequence is empty.")
    var last = iterator.next()
    while (iterator.hasNext())
        last = iterator.next()
    return last
}

/**
 * Returns the last element matching the given [predicate].
 * @throws NoSuchElementException if no such element is found.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.last(predicate: (T) -> Boolean): T? {
    var last: T? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            last = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Sequence contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return last as T
}

/**
 * Returns last index of the last element matching the given [predicate], or -1 if the sequence does not contain such element.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.lastIndexOfLast(predicate: (T) -> Boolean): Int {
    var lastIndex = -1
    var index = 0
    for (item in this) {
        checkIndexOverflow(index)
        if (predicate(item))
            lastIndex = index
        index++
    }
    return lastIndex
}

/**
 * Returns last element, or `null` if the sequence is empty.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.lastOrNull(): T? {
    val iterator = iterator()
    if (!iterator.hasNext())
        return null
    var last = iterator.next()
    while (iterator.hasNext())
        last = iterator.next()
    return last
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.lastOrNull(predicate: (T) -> Boolean): T? {
    var last: T? = null
    for (element in this) {
        if (predicate(element)) {
            last = element
            if (!found) throw
        }
    }
    return last
}

/**
 * Returns the single element,
or throws an exception if the sequence is empty or has more than one element. The operation is _terminal_.

*public fun <T> Sequence<T>.single(): T { val iterator = iterator(); if (!iterator.hasNext()) throw NoSuchElementException("Sequence is empty."); val single = iterator.next(); if (iterator.hasNext()) throw IllegalArgumentException("Sequence contains more than one matching element."); return single; }

*public fun <T> Sequence<T>.singleOrNull(): T { val iterator = iterator(); if (!iterator.hasNext()) return null; val single = iterator.next(); if (iterator.hasNext()) return null; return single; }

*public inline fun <T> Sequence<T>.single(predicate: (T) -> Boolean): T { var single: T? = null; var found = false; for (element in this) { if (predicate(element)) { single = element; found = true; } } if (!found) throw NoSuchElementException("Sequence contains no matching element."); @Suppress("UNCHECKED_CAST") return single as T; }

*public inline fun <T> Sequence<T>.singleOrNull(predicate: (T) -> Boolean): T? { var single: T? = null; var found = false; for (element in this) { if (predicate(element)) { if (found) return null; single = element; found = true; } } if (!found) return null; return single; }

*public fun <T> Sequence<T>.drop(n: Int): Sequence<T> { require(n >= 0) { "Requested element count $n is less than zero." }; return when {
    n == 0 -> this
    this is DropTakeSequence -> this.drop(n)
    else -> DropSequence(this, n)
};

*public fun <T> Sequence<T>.dropWhile(predicate: (T) -> Boolean): Sequence<T> { return DropWhileSequence(this, predicate); }

*public fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> { return FilteringSequence(this, true, predicate); }

*public fun <T> Sequence<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): Sequence<T> { // TODO: Rewrite with generalized MapFilterIndexingSequence return TransformingSequence(FilteringSequence(IndexingSequence(this), true, { predicate(it.index, it.value) })); }

*public inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C { forEachIndexed { index, element -> if (predicate(index, element)) destination.add(element); } return destination; }

*public fun <reified R>: Sequence<@kotlin.internal.NoInfer R> { return filter { it is R } as Sequence<R>; }

*public fun <T> Sequence<T>.filterIsInstance(): Sequence<@kotlin.internal.NoInfer T> { return filter { it is T } as Sequence<T>; }
@sample samples.collections.Collections.Filtering.filterIsInstanceTo

```kotlin
public inline fun <reified R, C : MutableCollection<in R>> Sequence<*>.filterIsInstanceTo(destination: C): C {
    for (element in this) if (element is R) destination.add(element)
    return destination
}
```

```kotlin
/**
 * Returns a sequence containing all elements not matching the given [predicate].
 * The operation is _intermediate_ and _stateless_.
 *
 * @sample samples.collections.Collections.Filtering.filter
 */
public fun <T> Sequence<T>.filterNot(predicate: (T) -> Boolean): Sequence<T> {
    return FilteringSequence(this, false, predicate)
}
```

```kotlin
/**
 * Returns a sequence containing all elements that are not `null`.
 * The operation is _intermediate_ and _stateless_.
 *
 * @sample samples.collections.Collections.Filtering.filterNotNull
 */
public fun <T : Any> Sequence<T?>.filterNotNull(): Sequence<T> {
    return FilteringSequence(this, true, { it == null })
```

```kotlin
/**
 * Appends all elements that are not `null` to the given [destination].
 * The operation is _terminal_.
 *
 * @sample samples.collections.Collections.Filtering.filterNotNullTo
 */
public fun <C : MutableCollection<in T>, T : Any> Sequence<T?>.filterNotNullTo(destination: C): C {
    for (element in this) if (element != null) destination.add(element)
    return destination
}
```

```kotlin
/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 * The operation is _terminal_.
 *
 * @sample samples.collections.Collections.Filtering.filterTo
 */
public inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}
```

```kotlin
/**
 * Appends all elements matching the given [predicate] to the given [destination].
 * The operation is _terminal_.
 *
 * @sample samples.collections.Collections.Filtering.filterTo
 */
public inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}
```

```kotlin
/**
 * Returns a sequence containing first [n] elements.
 * The operation is _intermediate_ and _stateless_.
 *
 * @throws IllegalArgumentException if [n] is negative.
 *
 * @sample samples.collections.Collections.Transformations.take
 */
public fun <T> Sequence<T>.take(n: Int): Sequence<T> {
    require(n >= 0) { "Requested element count \$n is less than zero." }
    return when {
        n == 0 -> emptySequence()
        this is DropTakeSequence -> this.take(n)
        else -> TakeSequence(this, n)
    }
}
```

```kotlin
/**
 * Returns a sequence containing first elements satisfying the given [predicate].
 * The operation is _intermediate_ and _stateless_.
 *
 * @sample samples.collections.Collections.Transformations.takeWhile
 */
public fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {
    return TakeWhileSequence(this, predicate)
}
```

```kotlin
/**
 * Returns a sequence that yields elements of this sequence sorted according to their natural sort order.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * The operation is _intermediate_ and _stateful_.
 *
 * @sample samples.collections.Collections.Sorting.sorted
 */
public fun <T : Comparable<T>> Sequence<T>.sorted(): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sorted.toMutableList()
            sortedList.sort()
            return sortedList.iterator()
        }
    }
}
```

```kotlin
/**
 * Returns a sequence that yields elements of this sequence sorted descending according to natural sort order.
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 * The operation is _intermediate_ and _stateful_.
 *
 * @sample samples.collections.Collections.Sorting.sortedDescending
 */
public fun <T : Comparable<T>> Sequence<T>.sortedDescending(): Sequence<T> {
    return sortedWith(reverseOrder())
}
```
to the specified [comparator].

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> { val sortedList = this@sortedWith.toMutableList() sortedList.sortWith(comparator) return sortedList.iterator() }

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.

* * *

public fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {
    return object : Sequence<T> {
        override fun iterator(): Iterator<T> {
            val sortedList = this@sortedWith.toMutableList()
            sortedList.sortWith(comparator)
            return sortedList.iterator()
        }
    }
}

* * *

The sort is _stable_. It means that equal elements preserve their order relative
to each other after sorting.

The operation is _intermediate_ and _stateful_.
valueSelector)

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each
 * element of the given sequence. If any two elements are equal, the last one overwrites
 * the former value in the map.
 * The operation is _terminal_.*
 */
@SinceKotlin("1.3")
public inline fun <K, V, M : MutableMap<in K, in V>> Sequence<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Appends all elements to the given [destination] collection.
 * The operation is _terminal_.*
 */
@SinceKotlin("1.3")
public inline fun <T, C : MutableCollection<in T>> Sequence<T>.toCollection(destination: C) : C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Returns a new [HashSet] of all elements.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.toHashSet(): HashSet<T> {
    return toCollection(HashSet<T>())
}

/**
 * Returns a [List] containing all elements.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.toList(): List<T> {
    return this.toMutableList().optimizeReadOnlyList()
}

/**
 * Returns a new [MutableList] filled with all elements of this sequence.
 * The operation is _terminal_.*
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("flatMapIterable")
public fun <T, R>
    Sequence<T>.flatMap(transform: (T) -> Iterable<R>): Sequence<R> {
    return FlatteningSequence(this,
        transform, Iterable<R>::iterator)
}

/**
 * Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.
 * The operation is _intermediate_ and _stateless_.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.jvm.JvmName("flatMapIndexedIterable")
public fun <T, R>
    Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): Sequence<R> {
    return flatMapIndexed(this, transform, Iterable<R>::iterator)
}

/*
*
*/
Open Source Used In DNAC Platform Antelope 13884
(index: Int, T) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(checkIndexOverflow(index++), element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each element
 * and its index in the original sequence, to the given [destination]. The operation is _terminal_
 * @since Kotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
 ByLambdaReturnType
public inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform:
(index: Int, T) -> Sequence<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(checkIndexOverflow(index++), element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of [transform] function being invoked on each
 * element of original sequence, to the given [destination]. The operation is _terminal_
 * @since Kotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
 ByLambdaReturnType
public inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Groups elements of the original sequence by the key returned by the given [keySelector]
 * function and returns a map where each group key is associated with a list of corresponding elements.
 * The returned map preserves the entry iteration order of the keys produced from the original sequence.
 * The operation is _terminal_
 * @sample samples.collections.Collections.Transformations.groupBy
 */
public inline fun <T, K, M : MutableMap<in K, MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<T>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each element of the original sequence
 * by the key returned by the given [keySelector] function applied to the element and puts to the [destination]
 * map each group key associated with a list of corresponding values.
 * The operation is _terminal_
 * @sample samples.collections.Collections.Transformations.groupByKeysAndValues
 */
public inline fun <T, K, V, M : MutableMap<in K, MutableList<V>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}
operations
* using the specified [keySelector] function to extract a key from each element.
* The operation is
* _intermediate_ and _stateless_.
* @sample samples.collections.Grouping
* groupingByEachCount
* @SinceKotlin("1.1.")
* public inline fun <T, K> Sequence<T>.groupingBy(crossinline keySelector: (T) -> K):
* Grouping<T, K> {
*   return object : Grouping<T, K> {
*     override fun sourceIterator(): Iterator<T> =
*       this@groupingBy.iterator()
*     override fun keyOf(element: T): K = keySelector(element)
*   }
* }
* Returns a sequence containing the results of applying the given [transform] function
* to each element in the original sequence.
* The operation is _intermediate_ and _stateless_.
* @sample
* samples.collections.Collections.Transformations.map
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* @param [transform] function that takes the index of an element and the element itself
* and returns the result of the transform applied to the element.
* The operation is _intermediate_ and _stateless_.
* @param [transform] function that takes the index of an element and the element itself
* and returns the result of the transform applied to the element.
* The operation is _intermediate_ and _stateless_.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing the results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
* The operation is _intermediate_ and _stateless_.
* Returns a sequence containing only the non-null results of applying the given [transform] function
* to each element and its index in the original sequence.
samples.collections.Collections.Transformations.distinctAndDistinctBy

public fun <T>
    Sequence<T>.distinct(): Sequence<T> {
        return this.distinctBy { it }
    }

/**
 * Returns a sequence containing only elements from the given sequence
 * having distinct keys returned by the given [selector].
 * Among elements of the given sequence with equal keys, only the first one will be present in the
 * resulting sequence.
 * The elements in the resulting sequence are in the same order as they were in the source
 * sequence.
 * The operation is _intermediate_ and _stateful_.
 */
public fun <T, K>
    Sequence<T>.distinctBy(selector: (T) -> K): Sequence<T> {
        return DistinctSequence(this, selector)
    }

/**
 * Returns a new [MutableSet] containing all distinct elements from the given sequence
 * The returned set preserves the element iteration order of the original sequence.
 * The operation is _terminal_ and _stateful_.
 */
public fun <T> Sequence<T>.toMutableSet(): MutableSet<T> {
    val set = LinkedHashSet<T>()
    for (item in this)
        set.add(item)
    return set
}

/**
 * Returns `true` if all elements match the given [predicate].
 * The operation is _terminal_.
 * @sample samples.collections.Collections.Aggregates.all
 */
public inline fun <T> Sequence<T>.all(predicate: (T) -> Boolean): Boolean {
    for (element in this)
        if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if sequence has at least one element.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.any(): Boolean {
    return iterator().hasNext()
}

/**
 * Returns `true` if at least one element
 * matches the given [predicate].
 * The operation is _terminal_.
 * @sample samples.collections.Collections.Aggregates.anyWithPredicate
 */
public inline fun <T> Sequence<T>.any(predicate: (T) -> Boolean): Boolean {
    for (element in this)
        if (predicate(element)) return true
    return false
}

/**
 * Returns the number of elements in this sequence.
 * The operation is _terminal_.
 */
public fun <T> Sequence<T>.count(): Int {
    var count = 0
    for (element in this)
        checkCountOverflow(++count)
    return count
}

/**
 * Returns the number of elements matching the given [predicate].
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.count(predicate: (T) -> Boolean): Int {
    var count = 0
    for (element in this)
        if (predicate(element))
            checkCountOverflow(++count)
    return count
}

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to
 * right to current accumulator value and each element.
 * The operation is _terminal_.
 */
public inline fun <T, R>
    Sequence<T>.fold(initial: R, operation: (acc: R, T) -> R): R {
        var accumulator = initial
        for (element in this)
            accumulator = operation(accumulator, element)
        return accumulator
    }

/**
 * Accumulates value starting with [initial] value and applying [operation] from left to
 * right to current accumulator value and each element with its index in the original sequence.
 * The operation is _terminal_.
 */
public inline fun <T, R>
    Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {
        var index = 0
        var accumulator = initial
        for (item in this)
            accumulator = operation(checkIndexOverflow(index++), accumulator, item)
        return accumulator
    }

/**
 * Performs the given [action] on each element.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.forEach(action: (T) -> Unit): Unit {
    for (element in this)
        action(element)
}

/**
 * Performs the given [action] on each element, providing sequential index with the
 * element.
 * The operation is _terminal_.
 */
public inline fun <T> Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {
    var index = 0
    for (item in this)
        action(checkIndexOverflow(index++), item)
}

@Deprecated("Use maxOrNull instead.",
    ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun Sequence<Double>.max(): Double? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.",
    ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun Sequence<Float>.max(): Float? {
    return maxOrNull()
}
selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        maxValue = maxOf(maxValue, v)!
    }
    return maxValue!
}

/**
 * Returns the largest value among all values produced
 * by [selector] function
 * applied to each element in the sequence or `null` if there are no elements.
 * * The operation is _terminal_.
 * * @throws NoSuchElementException if the sequence is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.experimental.InlineOnly
public inline fun <T, R : Comparable<R>>
Sequence<T>.maxOfOrNull(selector: (T) -> R): R?
    { val val = selector(iterator.next())
        if (!iterator.hasNext()) return null!
        max = selector(iterator.next())!
        while (iterator.hasNext()) {
            val v = selector(iterator.next())!
                if (max < v) max = v!
        }
        return max!
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the sequence or `null` if there are no elements.
 * * The operation is _terminal_.
 * * @throws NoSuchElementException if the sequence is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.experimental.InlineOnly
public inline fun <T, R>
Sequence<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R
    { val val = selector(iterator.next())!
        if (!iterator.hasNext()) throw NoSuchElementException()
        max = selector(iterator.next())!
        while (iterator.hasNext()) {
            val v = selector(iterator.next())!
                if (comparator.compare(max, v) < 0) max = v!
        }
        return max!
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the sequence or `null` if there are no elements.
 * * The operation is _terminal_.
 * * @throws NoSuchElementException if the sequence is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.experimental.InlineOnly
public inline fun <T : Comparable<T>>
Sequence<T>.maxOrNull(): T
    { val val = iterator.next()!
        if (!iterator.hasNext()) return null!
        max = iterator.next()!
        while (iterator.hasNext()) {
            val e = iterator.next()!
                if (max < e) max = e!
        }
        return max!
}

@Deprecated("Use maxWithOrNull instead.",
ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun <T>
Sequence<T>.maxWith(comparator: Comparator<in T>): T
    { val val = iterator.next()!
        if (!iterator.hasNext()) return null!
        max = iterator.next()!
        while (iterator.hasNext()) {
            val e = iterator.next()!
                if (max < e) max = e!
        }
        return max!
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.1")
public fun Sequence<Double>.min(): Double? {
    return minOrNull()
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.1")
public fun Sequence<Float>.min(): Float? {
    return minOrNull()
}

@Deprecated("Use minOrNull instead.",
ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.1")
public fun <T : Comparable<T>> Sequence<T>.minOrNull(): T? {
    { val val = iterator.next()!
        if (!iterator.hasNext()) return null!
        min = iterator.next()!
        while (iterator.hasNext()) {
            val e = iterator.next()!
                if (max < e) max = e!
        }
        return max!
}
ReplaceWith("minOrNull()")\n@DeprecatedSinceKotlin(warningSince = "1.4")\n@SinceKotlin("1.1")\npublic fun Sequence<Float>.min(): Float? \n    return minOrNull()\n\n@DeprecatedSinceKotlin(warningSince = "1.4")\npublic fun <T : Comparable<T>> Sequence<T>.min(): T? \n    return minOrNull()\n\n@SinceKotlin("1.4")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.minBy(selector: (T) -> R): T? \n    {\n        val iterator = iterator()\n        if (!iterator.hasNext()) return null\n        var minElem = iterator.next()\n        var minValue = selector(minElem)\n        do {\n            val e = iterator.next()\n            val v = selector(e)\n            if (minValue > v) {\n                minElem = e\n                minValue = v\n            }\n        } while (iterator.hasNext())\n        return minElem\n    }\n\n/**
 * Returns the first element yielding the smallest value of the given function or
 * `null` if there are no elements.\n * 
 * The operation is _terminal_.\n * 
 * The operation is _terminal_.\n */
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\n@LambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) -> Double): Double? \n    {\n        val iterator = iterator()\n        if (!iterator.hasNext()) throw NoSuchElementException()\n        var minValue = selector(iterator.next())\n        while (iterator.hasNext()) {\n            val v = selector(iterator.next())\n            minValue = minOf(minValue, v)\n        }\n        return minValue\n    }\n\n/**
 * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence.\n * 
 * If any of values produced by [selector] function is `NaN`, the returned result
 * is `NaN`.\n * 
 * The operation is _terminal_.\n * 
 * The operation is _terminal_.\n */
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\n@LambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Double): Double? \n    {\n        val iterator = iterator()\n        if (!iterator.hasNext()) return null\n        var minValue = selector(iterator.next())\n        while (iterator.hasNext()) {\n            val v = selector(iterator.next())\n            minValue = minOf(minValue, v)\n        }\n        return minValue\n    }\n\n/**
 * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * 
 * If any of values produced by [selector] function is `NaN`, the returned result
 * is `NaN`.\n * 
 * The operation is _terminal_.\n */
ByLambdaReturnType\n
@kotlin.internal.InlineOnly\n
public inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Float): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the sequence or `null` if there are no elements.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n
public inline fun <T, R : Comparable<R>> Sequence<T>.minOfOrNull(selector: (T) -> R): R? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var minValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the sequence.
 * `null` if there are no elements.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n
public inline fun <T, R> Sequence<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {
    val iterator = iterator()
    if (!iterator.hasNext()) throw NoSuchElementException()
    var minValue = selector(iterator.next())
    while (iterator.hasNext()) {
        val v = selector(iterator.next())
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the sequence or `null` if there are no elements.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n
public inline fun <T> Sequence<T>.minOfWithOrNull(comparator: Comparator<in T>, selector: (T) -> T): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (min > e) min = e
    }
    return min
}

/**
 * Returns the smallest element among all values produced by [selector] function applied to each element in the sequence or `null` if there are no elements.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n
public fun Sequence<Double>.minOrNull(): Double? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        min = minOf(min, e)
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * If any of elements is `NaN` returns `NaN`.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n
public fun Sequence<Float>.minOrNull(): Float? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        min = minOf(min, e)
    }
    return min
}

/**
 * Returns the smallest element or `null` if there are no elements.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n
public fun <T : Comparable<T>>Sequence<T>.minOrNull(): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (min > e) min = e
    }
    return min
}

@Deprecated("Use minWithOrNull instead.", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun <T> Sequence<T>.minWith(comparator: Comparator<in T>): T? {
    return minWithOrNull(comparator)
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n
public fun <T> Sequence<T>.minWithOrNull(comparator: Comparator<in T>): T? {
    val iterator = iterator()
    if (!iterator.hasNext()) return null
    var min = iterator.next()
    while (iterator.hasNext()) {
        val e = iterator.next()
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}
@sample samples.collections.Collections.Aggregates.none

/**
 * Returns `true` if no elements match the given 
 [predicate].
 *
 * The operation is _terminal_ and 
 [stateless].
 */
@SinceKotlin("1.1")
public fun <T> Sequence<T>.none(): Boolean {
    return !iterator().hasNext()
}

/**
 * Returns `true` if no elements match the given 
 [predicate].
 *
 * The 
 [operation] is _terminal_.
 * 
 * @sample samples.collections.Collections.Aggregates.noneWithPredicate
 */
public inline fun <T> Sequence<T>.none(predicate: (T) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns a sequence which performs the given 
 [operation] on each element of the original sequence as they pass through it.
 *
 * The operation is _intermediate_ and 
 [stateless].
 */
@SinceKotlin("1.4")
public fun <T> Sequence<T>.onEachIndexed(operation: (index: Int, T) -> Unit): Sequence<T> {
    return mapIndexed { index, element ->
        operation(index, element)
        element
    }
}

/**
 * Accumulates value starting with the first element and applying 
 [operation] from left to right.
 * to current accumulator value and each element.
 *
 * The operation is _terminal_ and 
 [stateless].
 */
@SinceKotlin("1.4")
public inline fun <S, T : S> Sequence<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.")
    var index = 1
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
        index++
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying 
 [operation] to each element.
 * 
 * Returns `null` if the sequence is empty.
 * 
 * The operation is _intermediate_ and 
 [stateless].
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> Sequence<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return null
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
        index++
    }
    return accumulator
}

@sample samples.collections.Collections.Aggregates.none

@public inline fun <T> Sequence<T>.none(): Boolean {
    return !iterator().hasNext()
}

/**
 * Returns `true` if no elements match the given 
 [predicate].
 *
 * The operation is _terminal_ and 
 [stateless].
 */
@sample samples.collections.Collections.Aggregates.noneWithPredicate

/**
 * Returns `true` if no elements match the given 
 [predicate].
 *
 * The operation is _intermediate_ and 
 [stateless].
 */
@SinceKotlin("1.4")
public fun <T> Sequence<T>.onEachIndexed(operation: (index: Int, T) -> Unit): Sequence<T> {
    return mapIndexed { index, element ->
        operation(index, element)
        element
    }
}

/**
 * Accumulates value starting with the first element and applying 
 [operation] from left to right.
 * to current accumulator value and each element.
 *
 * The operation is _terminal_ and 
 [stateless].
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> Sequence<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) throw UnsupportedOperationException("Empty sequence can't be reduced.")
    var index = 1
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
        index++
    }
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying 
 [operation] from left to right.
 * to current accumulator value and each element.
 *
 * Returns `null` if the sequence is empty.
 * 
 * The operation is _intermediate_ and 
 [stateless].
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun <S, T : S> Sequence<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {
    val iterator = this.iterator()
    if (!iterator.hasNext()) return null
    var accumulator: S = iterator.next()
    while (iterator.hasNext()) {
        accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
        index++
    }
    return accumulator
}
accumulator = operation(accumulator, iterator.next())
\nreturn accumulator

* Returns a sequence containing successive accumulation values generated by applying [operation] from left to right \n* to each element and current accumulator value that starts with [initial] value.
\n* Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting sequence.
\n* The [initial] value should also be immutable (or should not be mutated) as it may be passed to [operation] function later because of sequence's lazy nature.
\n* @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
\n* The operation is _intermediate_ and _stateless_.

* @sample samples.collections.Collections.Aggregates.runningFold

*\n@SinceKotlin("1.4")\npublic fun <T, R> Sequence<T>.runningFold(initial: R, operation: (acc: R, T) -> R): Sequence<R> {
    \n    \n    return sequence {
        yield(initial)
        var accumulator = initial
        for (element in this@runningFold) {
            accumulator = operation(accumulator, element)
            yield(accumulator)
        }
    }
}

*\n@SinceKotlin("1.4")\n@public fun <T, R> Sequence<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {
    \n    \n    return sequence {
        yield(initial)
        var index = 0
        var accumulator = initial
        for (element in this@runningFoldIndexed) {
            accumulator = operation(checkIndexOverflow(index++), accumulator, element)
            yield(accumulator)
        }
    }
}

*\n@SinceKotlin("1.4")\n@public fun <S, T : S> Sequence<T>.runningReduce(operation: (acc: S, T) -> S): Sequence<S> {
    \n    \n    return sequence {
        val iterator = iterator()
        if (iterator.hasNext()) {
            var accumulator: S = iterator.next()
            yield(accumulator)
            while (iterator.hasNext()) {
                accumulator = operation(accumulator, iterator.next())
                yield(accumulator)
            }
        }
    }
}

*\n@SinceKotlin("1.4")\n@public fun <S, T : S> Sequence<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): Sequence<S> {
    \n    \n    return sequence {
        val iterator = iterator()
        if (iterator.hasNext()) {
            var accumulator: S = iterator.next()
            yield(accumulator)
            var index = 0
            while (iterator.hasNext()) {
                accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())
                yield(accumulator)
            }
        }
    }
}
(or should not be mutated)\n * as it may be passed to \[operation\] function later because of sequence's lazy nature.\n * \n * \n * \n * \n * \n * The operation is _intermediate_ and _stateless_. \n * \n * @param \[operation\] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * Returns a sequence containing successive accumulation values generated by applying \n * operation from left to right.\n * to each element, its index in the original sequence and current accumulator value \n * that starts with \[initial\] value.\n * \n * Note that 'acc' value passed to \[operation\] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * The \[initial\] value should also be immutable \n * as it may be passed to \[operation\] function later because of sequence's lazy nature.\n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * The operation is _intermediate_ and _stateless_. \n * \n * @sample samples.collections.Collections.Aggregates.scan

\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R>
Sequence<T>.scan(initial: R, operation: (acc: R, T) -> R): Sequence<R> {
    return runningFold(initial, \n    operation)\n}

/**
 * Returns a sequence containing successive accumulation values generated by applying \n * operation from left to right.\n * to each element, its index in the original sequence and current accumulator value \n * that starts with \[initial\] value.\n * \n * Note that 'acc' value passed to \[operation\] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * The \[initial\] value should also be immutable \n * as it may be passed to \[operation\] function later because of sequence's lazy nature.\n * \n * \n * \n * \n * \n * \n * \n * \n * \n * The operation is _intermediate_ and _stateless_. \n * \n * @param \[operation\] function that takes the index of an element, current accumulator value \n * and the element itself, and calculates the next accumulator value.\n * \n * Returns a sequence containing successive accumulation values generated by applying \n * operation from left to right.\n * to each element, its index in the original sequence and current accumulator value \n * that starts with \[initial\] value.\n * \n * Note that 'acc' value passed to \[operation\] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * The \[initial\] value should also be immutable \n * as it may be passed to \[operation\] function later because of sequence's lazy nature.\n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * \n * The operation is _intermediate_ and _stateless_. \n * \n * @sample samples.collections.Collections.Aggregates.scanIndexed

\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R>
Sequence<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {
    return runningFoldIndexed(initial, \n    operation)\n}

/**
 * Returns a sequence containing successive accumulation values generated by applying \n * operation from left to right.\n * to each element, its index in the original sequence and current accumulator value \n * that starts with \[initial\] value.\n * \n * Note that 'acc' value passed to \[operation\] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * The \[initial\] value should also be immutable \n * as it may be passed to \[operation\] function later because of sequence's lazy nature.\n * \n * \n * \n * \n * \n * \n * \n * \n * \n * The operation is _intermediate_ and _stateless_. \n * \n * @sample samples.collections.Collections.Aggregates.scanReduce

\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi::class)\npublic fun <S, T : S>
Sequence<T>.scanReduce(operation: (acc: S, T) -> S): Sequence<S> {
    return runningReduce(operation)\n}

/**
 * Returns a sequence containing successive accumulation values generated by applying \n * operation from left to right.\n * to each element, its index in the original sequence and current accumulator value \n * that starts with \[initial\] value.\n * \n * Note that 'acc' value passed to \[operation\] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * The \[initial\] value should also be immutable \n * as it may be passed to \[operation\] function later because of sequence's lazy nature.\n * \n * \n * \n * \n * \n * \n * \n * The operation is _intermediate_ and _stateless_. \n * \n * @sample samples.collections.Collections.Aggregates.scanReduceIndexed

\n@SinceKotlin("1.3")\n@ExperimentalStdlibApi::class)\npublic fun <S, T : S>
Sequence<T>.scanReduceIndexed(operation: (index: Int, acc: S, T) -> S): Sequence<S> {
    return runningReduceIndexed(operation)\n}

/**
 * Returns the sum of all values produced by \[selector\] function applied to each element in the sequence.\n * \n * The operation is _terminal_. \n * \n * @param \[selector\] function that takes each element of the sequence and returns its value as \n * \n * @return \n * \n * The sum of all values produced by \[selector\] function applied to each element in the sequence.\n * \n * The operation is _terminal_. \n * \n * @sample samples.collections.Collections.Aggregates.sumBy

\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfDouble")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.sumOf(selector: (T) -> Double): Double {
    var sum: Double = 0.toDouble()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by \[selector\] function applied to each element in the sequence.\n * \n * The operation is _terminal_. \n * \n * @param \[selector\] function that takes each element of the sequence and returns its value as \n * \n * @return \n * \n * The sum of all values produced by \[selector\] function applied to each element in the sequence.\n * \n * The operation is _terminal_. \n * \n * @sample samples.collections.Collections.Aggregates.sumOfInt

\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfInt")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.sumOf(selector: (T) -> Int): Int {
    var sum: Int = 0.toInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by \[selector\] function applied to each element in the sequence.\n * \n * The operation is _terminal_. \n * \n * @param \[selector\] function that takes each element of the sequence and returns its value as \n * \n * @return \n * \n * The sum of all values produced by \[selector\] function applied to each element in the sequence.\n * \n * The operation is _terminal_. \n * \n * @sample samples.collections.Collections.Aggregates.sumOfLong

\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfLong")\n@kotlin.internal.InlineOnly\npublic inline fun <T>
Sequence<T>.sumOf(selector: (T) -> Long): Long {
    var sum: Long = 0.toLong()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}
ByLambdaReturnType\n@kotlin.jvm.JvmName("sumOfUInt")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> UInt): UInt {
    var sum: UInt = 0.toUInt()
    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the sequence.
 * The operation is _terminal_.
 */
@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
@LambdaReturnType
@kotlin.jvm.JvmName("sumOfULong")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> ULong): ULong {
    var sum: ULong = 0.toULong()\n    for (element in this) {
        sum += selector(element)
    }
    return sum
}

/**
 * Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.
 * The operation is _intermediate_ and _stateless_.
 */
public fun <T : Any> Sequence<T>.requireNoNulls(): Sequence<T> {
    return map { it ?: throw IllegalArgumentException("null element found in $this.") }
}

/**
 * Splits this sequence into a sequence of lists each not exceeding the given size.
 * The last list in the resulting sequence may have less elements than the given size.
 * * @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this sequence.
 * * The operation is _intermediate_ and _stateful_.
 */
@SinceKotlin("1.2")
public fun <T> Sequence<T>.chunked(size: Int): Sequence<List<T>> {
    return windowed(size, size, partialWindows = true)
}

/**
 * Splits this sequence into several lists each not exceeding the given size
 * and applies the given [transform] function to an each.
 * * @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this sequence.
 * * The operation is _intermediate_ and _stateful_.
 * * @sample text.Strings.chunkedTransform
 */
@SinceKotlin("1.2")
public fun <T, R> Sequence<T>.chunked(size: Int, transform: (List<T>) -> R): Sequence<R> {
    return windowed(size, size, partialWindows = true, transform = transform)
}

/**
 * Returns a sequence containing all elements of the original sequence except the given [element].
 * The operation is _intermediate_ and _stateful_.
 */
public operator fun <T> Sequence<T>.minus(element: T): Sequence<T> {
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            var removed = false
            return this@minus.filter { if (!removed && it == element) { removed = true; false } else true }.iterator()
        }
    }
}

/**
 * Returns a sequence containing all elements of original sequence except the elements contained in the given [elements] array.
 * Note that the source sequence and the array being subtracted are iterated only when an `iterator` is requested from it.
 * The operation is _intermediate_ and _stateful_.
 */
public operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {
    if (elements.isEmpty()) return this
    return object: Sequence<T> {
        override fun iterator(): Iterator<T> {
            val other = elements.toHashSet()
            if }
(other.isEmpty())
        return this@minus.iterator()
    else
        return this@minus.filterNot { it in other }.iterator()

    }
}

/**
 * Returns a sequence containing all elements of original sequence except the elements contained in the given [elements] sequence.
 * Note that the source sequence and the sequence being subtracted are iterated only when an `iterator` is requested from it.
 * The operation is _intermediate_ for this sequence and _terminal_ and _stateful_ for the [elements] sequence.
 */
public operator fun <T> Sequence<T>.minus(elements: Sequence<T>): Sequence<T> {
    return sequenceOf(this, sequenceOf(elements).asSequence()).flatten()
}

/**
 * Returns a sequence containing all elements of original sequence and then the given [element].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.plus(element: T): Sequence<T> {
    return sequenceOf(this, sequenceOf(element)).flatten()
}

/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] array.
 * Note that the source sequence and the array being added are iterated only when an `iterator` is requested from it.
 * The operation is _intermediate_ and _stateless_.
 */
public operator fun <T> Sequence<T>.plus(elements: Array<out T>): Sequence<T> {
    return sequenceOf(this, elements.asList()).flatten()
}

/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] collection.
 * Note that the source sequence and the collection being added are iterated only when an `iterator` is requested from it.
 * The operation is _intermediate_ and _stateless_.
 */
public operator fun <T> Sequence<T>.plus(elements: Iterable<T>): Sequence<T> {
    return sequenceOf(this, elements.asSequence()).flatten()
}

/**
 * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] sequence.
 * Note that the source sequence and the sequence being added are iterated only when an `iterator` is requested from it.
 * The operation is _intermediate_ and _stateless_.
 */
public operator fun <T> Sequence<T>.plus(elements: Sequence<T>): Sequence<T> {
    return sequenceOf(this, elements).flatten()
}

/**
 * Returns a sequence of snapshots of the window of the given [size] sliding along this sequence with the given [step], where each
 * snapshot is a list.
 * Several last lists may have less elements than the given [size].
 */

Open Source Used In DNAC Platform Antelope 13896
public fun <T> Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<Sequence<T>>
    return windowedSequence(size, step, partialWindows, reuseBuffer = true)

    return windowedSequence(size, step, partialWindows, reuseBuffer = false)

/**
 * Returns a sequence of values built from the elements of 'this' sequence and the [other] sequence with the same index.
 * The resulting sequence ends as soon as the shortest input sequence ends.
 * The operation is _intermediate_ and _stateless_.
 * @sample samples.collections.Sequences.Transformations.averageWindows
 */
    return windowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform)

/**
 * Returns a sequence of pairs of each two adjacent elements in this sequence.
 * The returned sequence is empty if this sequence contains less than two elements.
 * The operation is _intermediate_ and _stateless_.
 * @sample samples.collections.Sequences.Transformations.zipWithNext
 */
@SinceKotlin("1.2")
public fun <T, R> Sequence<T>.zipWithNext(transform: (a: T, b: T) -> R): Sequence<R>
    return sequence result@{
        val iterator = iterator()
        if (!iterator.hasNext()) return@result
        var current = iterator.next()
        while (iterator.hasNext()) {
            val next = iterator.next()
            yield(transform(current, next))
            current = next
        }
    }

/**
 * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.
 * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit] elements will be appended, followed by the [truncated] string (which defaults to "...").
 * The operation is _terminal_.
 * @sample samples.collections.Collections.Transformations.joinTo
 */
public fun <T, A : Appendable> Sequence<T>.joinTo(buffer: A, separator: CharSequence = ", ", prefix: CharSequence = "", postfix: CharSequence = "", limit: Int = -1, truncated: CharSequence = "\"...\"\", transform: (T) -> CharSequence)?: null
    return sequence result@{
        val iterator = iterator()
        if (!iterator.hasNext()) return@result
        var current = iterator.next()
        while (iterator.hasNext()) {
            val next = iterator.next()
            yield(transform(next))
            current = next
        }
    }

    return buffer.append(prefix) 
        buffer.append(separator) 
        buffer.append(last) 
        buffer.append(postfix) 

return buffer

    return buffer

[limit], in which case only the first [limit]n * elements will be appended, followed by the [truncated] string (which defaults to ",",n *).n * The operation is _terminal_.n * \n @sample
samples.collections.Collections.Transformations.joinToString
*/
public fun <T>
    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()
}

/**
 * Creates an [Iterable] instance that wraps the original sequence returning its elements when being iterated.
 */
public fun <T>
Sequence<T>.asIterable(): Iterable<T> {
    return Iterable { this.iterator() }
}

/**
 * Returns this sequence as
 * a [Sequence].
 */
@kotlin.internal.InlineOnly
public inline fun <T> Sequence<T>.asSequence(): Sequence<T> {
    return this
}

/**
 * Returns an average value of elements in the sequence.
 */
public fun Sequence<Byte>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 */
public fun Sequence<Short>.average(): Double {
    var sum:Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 */
public fun Sequence<Int>.average(): Double {
    var sum: Double = 0.0
    var count: Int = 0
    for (element in this) {
        sum += element
        checkCountOverflow(++count)
    }
    return if (count == 0) Double.NaN else sum / count
}

/**
 * Returns an average value of elements in the sequence.
 */
public fun Sequence<Long>.average(): Double {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns an average value of elements in the sequence.
 */
public fun Sequence<Float>.average(): Double {
    var sum:Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns an average value of elements in the sequence.
 */
public fun Sequence<Double>.average(): Double {
    var sum:Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 */
public fun Sequence<Byte>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 */
public fun Sequence<Short>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 */
public fun Sequence<Int>.sum(): Int {
    var sum: Int = 0
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 */
public fun Sequence<Long>.sum(): Long {
    var sum: Long = 0L
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 */
public fun Sequence<Float>.sum(): Float {
    var sum: Float = 0.0f
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the sequence.
 */
public fun Sequence<Double>.sum(): Double {
    var sum: Double = 0.0
    for (element in this) {
        sum += element
    }
    return sum
}

"/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

/*
 * Contributors:
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("SetsKt")

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib
//

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns a set containing all elements of the original set except the given [element].
 * The returned set preserves the element iteration order of the original set.
 * 
 * @param element the element to remove
 * @return a set containing all elements of the original set except the given [element]
 */
public operator fun <T> Set<T>.minus(element: T): Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(size))
    var removed = false
    return this.filterTo(result) { if (!removed && it == element) { removed = true; false } else true }
}

/**
 * Returns a set containing all elements of the original set except the elements contained in the given [elements] array.
 * The returned set preserves the element iteration order of the original set.
 * 
 * @param elements the elements to remove
 * @return a set containing all elements of the original set except the given [elements]
 */
public operator fun <T> Set<T>.minus(elements: Array<out T>): Set<T> {
    val result = LinkedHashSet<T>(this)
    result.removeAll(elements)
    return result
}

/**
 * Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.
 * The returned set preserves the element iteration order of the original set.
 * 
 * @param elements the elements to remove
 * @return a set containing all elements of the original set except the given [elements]
 */
public operator fun <T> Set<T>.minus(elements: Iterable<T>): Set<T> {
    val result = LinkedHashSet<T>(this)
    result.removeAll(elements)
    return result
}

/**
 * Returns a set containing all elements of the original set except the given [element].
 * The returned set preserves the element iteration order of the original set.
 * 
 * @param element the element to remove
 * @return a set containing all elements of the original set except the given [element]
 */
@kotlin.internal.InlineOnly
public inline fun <T> Set<T>.minusElement(element: T): Set<T> {
    return minus(element)
}

/**
 * Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * 
 * @param element the element to add
 * @return a set containing all elements of the original set and then the given [element]
 */
public operator fun <T> Set<T>.plus(element: T): Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(size + 1))
    result.addAll(this)
    result.add(element)
    return result
}

/**
 * Returns a set containing all elements of the original set and the given [elements] array,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * 
 * @param elements the elements to add
 * @return a set containing all elements of the original set and the given [elements]
 */
public operator fun <T> Set<T>.plus(elements: Array<out T>): Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(this.size + elements.size))
    result.addAll(this)
    result.addAll(elements)
    return result
}

/**
 * Returns a set containing all elements of the original set and the given [elements] sequence,
 * which aren't already in this set.
 * The returned set preserves the element iteration order of the original set.
 * 
 * @param elements the elements to add
 * @return a set containing all elements of the original set and the given [elements]
 */
public operator fun <T> Set<T>.plus(elements: Sequence<T>): Set<T> {
    val result = LinkedHashSet<T>(mapCapacity(this.size + 2))
    result.addAll(this)
    result.addAll(elements)
    return result
}
element iteration order of the original set.

```kotlin
Set<T> plusElement(element: T): Set<T> { return plus(element).unmodifiable(); }
```

* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.

Package `kotlin.text`

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt

```
import kotlin.random.*

/**
 * Returns a character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char sequence.
 * @sample samples.collections.Collections.Elements.elementAt
 */
public fun CharSequence.elementAt(index: Int): Char

/**
 * Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.
 * @sample samples.collections.Collections.Elements.elementAt_elseOrThrow
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns the first character.
 * @throws NoSuchElementException if the char sequence is empty.
 */
public fun CharSequence.first(): Char {
    if (isEmpty())
        throw NoSuchElementException("Char sequence is empty.")
    return this[0]
}

/**
 * Returns the first character matching the given [predicate], or `null` if no such character was found.
 * @throws NoSuchElementException if the char sequence is empty.
 */
public inline fun CharSequence.first(predicate: (Char) -> Boolean): Char {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Char sequence contains no character matching the predicate.")
}

/**
 * Returns the last character.
 * @throws NoSuchElementException if the char sequence is empty.
 */
public fun CharSequence.last(): Char {
    if (isEmpty())
        throw NoSuchElementException("Char sequence is empty.")
    return this[lastIndex]
}

/**
 * Returns the last character matching the given [predicate], or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.
 * @sample samples.collections.Collections.Elements.lastOrDefault
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.lastOrDefault(index: Int, defaultValue: (Int) -> Char): Char {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```
@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.

@throws NoSuchElementException if no such character is found.
IllegalArgumentException if \( n \) is negative.\n
* @sample samples.text.Strings.drop
*/
public fun String.dropLast(n: Int): String {
    require(n >= 0) { "Requested character count \( n \) is less than zero." }
    return take((length - n).coerceAtLeast(0)).unaryPlus()
}

* Returns a subsequence of this char sequence containing all characters except last characters that satisfy the given \[ predicate \].\n
* @sample samples.text.Strings.drop
*/
public fun CharSequence.dropLastWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in lastIndex downTo 0) if (!predicate(this[index]))
        return subSequence(0, index + 1)
    return ""
}

* Returns a string containing all characters except last characters that satisfy the given \[ predicate \].\n
* @sample samples.text.Strings.drop
*/
public fun String.dropLastWhile(predicate: (Char) -> Boolean): String {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index]))
            return substring(0, index + 1)
    return ""
}

* Returns a subsequence of this char sequence containing all characters except first characters that satisfy the given \[ predicate \].\n
* @sample samples.text.Strings.drop
*/
public fun CharSequence.dropWhile(predicate: (Char) -> Boolean): CharSequence {
    for (index in this.indices)
        if (!predicate(this[index]))
            return subSequence(index, length)
    return ""
}

* Returns a string containing all characters except first characters that satisfy the given \[ predicate \].\n
* @sample samples.text.Strings.drop
*/
public fun String.dropWhile(predicate: (Char) -> Boolean): String {
    for (index in this.indices)
        if (!predicate(this[index]))
            return substring(index)
    return ""
}

* Returns a char sequence containing only those characters from the original char sequence that match the given \[ predicate \].\n
* @param \[ predicate \] function that takes the index of a character and the character itself\n
* and returns the result of predicate evaluation on the character.\n
* @sample samples.collections.Collections.Filtering.filter
*/
public inline fun String.filter(predicate: (Char) -> Boolean): String {
    return filterIndexedTo(StringBuilder(), predicate).toString()
}

* Returns a char sequence containing only those characters from the original char sequence that do not match the given \[ predicate \].\n
* @param \[ predicate \] function that takes the index of a character and the character itself\n
* and returns the result of predicate evaluation on the character.\n
* @sample samples.collections.Collections.Filtering.filterNot
*/
public inline fun String.filterNot(predicate: (Char) -> Boolean): String {
    return filterNotTo(StringBuilder(), predicate).toString()
}

* Appends all characters matching the given \[ predicate \] to the given \[ destination \].\n
* @param \[ predicate \] function that takes the index of a character and the character itself\n
* and returns the result of predicate evaluation on the character.\n
* @sample samples.collections.Collections.Filtering.filterTo
*/
public inline fun <C : Appendable> CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    forEachIndexed { index, element ->
        if (predicate(index, element)) destination.append(element)
    }
    return destination
}

* Appends all characters not matching the given \[ predicate \] to the given \[ destination \].\n
* @param \[ predicate \] function that takes the index of a character and the character itself\n
* and returns the result of predicate evaluation on the character.\n
* @sample samples.collections.Collections.Filtering.filterTo
*/
public inline fun <C : Appendable> CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    forEachIndexed { index, element ->
        if (!predicate(index, element)) destination.append(element)
    }
    return destination
}

Open Source Used In DNAC Platform Antelope 13902
Open Source Used In DNAC Platform Antelope 13903

```kotlin
CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {
    val
    if (predicate(element)) destination.append(element)
    return destination
}

public fun CharSequence.slice(indices: IntRange): CharSequence {
    if (indices.isEmpty()) return this
    return subSequence(indices)
}

/**
 * Returns a char sequence containing characters of the original char sequence at
 * the specified range of [indices].
 */
public fun String.slice(indices: IntRange): String {
    return subString(indices)
}

/**
 * Returns a string containing characters of the original string at specified [indices].
 */
public fun String.slice(indices: Iterable<Int>): String {
    val result = StringBuilder()
    for (index in indices) {
        result.append(subString(index))
    }
    return result
}

/**
 * Returns a subsequence of this char sequence containing the first [n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.
 */
public fun CharSequence.take(n: Int): CharSequence {
    return subSequence(0, n)
}

/**
 * Returns a string containing characters of the original string at specified [indices].
 */
public fun String.take(n: Int): String {
    return substring(0, n)
}

/**
 * Returns a subsequence of this char sequence containing last [n] characters from this string, or the entire string if this string is shorter.
 */
public fun CharSequence.takeLast(n: Int): CharSequence {
    val length = length
    return subSequence(length - n, length)
}

/**
 * Returns a string containing last [n] characters from this string, or the entire string if this string is shorter.
 */
public fun String.takeLast(n: Int): String {
    val length = length
    return substring(length - n, length)
}

/**
 * Returns a char sequence with characters in reversed order.
 */
public fun CharSequence.reversed(): CharSequence {
    return StringBuilder() + this
}

/**
 * Returns a string with characters in reversed order.
 */
public fun String.reversed(): String {
    return StringBuilder().append(this)
}
```

---

**Open Source Used In DNAC Platform Antelope 13903**

- **CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C**
  - For (index in 0 until length) { val element = get(index) if (predicate(element)) destination.append(element) return destination } Returns a char sequence containing characters of the original char sequence at the specified range of [indices].

- **CharSequence.slice(indices: IntRange): CharSequence**
  - If (indices.isEmpty()) return \"\" return subSequence(indices) **Returns a string containing characters of the original string at the specified range of [indices].**

- **String.slice(indices: IntRange): String**
  - If (indices.isEmpty()) return \"\" return substring(indices) **Returns a char sequence containing characters of the original char sequence at specified [indices].**

- **Iterable<Int>: CharSequence**
  - val size = indices.collectionSizeOrDefault(10) if (size == 0) return \"\" val result = StringBuilder() for (i in indices) { result.append(get(i)) } return result

- **String.slice(indices: Iterable<Int>): String**
  - val result = StringBuilder() for (i in indices) { result.append(get(i)) } return result

- **CharSequence.slice(indices: Iterable<Int>): CharSequence**
  - val value = get(index) if (predicate(element)) destination.append(value)

- **CharSequence.slice(indices: IntRange): CharSequence**
  - If (indices.isEmpty()) return \"\" return subSequence(indices)

- **CharSequence.take(n: Int): CharSequence**
  - require(n >= 0) \"Requested character count \n is less than zero.\" return subSequence(0, n.coerceAtMost(length))

- **CharSequence.takeLast(n: Int): CharSequence**
  - require(n >= 0) \"Requested character count \n is less than zero.\" return subSequence(length - n.coerceAtMost(length), length)

- **CharSequence.takeWhile(predicate: (Char) -> Boolean): CharSequence**
  - for (index in lastIndex downTo 0) if (!predicate(this[index])) return subSequence(index + 1, length)

- **CharSequence.slice(indices: IntRange): String**
  - if (indices.isEmpty()) return \"\" return subString(0, length)

- **CharSequence.takeLastWhile(predicate: (Char) -> Boolean): CharSequence**
  - for (index in lastIndex downTo 0) if (!predicate(this[index])) return subSequence(index + 1, length)

- **CharSequence.takeWhile(predicate: (Char) -> Boolean): String**
  - for (index in lastIndex downTo 0) if (!predicate(this[index])) return substring(index + 1)

- **CharSequence.reversed(): CharSequence**
  - return StringBuilder() + this

- **String.reversed(): String**
  - return StringBuilder().append(this)

---

---
same key the last one gets added to the map.\n* \n* @sample samples.text.Strings.associate
*\n
public inline fun <K, V>
CharSequence.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {\n    val capacity =
    mapCapacity(length).coerceAtLeast(16)\n    return associateTo(LinkedHashMap<K, V>(capacity),
    transform)\n}

public inline fun <K, V>
CharSequence.associateBy(keySelector: (Char) -> K): Map<K, Char> {\n    val capacity =
    mapCapacity(length).coerceAtLeast(16)\n    return associateByTo(LinkedHashMap<K, Char>(capacity),
    keySelector)\n}

public inline fun <K, V, M : MutableMap<in K, in V>>
CharSequence.associateByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {\n    for (element in this) {\n        destination.put(keySelector(element), valueTransform(element))\n    }\n    return destination\n}

public inline fun <V, M : MutableMap<in Char, in V>>
CharSequence.associateWithTo(destination: M, valueSelector: (Char) -> V): M {\n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}

public inline fun <V>
CharSequence.associateWith(valueSelector: (@SinceKotlin("1.3.3"))\npublic inline fun <V>
CharSequence.associateWith(valueSelector: (Char) -> V): Map<Char, V> {\n    val result = LinkedHashMap<Char, V>(mapCapacity(length.coerceAtMost(128)).coerceAtLeast(16))\n    return associateWithTo(result, valueSelector)\n}

public inline fun <V, M : MutableMap<in Char, in V>>
CharSequence.associateWithTo(destination: M, valueSelector: (Char) -> V): M {\n    for (element in this) {\n        destination.put(element, valueSelector(element))\n    }\n    return destination\n}
the given [destination] collection.

\*/
public fun <C : MutableCollection<in Char>>

CharSequence.toCollection(destination: C): C {
    for (item in this) {
        destination.add(item)
    }
    return destination
}

/**
 * Returns a new [HashSet] of all characters.
 */
public fun CharSequence.toHashSet(): HashSet<Char> {
    return toCollection(HashSet<Char>(mapCapacity(length.coerceAtMost(128))))
}

/**
 * Returns a [List] containing all characters.
 */
public fun CharSequence.toList(): List<Char> {
    return when (length) {
        0 -> emptyList()
        1 -> listOf(this[0])
        else -> this.toMutableList()
    }
}

/**
 * Returns a new [MutableList] filled with all characters of this char sequence.
 */
public fun CharSequence.toMutableList(): MutableList<Char> {
    return toCollection(ArrayList<Char>(length))
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each character of original char sequence.
 */
public fun CharSequence.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {
    return groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)
}

/**
 * Groups characters of the original char sequence by the key returned by the given [keySelector] function applied to each character and puts to the [destination] map each group key associated with a list of corresponding characters.
 */
public fun CharSequence.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {
    return groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)
}
* @return The [destination] map.
 * 
 * @sample samples.collections.Collections.Transformations.groupBy
 *\n
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<Char>() }
        list.add(element)
    }
    return destination
}

/**
 * Groups values returned by the [valueTransform] function applied to each character of the original char sequence
 * by the key returned by the given [keySelector] function applied to the character in the char sequence
 * and puts to the [destination] map each group key associated with a list of corresponding values.
 * * @return The [destination] map.
 * * @sample samples.collections.Collections.Transformations.groupBy
 *\n
 */
public inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharSequence.mapByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) { ArrayList<V>() }
        list.add(valueTransform(element))
    }
    return destination
}

/**
 * Creates a [Grouping] source from a char sequence to be used later with one of group-and-fold operations
 * using the specified [keySelector] function to extract a key from each character.
 * * @sample samples.collections.Grouping.groupingByEachCount
 */
@SinceKotlin("1.1")
public inline fun <K> CharSequence.groupingBy(crossinline keySelector: (Char) -> K): Grouping<Char, K> {
    return object :
        Grouping<Char, K> {
            override fun sourceIterator(): Iterator<Char> = this@groupingBy.iterator()

            override fun keyOf(element: Char): K = keySelector(element)
        }
}

/**
 * Returns a list containing the results of applying the given [transform] function
 * to each character in the original char sequence.
 * * @param [transform] function that takes the index of a character and the character itself
 * * and returns the result of the transform applied to the character.
 * * @sample samples.text.Strings.map
 */
public inline fun <R> CharSequence.map(transform: (Char) -> R): List<R> {
    return mapTo(ArrayList<R>(length), transform)
}

/**
 * Returns a list containing the results of applying the given [transform] function
 * to each character and its index in the original char sequence.
 * * @param [transform] function that takes the index of a character and the character itself
 * * and returns the result of the transform applied to the character.
 * * @sample samples.collections.Collections.Transformations.mapIndexed
 */
public inline fun <R> CharSequence.mapIndexed(transform: (index: Int, Char) -> R): List<R> {
    return mapIndexedTo(ArrayList<R>(length), transform)
}

/**
 * Returns a list containing only the non-null results of applying the given [transform] function
 * to each character and its index in the original char sequence.
 * * @param [transform] function that takes the index of a character and the character itself
 * * and returns the result of the transform applied to the character.
 * * @sample samples.collections.Collections.Transformations.mapIndexedNotNull
 */
public inline fun <R : Any> CharSequence.mapIndexedNotNull(transform: (index: Int, Char) -> R?): List<R> {
    return mapIndexedNotNullTo(ArrayList<R>(), transform)
}

/**
 * Applies the given [transform] function to each character of the original char sequence
 * and appends the results to the given [destination].
 * * @param [transform] function that takes the index of a character and the character itself
 * * and returns the result of the transform applied to the character.
 * * @sample samples.collections.Collections.Transformations.mapNotNull
 */
public inline fun <R : Any> CharSequence.mapNotNull(transform: (Char) -> R?): List<R> {
    return mapNotNullTo(ArrayList<R>(), transform)
}

/**
 * Applies the given [transform] function to each character in the original char sequence
 * and appends only the non-null results to the given [destination].
 * * @param [transform] function that takes the index of a character and the character itself
 * * and returns the result of the transform applied to the character.
 * * @sample samples.collections.Collections.Transformations.mapNotNullTo
 */
public inline fun <R> CharSequence.mapNotNullTo(destination: R) {
    for (element in this) {
        transform(element) let { destination.add(it) } } 
    return destination
}

/**
 * Applies the given [transform] function to each character in the original char sequence
 * and appends only the non-null results to the given [destination].
 * * @param [transform] function that takes the index of a character and the character itself
 * * and returns the result of the transform applied to the character.
 * * @sample samples.collections.Collections.Transformations.mapNotNullTo
 */
public inline fun <R> CharSequence.mapNotNullTo(destination: R) {
    for (element in this) {
        transform(element) let { destination.add(it) } } 
    return destination
}
public inline fun <R, C : MutableCollection<in R>> CharSequence.mapTo(destination: C, transform: (Char) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each character of the original char sequence into an
 [IndexedValue] containing the index of that character and the character itself. *
 */
public inline fun <R, C : MutableCollection<in R>> CharSequence.mapTo(destination: C, transform: (Char) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns `true` if all characters match the given [predicate]. *
 */
public inline fun CharSequence.all(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if at least one character matches the given [predicate]. *
 */
public fun CharSequence.any(): Boolean {
    return !isEmpty()
}

/**
 * Returns `true` if at least one character matches the given [predicate]. *
 */
public inline fun CharSequence.any(predicate: (Char) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return true
    return false
}

/**
 * Returns the lazy [Iterable] that wraps each character of the original char sequence into an
 [IndexedValue] containing the index of that character and the character itself. *
 */
public fun CharSequence.withIndex(): Iterable<IndexedValue<Char>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns `true` if all characters match the given [predicate]. *
 */
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
}

@deprecated("Use maxOrNull instead.", replaceWith = "maxOrNull()")
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.max(): Char? {
    return maxOrNull()
ReplaceWith("maxByOrNull(selector)")\n@DeprecatedSinceKotlin(warningSince = "\"1.4\"\")\npublic inline fun <R : Comparable<R>> CharSequence.maxBy(selector: (Char) -> R): Char? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem[0]\n    var maxValue = selector(maxElem[0]) for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return maxElem[0]\n} \n
/**\n * Returns the first character yielding the largest value of the given function or `null` if there are no characters.\n * @example samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\nInlineOnly\npublic inline fun CharSequence.maxByOrNull(selector: (Char) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0]) for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        var maxVal = maxOf(maxValue, v)\n        return maxVal\n    }\n    return null\n} \n
/**\n * Returns the first character yielding the largest value of the given function or `null` if there are no characters.\n * @example samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\npublic inline fun <R : Comparable<R>> CharSequence.maxByOrNull(selector: (Char) -> R): R? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem[0]\n    var maxValue = selector(maxElem[0]) for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return maxElem[0]\n} \n
/**\n * Returns the largest value among all values produced by [selector] function applied to each character in the char sequence.\n * @example samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\nInlineOnly\npublic inline fun CharSequence.maxOf(selector: (Char) -> Double): Double {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0]) for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        var maxVal = maxOf(maxValue, v)\n        return maxVal\n    }\n    return null\n} \n
/**\n * Returns the largest value among all values produced by [selector] function applied to each character in the char sequence or `null` if there are no characters.\n * @example samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\nInlineOnly\npublic inline fun CharSequence.maxOfOrNull(selector: (Char) -> Double): Double? {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0]) for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        var maxVal = maxOf(maxValue, v)\n        return maxVal\n    }\n    return null\n} \n
/**\n * Returns the largest value among all values produced by [selector] function applied to each character in the char sequence or `null` if there are no characters.\n * @example samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\nInlineOnly\npublic inline fun CharSequence.maxOfOrNull(selector: (Char) -> R): R? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem[0]\n    var maxValue = selector(maxElem[0]) for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return maxElem[0]\n} \n
/**\n * Returns the largest value among all values produced by [selector] function applied to each character in the char sequence or `null` if there are no characters.\n * @example samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\nInlineOnly\npublic inline fun CharSequence.maxOfOrNull(selector: (Char) -> Float): Float? {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0]) for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        var maxVal = maxOf(maxValue, v)\n        return maxVal\n    }\n    return null\n} \n
/**\n * Returns the largest value among all values produced by [selector] function applied to each character in the char sequence or `null` if there are no characters.\n * @example samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution\nByLambdaReturnType\nInlineOnly\npublic inline fun <R : Comparable<R>> CharSequence.maxOfOrNull(selector: (Char) -> R): R? {\n    if (isEmpty()) return null\n    var maxElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return maxElem[0]\n    var maxValue = selector(maxElem[0]) for (i in 1..lastIndex) {\n        val e = this[i]\n        val v = selector(e)\n        if (maxValue < v) {\n            maxElem = e\n            maxValue = v\n        }\n    }\n    return maxElem[0]\n} \n
Open Source Used In DNAC Platform Antelope 13908
maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each character in the char sequence.
 * @throws NoSuchElementException if the char sequence is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R> CharSequence.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) { val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each character in the char sequence.
 * @throws NoSuchElementException if the char sequence is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R> CharSequence.maxOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) { val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator] among all values produced by [selector] function applied to each character in the char sequence or 'null' if there are no characters.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R> CharSequence.maxOfOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) { val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest character or 'null' if there are no characters.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public fun <R> CharSequence.maxOrNull(): R? {
    if (isEmpty()) return null
    var max = this[0]
    for (i in 1..lastIndex) { val e = this[i]
        if (max < e) max = e
    }
    return max
}

@Deprecated("Use maxWithOrNull instead.", ReplaceWith("maxWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.maxWith(comparator: Comparator<in Char>): Char? {
    return maxWithOrNull(comparator)
}

/**
 * Returns the first character yielding the smallest value of the given function or 'null' if there are no characters.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> CharSequence.minByOrNull(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var minElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return minElem
    var minValue = selector(minElem)
    for (i in 1..lastIndex) { val e = this[i]
        val v = selector(e)
        if (minValue > v) {
            minElem = e
            minValue = v
        }
    }
    return minElem
}

/**
 * Returns the smallest value among all values produced by [selector] function or `null` if there are no characters.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> CharSequence.minByOrNull(selector: (Char) -> R): R? {
    if (isEmpty()) return null
    var minElement = selector(minElement)
    for (i in 1..lastIndex) { val e = this[i]
        minElement = minOf(minElement, e)
        if (minElement < minElement) minElement = e
    }
    return minElement
}

@Deprecated("Use minOrNull instead.", ReplaceWith("minOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.min(): Char? {
    return minOrNull()
}

@Deprecated("Use minByOrNull instead.", ReplaceWith("minByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public inline fun <R : Comparable<R>> CharSequence.minByOrNull(selector: (Char) -> R): Char? {
    if (isEmpty()) return null
    var minElement = selector(minElement)
    for (i in 1..lastIndex) { val e = this[i]
        minElement = minOf(minElement, e)
        if (minElement < minElement) minElement = e
    }
    return minElement
}
1..lastIndex) {
    val v = selector(this[i])
    minValue = minOf(minValue, v)
} return
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each character
 * in the char sequence.
 * @throws NoSuchElementException if the char sequence is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> CharSequence.minOf(selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each character
 * in the char sequence or `null` if there are no characters.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun CharSequence.minOfOrNull(selector: (Char) -> Double?): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each character
 * in the char sequence or `null` if there are no characters.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun CharSequence.minOfOrNull(selector: (Char) -> Float?): Float? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each character
 * in the char sequence or `null` if there are no characters.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun <R> CharSequence.minOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each character
 * in the char sequence or `null` if there are no characters.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@kotlin.internal.InlineOnly
public inline fun CharSequence.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the first character having the smallest value according to the
 * provided [comparator] function.
 */
@SinceKotlin("1.4")
public fun CharSequence.minOrNull(): Char? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (min > e) min = e
    }
    return min
}

@Deprecated("Use minWithOrNull instead.", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
public fun CharSequence.minWith(comparator: Comparator<in Char>): Char? {
    return minWithOrNull(comparator)
}

/**
 * Returns the first character having the smallest value according to the
provided [comparator] or `null` if there are no characters.

```kotlin
 CharSequence.minWithOrNull(comparator: Comparator<in Char>): Char? {
    var min = this[0] for (i in 1..lastIndex) {
        val e = this[i] if (comparator.compare(min, e) > 0) min = e
        return min
    } Returns `true` if the char sequence has no characters.

    * @sample samples.collections.Collections.Aggregates.none

    } * Returns `true` if no characters match the given [predicate].

    * @sample samples.collections.Collections.Aggregates.withPredicate

    */
@SinceKotlin("1.4")
public fun CharSequence.minWithOrNull(comparator: Comparator<in Char>): Char? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns `true` if the char sequence can't be reduced.
 * 
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public inline fun CharSequence.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}```
from right to left\n* to each character and current accumulator value.\n* \n* \n* Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n* \n* please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n* \n* \n* @param [operation] function that takes a character and current accumulator value,\n* \n* and calculates the next accumulator value.\n* \n* \n* @sample samples.collections.Collections.Aggregates.reduceRightOrNull

CharSequence.reduceRight(operation: (Char, acc: Char) -> Char): Char {\n    \n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.\n")\n    \n    var accumulator = get(index-\n    while (index >= 0) {\n        \n        accumulator = operation(get(index--), accumulator)\n    }\n    \n    return accumulator\n}

CharSequence.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {\n    \n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException("Empty char sequence can't be reduced.\n")\n
    var accumulator = get(index--}\n    while (index >= 0) {\n        \n        accumulator = operation(index, get(index), accumulator)\n        --index
    }\n    \n    return accumulator
}

CharSequence.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? {\n    \n    var index = lastIndex\n    if (index < 0) return null\n
    var accumulator = get(index--}\n    while (index >= 0) {\n        \n        accumulator = operation(index, get(index), accumulator)\n        --index
    }\n    \n    return accumulator
}

CharSequence.runningFold(initial: R, operation: (acc: R, Char) -> R): List<R> {\n    \n    if (isEmpty()) return listOf(initial)\n    \n    val result = ArrayList<R>(length + 1).apply { add(initial) }\n    \n    var accumulator = initial\n    for (element in this) {\n        \n        accumulator = operation(accumulator, element)\n    }\n    \n    return result
}

CharSequence.runningFoldIndexed(initial: R, operation: (index: Int, Char, acc: Char) -> R): List<R> {\n    \n    if (isEmpty()) return listOf(initial)\n
    val result = ArrayList<R>(length + 1).apply { add(initial) }\n    \n    var accumulator = initial\n
    for (element in this) {\n        \n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n    \n    return result
}
Int, acc: R, Char) -> R: List<R> { 
    if (isNotEmpty()) return listoff(initial)
    val result = ArrayList<R>(length + 1).apply { 
        add(initial)
    } var accumulator = initial
    for (index in indices) { 
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each character and current accumulator value that starts with the first character of this char sequence.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * Otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and a character, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@ExperimentalStdlibApi
public inline fun <R> CharSequence.scan(initial: R, operation: (acc: R, Char) -> R): List<R> { 
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each character and current accumulator value that starts with the first character of this char sequence.
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * Otherwise it would affect the previous value in resulting list.
 * @param [operation] function that takes current accumulator value and a character, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@ExperimentalStdlibApi
public inline fun <R> CharSequence.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> { 
    return runningFoldIndexed(initial, operation)
}
```kotlin
int var sum; int = 0
for (element in this) {
    sum += selector(element);
} return sum

// A public inline fun CharSequence.sumByDouble(selector: (Char) -> Double): Double {
int var sum; Double = 0
for (element in this) {
    sum += selector(element);
} return sum
}

/**
 * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@kotlin.experimental.ExperimentalTypeInference
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
public inline fun CharSequence.sumOf(selector: (Char) -> Int): Int {
    int var sum; Int = 0
    for (element in this) {
        sum += selector(element);
    } return sum
}

/**
 * Splits this char sequence into several char sequences each not exceeding the given [size].
 * The last string in the resulting sequence may have less characters than the given [size].
 * Must be positive and can be greater than the number of elements in this char sequence.
 */
@SinceKotlin("1.2")
public fun CharSequence.chunked(size: Int): List<String> {
    return windowed(size, size, partialWindows = true)
}
```

---

Open Source Used In DNAC Platform Antelope 13914
exceeding the given `size` and applies the given [transform] function to each sequence. Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function. You should not store it or allow it to escape in any way, unless you made a snapshot of it. The last char sequence may have less characters than the given `size`. The number of elements to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence. 

```kotlin
CharSequence.chunkedSequence(size: Int, transform: (CharSequence) -> R): Sequence<R> {
    return windowedSequence(size, size, partialWindows = true, transform = transform)
}

CharSequence.partition(predicate: Boolean): Pair<CharSequence, CharSequence> {
    val first = StringBuilder()
    val second = StringBuilder()
    for (element in this) {
        if (predicate(element)) {
            first.append(element)
        } else {
            second.append(element)
        }
    }
    return Pair(first, second)
}

CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<String> {
    return windowed(size, step, partialWindows) { it.toString() }
}
```

```kotlin
CharSequence.chunkedTransformToSequence(size: Int, transform: (CharSequence) -> R): Sequence<String> {
    return windowedSequence(size, size, partialWindows = true, transform = transform)
}
```

```kotlin
CharSequence.chunkedSequence(size: Int, transform: (CharSequence) -> R): Sequence<String> {
    return windowedSequence(size, size, partialWindows = true, transform = transform)
}
```

```kotlin
CharSequence.averageWindows(size: Int, step: Int = 1, partialWindows: Boolean = false): List<Double> {
    return averageWindows(size, step, partialWindows) { it.toDouble() }
}
```

```kotlin
CharSequence.averageWindows(size: Int, step: Int = 1, partialWindows: Boolean = false): List<Double> {
    return averageWindows(size, step, partialWindows) { it.toDouble() }
}
```

```kotlin
CharSequence.averageWindows(size: Int, step: Int = 1, partialWindows: Boolean = false): List<Double> {
    return averageWindows(size, step, partialWindows) { it.toDouble() }
}
```

```kotlin
CharSequence.averageWindows(size: Int, step: Int = 1, partialWindows: Boolean = false): List<Double> {
    return averageWindows(size, step, partialWindows) { it.toDouble() }
}
```

```kotlin
CharSequence.averageWindows(size: Int, step: Int = 1, partialWindows: Boolean = false): List<Double> {
    return averageWindows(size, step, partialWindows) { it.toDouble() }
}
```

```kotlin
CharSequence.averageWindows(size: Int, step: Int = 1, partialWindows: Boolean = false): List<Double> {
    return averageWindows(size, step, partialWindows) { it.toDouble() }
}
```
window
* @param step the number of elements to move the window forward by on each step, by default 1
* @param partialWindows controls whether or not to keep partial windows in the end if any,
* by default `false` which means partial windows won't be preserved
* @sample samples.collections.Sequences.Transformations.takeWindows

public fun <R> CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (CharSequence) -> R): Sequence<R> {
    checkWindowSizeStep(size, step)
    val windows = (if (partialWindows) indices else 0 until length - size + 1) step step
    return windows.map { index ->
        val end = index + size
        val coercedEnd = if (end < 0 || end > length) length else end
        transform(subSequence(index, coercedEnd))
    }
}

public infix fun CharSequence.zip(other: CharSequence): List<Pair<Char, Char>> {
    return zip(other) { c1, c2 -> c1 to c2 }
}

public inline fun <V> CharSequence.zip(other: CharSequence, transform: (a: Char, b: Char) -> V): List<V> {
    val length = minOf(this.length, other.length)
    val list = ArrayList<V>(length)
    for (i in 0 until length) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

public fun CharSequence.zipWithNext(): List<Pair<Char, Char>> {
    return zipWithNext { a, b -> a to b }
}

public inline fun <R> CharSequence.zipWithNext(transform: (a: Char, b: Char) -> R): List<R> {
    val size = length - 1
    if (size < 1) return emptyList()
    val result = ArrayList<R>(size)
    for (index in 0 until size) {
        result.add(transform(this[index], this[index + 1]))
    }
    return result
}

public fun CharSequence.asIterable(): Iterable<Char> {
    if (this is String && isEmpty()) return emptyList()
    return this::iterator
}

public fun CharSequence.asSequence(): Sequence<Char> {
    if (this is String && isEmpty()) return emptySequence()
    return this::iterator
}

package

Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.


leading and trailing characters matching the [predicate] removed.

CharSequence.trim(predicate: (Char) -> Boolean): CharSequence {
    var startFound = false
    var endFound = false
    var startIndex = 0
    var endIndex = length - 1
    val startIndex = startFound
    val endIndex = endFound
    while (startIndex <= endIndex) {
        val match = predicate(this[index])
        if (!startFound) {
            if (!match)
                startFound = true
            else
                startIndex += 1
        } else {
            if (!match)
                break
            else
                endIndex -= 1
        }
    }
    return subSequence(startIndex, endIndex + 1)
}

String.trim(predicate: (Char) -> Boolean): String =
    this as CharSequence.trim(predicate).toString()

String.trimStart(predicate: (Char) -> Boolean): String {
    var resulted = this
    for (index in resulted.indices)
        if (!predicate(resulted[index]))
            return subSequence(index, length)
    return ""
}

String.trimEnd(predicate: (Char) -> Boolean): String {
    var resulted = this
    for (index in resulted.indices.reversed())
        if (!predicate(resulted[index]))
            return subSequence(0, index)
    return ""
}

String.trim(vararg chars: Char): String =
    this as CharSequence.trim(vararg chars)

String.trimStart(vararg chars: Char): String =
    this as CharSequence.trimStart(vararg chars)

String.trimEnd(vararg chars: Char): String =
    this as CharSequence.trimEnd(vararg chars)

CharSequence.trim(): CharSequence = trim(Char::isWhitespace)

String.trim(): String = this as CharSequence.trim().toString()

CharSequence.trimStart(): CharSequence = trimStart(Char::isWhitespace)

String.trimStart(): String = this as CharSequence.trimStart().toString()

CharSequence.trimEnd(): CharSequence = trimEnd(Char::isWhitespace)

String.trimEnd(): String = this as CharSequence.trimEnd().toString()
character indices for this char sequence.

```kotlin
//public val CharSequence.indices: IntRange
    get() = 0..length - 1

//public val CharSequence.lastIndex: Int
    get() = this.length - 1

//public fun CharSequence.hasSurrogatePairAt(index: Int): Boolean {
    return index in 0..length - 2
        && this[index].isHighSurrogate()
        && this[index + 1].isLowSurrogate()
}

//public val CharSequence.indices: IntRange
    get() = 0..length - 1

//public val CharSequence.lastIndex: Int
    get() = this.length - 1

//public fun CharSequence.hasSurrogatePairAt(index: Int): Boolean {
    return index in 0..length - 2
        && this[index].isHighSurrogate()
        && this[index + 1].isLowSurrogate()
}

//public fun String.substring(range: IntRange): String = substring(range.start, range.endInclusive + 1)

//public fun CharSequence.subSequence(range: IntRange): CharSequence = subSequence(range.start, range.endInclusive + 1)

//public inline fun String.subSequence(start: Int, end: Int): CharSequence = subSequence(start, end)

//public inline fun CharSequence.substring(startIndex: Int, endIndex: Int = length): String = substring(0, endIndex)

//public fun String.substringBefore(delimiter: Char, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(0, index)
}

//public fun String.substringBefore(delimiter: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(0, index)
}

//public fun String.substringAfter(delimiter: Char, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(index + 1, length)
}

//public fun String.substringAfter(delimiter: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(index + delimiter.length, length)
}
```
// A string method that returns a substring after the last occurrence of a specified delimiter.

public fun String.substringAfterLast(delimiter: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else substring(index + delimiter.length, length)
}

/**
 * Returns a new string with a character range replaced by a new string.
 * @param old the range to replace.
 * @param replacement the new string.
 */
@OptIn(ExperimentalStdlibApi::class)
public fun CharSequence.replaceRange(old: IntRange, replacement: CharSequence): CharSequence {
    return replaceRange(old.start, old.endInclusive + 1, replacement)
}

/**
 * Removes a part of the string based on a range.
 * @param startIndex the index of the first character to remove.
 * @param endIndex the index of the first character after the removed part to keep.
 */
@kotlin.internal.InlineOnly
public inline fun String.removeRange(startIndex: Int, endIndex: Int): String =
    (this as CharSequence).removeRange(startIndex, endIndex).toString()

/**
 * Returns a new string with a prefix removed.
 * @param prefix the prefix to remove.
 */
@kotlin.internal.InlineOnly
public inline fun String.removePrefix(prefix: CharSequence): String =
    (this as CharSequence).removePrefix(prefix).toString()

/**
 * Returns a substring without a specified suffix.
 * @param suffix the suffix to remove.
 */
public fun CharSequence.removeSuffix(suffix: CharSequence): CharSequence {
    if (endsWith(suffix)) {
        return subSequence(0, length - suffix.length)
    }
    return this
}

/**
 * Returns a new string with a suffix removed.
 * @param suffix the suffix to remove.
 */
@kotlin.internal.InlineOnly
public inline fun String.removeSuffix(suffix: CharSequence): String =
    (this as CharSequence).removeSuffix(suffix).toString()
subSequence(0, length)\n}\n
/**
 * If this string ends with the given \[suffix\], returns a copy of this string\n * with the suffix removed. Otherwise, returns this string.\n */
public fun String.removeSuffix(suffix: CharSequence):
String \n if (endsWith(suffix)) \n return subSequence(0, length - suffix.length)\n \n return this\n
/**
 * When this char sequence starts with the given \[prefix\] and ends with the given \[suffix\],\n * returns a new char sequence having both the given \[prefix\] and \[suffix\] removed.\n * Otherwise returns a new char sequence with the same characters.\n */
public fun CharSequence.removeSurrounding(prefix: CharSequence, suffix: CharSequence):
CharSequence \n if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) \n return subSequence(prefix.length, length - suffix.length)\n \n return subSequence(0, length)\n
/**
 * When this char sequence starts with the given \[prefix\] and ends with the given \[suffix\],\n * returns a new char sequence having both the \[prefix\] and \[suffix\] removed from the start and end.\n * Otherwise returns a new char sequence with the same characters.\n */
public fun String.removeSurrounding(prefix: CharSequence, suffix: CharSequence):
String \n if ((length >= prefix.length + suffix.length) &&
startsWith(prefix) && endsWith(suffix)) \n return substring(prefix.length, length - suffix.length)\n \n return this\n
/**
 * If the string does not contain the delimiter, returns \[missingDelimiterValue\] which defaults to the original string.\n */
public fun String.replaceBefore(delimiter: Char,
replacement: String, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue
    else replaceRange(0, index, replacement)\n}

/**
 * If the string does not contain the delimiter, returns \[missingDelimiterValue\] which defaults to the original string.\n */
public fun String.replaceBefore(delimiter: String,
replacement: String, missingDelimiterValue: String = this): String {
    val index = indexOf(delimiter)
    return if (index == -1) missingDelimiterValue
    else replaceRange(0, index, replacement)\n}

/**
 * If the string does not contain the delimiter, returns \[missingDelimiterValue\] which defaults to the original string.\n */
public fun String.replaceAfter(delimiter: Char,
replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue
    else replaceRange(index + 1, length, replacement)\n}

/**
 * When the string does not contain the delimiter, returns \[missingDelimiterValue\] which defaults to the original string.\n */
public fun String.replaceAfter(delimiter: String,
replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue
    else replaceRange(index + delimiter.length, length, replacement)\n}

/**
 * If the string does not contain the delimiter, returns \[missingDelimiterValue\] which defaults to the original string.\n */
public fun String.replaceBeforeLast(delimiter: Char,
replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue
    else replaceRange(index, length - delimiter.length, replacement)\n}

/**
 * When the string does not contain the delimiter, returns \[missingDelimiterValue\] which defaults to the original string.\n */
public fun String.replaceBeforeLast(delimiter: String,
replacement: String, missingDelimiterValue: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue
    else replaceRange(index, length - delimiter.length, replacement)\n}
val index = lastIndexOf(delimiter)
    return if (index == -1) missingDelimiterValue else replaceRange(0, index,
        *[replacement]*
    )
}

/**
 * Replace part of string before the last occurrence of given delimiter with the
 * [replacement] string.
 */
public fun String.replaceBeforeLast(delimiter: String, replacement: String,
    *missingDelimiterValue*: String = this): String {
    val index = lastIndexOf(delimiter)
    return if (index == -1)
        *missingDelimiterValue* else replaceRange(0, index,
            *replacement*
        )
}

// public fun String.replace(oldChar: Char,
// newChar: Char, ignoreCase: Boolean): String // JVM- and JS-specific
// public fun String.replace(oldValue: String,
// newValue: String, ignoreCase: Boolean): String // JVM- and JS-specific

/**
 * Returns a new string obtained by replacing each substring of this char
 * sequence that matches the given regular expression with the given
 * [replacement] string.
 * Literally escape the replacement string if it contains literal text or
 * $-substitutions.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, *replacement*): String =
    regex.replace(this, *replacement*).

/**
 * Returns a new string obtained by replacing each substring of this char
 * sequence that matches the given regular expression with the result of the
 * given function that takes a [MatchResult] and returns a string to be used as
 * a replacement for that match.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.replace(regex: Regex, *noinline* transform: (MatchResult) -> CharSequence): String =
    regex.replace(this, *transform*).

/**
 * Replaces the first occurrence of the given regular expression in this char
 * sequence with specified [replacement] expression.
 * Literally escape the replacement string if it contains literal text or
 * $-substitutions.
 */
@kotlin.internal.InlineOnly
public inline fun CharSequence.replaceFirst(regex: Regex, *replacement*): String =
    regex.replaceFirst(this, *replacement*).

/**
 * Returns `true` if this char sequence matches the given regular expression.
 */
@kotlin.internal.InlineOnly
public inline infix fun CharSequence.matches(regex: Regex): Boolean =
    regex.matches(this).

/**
 * Implementation of [regionMatches] for CharSequences.
 * Invoked when it's already known that arguments are not Strings, so that no additional type checks are performed.
 */
internal fun CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int,
    ignoreCase: Boolean): Boolean {
    if ((otherOffset < 0) || (thisOffset < 0) || (thisOffset > this.length - length) ||
        (otherOffset > other.length - length)) {
        return false
    }

    for (index in 0 until length) {
        if (!this[thisOffset + index].equals(other[otherOffset + index], ignoreCase))
            return false
    }
    return true
}

/**
 * Returns `true` if this char sequence starts with the specified character.
 * Literally escape the specified character if it contains literal text or
 * $-substitutions.
 */
public fun CharSequence.startsWith(char: Char, ignoreCase: Boolean = false): Boolean =
    this.length > 0 &&
        this[0].equals(char, ignoreCase)

/**
 * Returns `true` if this char sequence ends with the specified character.
 * Literally escape the specified character if it contains literal text or
 * $-substitutions.
 */
public fun CharSequence.endsWith(char: Char, ignoreCase: Boolean = false): Boolean =
    this.length > 0 &&
        this[lastIndex].equals(char, ignoreCase)

/**
 * Returns `true` if this char sequence starts with the specified prefix.
 * Literally escape the specified prefix if it contains literal text or
 * $-substitutions.
 */
public fun CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && prefix is String)
        return this.startsWith(prefix)
    else
        return regionMatchesImpl(0, prefix, 0, prefix.length, ignoreCase)

/**
 * Returns `true` if a substring of this char sequence starting at the specified offset
 * starts with the specified prefix.
 */
public fun CharSequence.startsWith(prefix: CharSequence, startIndex: Int, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase && this is String && prefix is String)
        return this.startsWith(prefix, startIndex)
    else
        return regionMatchesImpl(startIndex, prefix, 0, prefix.length, ignoreCase)

/**
 * Returns the longest string `prefix` such that this char sequence and
 * [other] both start with this prefix,
 * taking care not to split surrogate pairs.
 */
public fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String =
    // implementation...
beginning to the end of this string, and finds at each position the first element in [strings] that matches this string at that position
* @param CharSequence findAnyOf(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Pair<Int, String>? =\n findAnyOf(strings, startIndex, ignoreCase, last = false)\n/* * Find last occurrence of any of the specified [strings] in this char sequence.
 * @param CharSequence findAnyOf(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Int =\n findAnyOf(strings, startIndex, ignoreCase, last = true)\n/* * Find any of the first occurrence of any of the specified [strings] in this char sequence.
 * @param CharSequence indexOfAny(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Int =\n indexOfAny(strings, startIndex, ignoreCase, last = false)\n/* * Find last occurrence of any of the specified [strings] in this char sequence.
 * @param CharSequence lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = 

index of character to start searching at. The search proceeds backward toward the beginning of the string.\n* @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n* @return An index of the last occurrence of [string] or -1 if none is found.\n* @fun public fun CharSequence.lastIndexOf(string: String,\n    startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n    if (ignoreCase || this is String)\n        indexOf(string, startIndex, 0, ignoreCase, last = true)\n    else\n        nativeLastIndexOf(string, startIndex)\n} \n* @suppress("INAPPLICABLE_OPERATOR_MODIFIER")\n* @public inline operator fun CharSequence.contains(regex: Regex): Boolean =\n    regex.containsMatchIn(this)\n* @kotlin.internal.InlineOnly\n* @public inline operator fun CharSequence.contains(char: Char, ignoreCase: Boolean = false): Boolean =\n    indexOf(char, ignoreCase) >= 0\n* @param ignoreCase `true` to ignore character case when comparing strings. By default `false`.\n* @return An index of the last occurrence of [char] or -1 if none is found.\n
The maximum number of substrings to return. Zero by default means no limit is set.

To avoid ambiguous results when strings in [delimiters] have characters in common, this method proceeds from $n @param *init* beginning to the end of this string, and finds at each position the first element in [delimiters]$n that matches this string at that position.

private fun CharSequence.rangesDelimitedBy(delimiters: Array<String>, startIndex: Int = 0, ignoreCase: Boolean = false, limit: Int = 0): Sequence<IntRange> {
    require(limit >= 0, { "Limit must be non-negative, but was $limit." })
    val delimitersList = delimiters.asList()

    return DelimitedRangesSequence(this, startIndex, limit, { currentIndex -> findAnyOf(delimitersList, currentIndex, ignoreCase = ignoreCase, last = false)?.let { it.first to it.second.length } })
}

// split

/**
 * Splits this char sequence to a sequence of strings around occurrences of the specified [delimiters].
 *
 * @param delimiters One or more strings to be used as delimiters.
 * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.
 * @param limit The maximum number of substrings to return. Zero by default means no limit is set.
 * To avoid ambiguous results when strings in [delimiters] have characters in common, this method proceeds from $n @param *init* beginning to the end of this string, and matches at each position the first element in [delimiters]$n that matches this string at that position.
 */
public fun CharSequence.splitToSequence(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): Sequence<String> =
    rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it) }

/**
 * Splits this char sequence to a list of strings around occurrences of the specified [delimiters].
 *
 * @param delimiters One or more strings to be used as delimiters.
 * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.
 * @param limit The maximum number of substrings to return. Zero by default means no limit is set.
 * To avoid ambiguous results when strings in [delimiters] have characters in common, this method proceeds from $n @param *init* beginning to the end of this string, and matches at each position the first element in [delimiters]$n that matches this string at that position.
 */
public fun CharSequence.split(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): List<String> {
    if (delimiters.size == 1) {
        val delimiter = delimiters[0]
        if (!delimiter.isEmpty()) {
            return split(delimiter, ignoreCase, limit)
        }
    }

    return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }
}

/**
 * Splits this char sequence to a sequence of strings around occurrences of the specified [delimiter].
 * This is specialized version of split which receives single non-empty delimiter and offers better performance.
 *
 * @param delimiter String used as delimiter
 * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.
 * @param limit The maximum number of substrings to return.
 */
private fun CharSequence.split(delimiter: String, ignoreCase: Boolean, limit: Int): List<String> {
    require(limit >= 0, { "Limit must be non-negative, but was $limit." })

    var currentOffset = 0
    var nextIndex = indexOf(delimiter, currentOffset, ignoreCase)
    if (nextIndex == -1 || limit == 1) {
        return listOf(this.toString())
    }

    val isLimited = limit > 0
    val result = ArrayList<String>(if (isLimited) limit.coerceAtMost(10) else 10)
    do {
        result.add(substring(currentOffset, nextIndex))
        currentOffset = nextIndex + delimiter.length
    } while (isLimited && result.size == limit - 1)
result.add(substring(currentOffset, length))
return result

// Splits this char sequence around matches of the given regular expression.
\nvar \n*param \nlimit \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @param limit \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @param limit \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @param \nlimit \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.
\n\n* @throws \nIndexOutOfBoundsException if either [offset] or [length] are less than zero or the [start] and [end] indexes overlap.
\n\n* @param \noffset \nNon-negative value specifying the maximum number of substrings to return.
\n* Zero by default means no limit is set.


the bytes in UTF-8 encoding in this array or its subrange. *param start-index the beginning (inclusive) of the subrange to decode, 0 by default. *param end-index the end (exclusive) of the subrange to decode, size of this array by default. *param throw-on-invalid-sequence specifies whether to throw an exception on malformed byte sequence or replace it by the replacement char \uFFFD. *param throws IndexOfBoundsException if [start-index] is less than zero or [end-index] is greater than the size of this array. *param throws IllegalArgumentException if [start-index] is greater than [end-index]. *param throws CharacterCodingException if the byte array contains malformed UTF-8 byte sequence and [throw-on-invalid-sequence] is true. *param throws IndexOfBoundsException if [start-index] is greater than [end-index]. *param throws CharacterCodingException if the byte array contains malformed UTF-8 byte sequence and [throw-on-invalid-sequence] is true.

AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size) return decodeUtf8(this, startIndex, endIndex, throwOnInvalidSequence) Encodes this string to an array of bytes in UTF-8 encoding.

*param startIndex the beginning (inclusive) of the subrange to decode, 0 by default.
*param endIndex the end (exclusive) of the subrange to decode, size of this array by default.
*param throwOnInvalidSequence specifies whether to throw an exception on malformed byte sequence or replace it by the replacement char \uFFFD. *param throws IndexOfBoundsException if [start-index] is less than zero or [end-index] is greater than the size of this array. *param throws IllegalArgumentException if [start-index] is greater than [end-index]. *param throws CharacterCodingException if the byte array contains malformed UTF-8 byte sequence and [throw-on-invalid-sequence] is true.

String.encodeToByteArray(): ByteArray { return encodeUtf8(this, 0, length, false) } Encodes this string or its substring to an array of bytes in UTF-8 encoding. *param startIndex the beginning (inclusive) of the substring to encode, 0 by default. *param endIndex the end (exclusive) of the substring to encode, length of this string by default. *param throw-on-invalid-sequence specifies whether to throw an exception on malformed char sequence or replace. *param throws IndexOfBoundsException if [start-index] is less than zero or [end-index] is greater than the length of this string. *param throws IllegalArgumentException if [startIndex] is greater than [endIndex]. *param throws CharacterCodingException if this string contains malformed char sequence and [throw-on-invalid-sequence] is true.

AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size) return encodeUtf8(this, startIndex, endIndex, throwOnInvalidSequence) Returns a copy of this string converted to upper case using the rules of the default locale. *param sample samples.text.Strings.toUpperCase *param @kotlin.internal.InlineOnly String.toUpperCase(): String = asDynamic().toUpperCase() Returns a copy of this string converted to lower case using the rules of the default locale. *param sample samples.text.Strings.toLowerCase *param @kotlin.internal.InlineOnly String.toLowerCase(): String = asDynamic().toLowerCase() String.nativeIndexOf(str: String, fromIndex: Int): Int = asDynamic().indexOf(str, fromIndex) String.nativeLastIndexOf(str: String, fromIndex: Int): Int = asDynamic().lastIndexOf(str, fromIndex) String.nativeStartsWith(s: String, position: Int): Boolean = asDynamic().startsWith(s, position) String.nativeEndsWith(s: String): Boolean = asDynamic().endsWith(s) String.substring(startIndex: Int): String = asDynamic().substring(startIndex) String.substring(startIndex: Int, endIndex: Int): String = asDynamic().substring(startIndex, endIndex) String.nativeUpperCase(str: String, fromIndex: Int): String = asDynamic().toUpperCase(str, fromIndex) String.nativeLowerCase(str: String, fromIndex: Int): String = asDynamic().toLowerCase(str, fromIndex) String.concat(str: String): String = asDynamic().concat(str) String.match(regex: String): Array<String>? = asDynamic().match(regex) String.replace(toImplement effective trimLeading and trimTrailing: String, ignoreCase: Boolean = false): Int { if (ignoreCase) { val n1 = this.length val n2 = other.length val min = Math.min(n1, n2) if (min == 0) return n1 - n2

Open Source Used In DNAC Platform Antelope 13928

var start = 0
while (true) {
    var end = minOf(start + 16, min)
    var s1 = this.substring(start, end)
    var s2 = other.substring(start, end)
    if (s1 != s2) {
        s1 = s1.toUpperCase()
        s2 = s2.toUpperCase()
        if (s1 != s2) {
            s1 = s1.toLowerCase()
            s2 = s2.toLowerCase()
            if (s1 != s2) {
                return s1.compareTo(s2)
            }
        }
    }
    if (end == min) break
    start = end
}
return n1 - n2
}

private val STRING_CASE_INSENSITIVE_ORDER = Comparator<String> { a, b -> a.compareTo(b, ignoreCase = true) }

@SinceKotlin("1.2")
public actual val String.Companion.CASE_INSENSITIVE_ORDER: Comparator<String> get() = STRING_CASE_INSENSITIVE_ORDER

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * DO NOT EDIT!\n
package kotlin

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UByteArray(@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS") @PublishedApi internal constructor(@PublishedApi val storage: ByteArray): Collection<UByte> { /** Creates a new array of the specified [size], with all elements initialized to zero. */ public constructor(size: Int): this(ByteArray(size)) /* * Returns the array element at the given [index]. This method can be called using the index operator. * * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS */
    operator fun get(index: Int): UByte = storage[index].toUByte() /* * Sets the element at the given [index] to the given [value]. This method can be called using the index operator. * * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS */
    operator fun set(index: Int, value: UByte) {
        storage[index] = value.toByte()
    }
    operator fun iterator(): UByteIterator = Iterator(storage)
    override fun contains(element: UByte): Boolean {
        // TODO: Eliminate this check after KT-30016 gets fixed.
        // Currently JS BE does not generate special bridge method for this method.
        return false
    }
    override fun containsAll(elements: Collection<UByte>): Boolean {
        return (elements as Collection<*>).all { it is UByte && it in storage }
    }
    override fun isEmpty(): Boolean = this.storage.isEmpty()
    override fun size: Int get() = this.storage.size
    /** Creates an iterator over the elements of the array. */
    public override operator fun iterator(): IntArray<UInt> = this(IntArray(storage.size)) /* * Returns the number of elements in the array. */
    public override val size: Int get() = storage.size
    /** Creates an iterator over the elements of the array. */
    public override operator fun iterator(): UByteIterator = this.UByteIterator() /* private var index = 0 */
    override fun hasNext() = index < storage.size
    override fun nextUByte() = if (index < storage.size)
        storage[index++].toUByte() else throw NoSuchElementException(index.toString())
    
    /** Creates a new array of the specified [size], where each element is calculated by calling the specified [init] function. * * The function [init] is called for each array element sequentially starting from the first one. */
    fun UByteArray(size: Int, init: (Int) -> UByte): UByteArray {
        return UByteArray(ByteArray(size) { index -> init(index).toByte() })
    }
    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    fun UByteArray(size: Int): UByteArray {
        return UByteArray(ByteArray(size))
    }
    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    fun UByteArray(size: Int, init: (Int) -> UByte): UByteArray {
        return UByteArray(ByteArray(size) { index -> init(index).toByte() })
    }
}*/

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * DO NOT EDIT!\n
package kotlin

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline class UIntArray(@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS") @PublishedApi internal constructor(@PublishedApi val storage: IntArray): Collection<UInt> { /** Creates a new array of the specified [size], where each element is calculated by calling the specified [init] function. * * The function [init] is called for each array element sequentially starting from the first one. */
    operator fun get(index: Int): UInt = this.storage[index]
    operator fun set(index: Int, value: UInt) {
        this.storage[index] = value
    }
    override fun contains(element: UInt): Boolean {
        // TODO: Eliminate this check after KT-30016 gets fixed.
        // Currently JS BE does not generate special bridge method for this method.
        return false
    }
    override fun containsAll(elements: Collection<UInt>): Boolean {
        return (elements as Collection<*>).all { it is UInt && it in storage }
    }
    override fun isEmpty(): Boolean = this.storage.isEmpty()
    override fun size: Int get() = this.storage.size
    /** Creates an iterator over the elements of the array. */
    public override operator fun iterator(): IntArray<UInt> = this(IntArray(storage.size)) /* * Returns the number of elements in the array. */
    public override val size: Int get() = storage.size
    /** Creates an iterator over the elements of the array. */
    public override operator fun iterator(): UIntIterator = this.UIntIterator() /* private var index = 0 */
    override fun hasNext() = index < storage.size
    override fun nextUInt() = if (index < storage.size)
        storage[index++].toUInt() else throw NoSuchElementException(index.toString())
    
    /** Creates a new array of the specified [size], where each element is calculated by calling the specified [init] function. * * The function [init] is called for each array element sequentially starting from the first one. */
    fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
        return UIntArray(ByteArray(size) { index -> init(index).toByte() })
    }
    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    fun UIntArray(size: Int): UIntArray {
        return UIntArray(ByteArray(size))
    }
    /** Creates a new array of the specified [size], with all elements initialized to zero. */
    fun UIntArray(size: Int, init: (Int) -> UInt): UIntArray {
        return UIntArray(ByteArray(size) { index -> init(index).toByte() })
    }
}*/
[IndexOutOfBoundsException] except in Kotlin/JS
* where the behavior is unspecified.
*\n*\n* public

*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
*\n
should return the value for an array element given its index.

/*
 * Copyright 2010-2020 JetBrains s.r.o.
 * Use of this source code is governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file
 */

// Auto-generated file. DO NOT EDIT!

```kotlin
package kotlin.collections

//\NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/blob/master/libraries/stdlib

import kotlin.random.*
import kotlin.ranges.contains
import kotlin.ranges.reversed

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an 
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.component1(): UInt {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an 
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UShortArray.component1(): UShort {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an 
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UCharArray.component1(): UChar {
    return get(0)
}

/**
 * Returns 1st *element* from the array.
 * 
 * If the size of this array is less than 1, throws an 
 * [IndexOutOfBoundsException] except in Kotlin/JS
 * where the behavior is unspecified.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component1(): UByte {
    return get(0)
}

/*
 * Copyright 2010-2020 JetBrains s.r.o.
 * Use of this source code is governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file
 */

// Auto-generated file. DO NOT EDIT!

```
* where the behavior is unspecified.

*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
ULongArray.component1(): ULong {\n    return get(0)\n}/**\n* Returns 1st *element* from the array.\n* * *\n* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UByteArray.component1(): UByte {\n    return get(0)\n}/**\n* Returns 1st *element* from the array.\n* * *\n* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UCharArray.component1(): UChar {\n    return get(0)\n}/**\n* Returns 1st *element* from the array.\n* * *\n* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UShortArray.component1(): UShort {\n    return get(0)\n}/**\n* Returns 1st *element* from the array.\n* * *\n* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UIntArray.component1(): UInt {\n    return get(0)\n}/**\n* Returns 1st *element* from the array.\n* * *\n* If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UShortArray.component2(): UShort {\n    return get(1)\n}/**\n* Returns 2nd *element* from the array.\n* * *\n* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UCharArray.component2(): UChar {\n    return get(1)\n}/**\n* Returns 2nd *element* from the array.\n* * *\n* If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UShortArray.component3(): UShort {\n    return get(2)\n}/**\n* Returns 3rd *element* from the array.\n* * *\n* If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UCharArray.component3(): UChar {\n    return get(2)\n}/**\n* Returns 3rd *element* from the array.\n* * *\n* If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
*An@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n\npublic inline operator fun
UShortArray.component4(): UShort {\n    return get(3)\n}/**\n* Returns 4th *element* from the array.\n* * *\n* If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n* where the behavior is unspecified.\n
Open Source Used In DNAC Platform Antelope 13932
is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.component4(): ULong { return get(3) }
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component4(): UByte { return get(3) }
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UShortArray.component4(): UShort { return get(3) }
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UIntArray.component5(): UInt { return get(4) }
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun ULongArray.component5(): ULong { return get(4) }
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UByteArray.component5(): UByte { return get(4) }
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline operator fun UShortArray.component5(): UShort { return get(4) }
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UIntArray.elementAt(index: Int): UInt
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun ULongArray.elementAt(index: Int): ULong
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UByteArray.elementAt(index: Int): UByte
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UShortArray.elementAt(index: Int): UShort
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

is unspecified.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}
```

is unspecified.
result of calling the [defaultValue] function if the [index] is out of bounds of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.
 * 
 * @sample
 * samples.collections.Collections.Elements.elementAtOrNull
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.elementAtOrNull(index: Int): UInt? {
    return this.getOrNull(index)
}

/**
 * Returns the first element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample
 * samples.collections.Collections.Elements.find
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.find(predicate: (UByte) -> Boolean): UByte? {
    return firstOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * 
 * @sample
 * samples.collections.Collections.Elements.find
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {
    return lastOrNull(predicate)
}

result of calling the [defaultValue] function if the [index] is out of bounds of this array.

```
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun ULongArray.findLast(predicate: (ULong) -> Boolean): ULong? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * @sample samples.collections.Collections.Elements.find
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun UByteArray.findLast(predicate: (UByte) -> Boolean): UByte? {
    return lastOrNull(predicate)
}

/**
 * Returns the last element matching the given [predicate], or `null` if no such element was found.
 * @sample samples.collections.Collections.Elements.find
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {
    return lastOrNull(predicate)
}

/**
 * Returns first element.
 * @throws NoSuchElementException if the array is empty.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun UIntArray.first(): UInt {
    return storage.first().toUInt()
}

/**
 * Returns first element.
 * @throws NoSuchElementException if the array is empty.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun ULongArray.first(): ULong {
    return storage.first().toULong()
}

/**
 * Returns first element.
 * @throws NoSuchElementException if the array is empty.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun UByteArray.first(): UByte {
    return storage.first().toUByte()
}

/**
 * Returns first element.
 * @throws NoSuchElementException if the array is empty.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun UShortArray.first(): UShort {
    return storage.first().toUShort()
}

/**
 * Returns the first element matching the given [predicate].
 * @throws NoSuchElementException if no such element is found.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun UIntArray.first(predicate: (UInt) -> Boolean): UInt {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws NoSuchElementException if no such element is found.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun ULongArray.first(predicate: (ULong) -> Boolean): ULong {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws NoSuchElementException if no such element is found.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun UByteArray.first(predicate: (UByte) -> Boolean): UByte {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element matching the given [predicate].
 * @throws NoSuchElementException if no such element is found.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public inline fun UShortArray.first(predicate: (UShort) -> Boolean): UShort {
    for (element in this) if (predicate(element)) return element
    throw NoSuchElementException("Array contains no element matching the predicate.")
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public fun UIntArray.firstOrNull(): UInt? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public fun ULongArray.firstOrNull(): ULong? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public fun UByteArray.firstOrNull(): UByte? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element, or `null` if the array is empty.
 */
`@SinceKotlin("1.3")` `@ExperimentalUnsignedTypes` `@kotlin.internal.InlineOnly` `public fun UShortArray.firstOrNull(): UShort? {
    return if (isEmpty()) null else this[0]
}
fun UShortArray.firstOrNull(): UShort? {
    return if (isEmpty()) null else this[0]
}

/**
 * Returns the first element matching the given [predicate], or `null` if no
 * element was not found.
 */
fun UIntArray.firstOrNull(predicate: (UInt) -> Boolean): UInt? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if no
 * element was not found.
 */
fun ULongArray.firstOrNull(predicate: (ULong) -> Boolean): ULong? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if no
 * element was not found.
 */
fun UByteArray.firstOrNull(predicate: (UByte) -> Boolean): UByte? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns the first element matching the given [predicate], or `null` if no
 * element was not found.
 */
fun UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {
    for (element in this) if (predicate(element)) return element
    return null
}

/**
 * Returns an element at the given [index] or the result of calling the
 * [defaultValue] function if the [index] is out of bounds of this array.
 */
fun UIntArray.getOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of calling the
 * [defaultValue] function if the [index] is out of bounds of this array.
 */
fun ULongArray.getOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of calling the
 * [defaultValue] function if the [index] is out of bounds of this array.
 */
fun UByteArray.getOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns an element at the given [index] or the result of calling the
 * [defaultValue] function if the [index] is out of bounds of this array.
 */
fun UShortArray.getOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {
    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)
}

/**
 * Returns first index of
**Imported from Antelope 13937**

```kotlin
// Antelope 13937

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.indexOf(element: UInt): Int {
    return storage.indexOf(element.toInt())
}

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.indexOf(element: ULong): Int {
    return storage.indexOf(element.toLong())
}

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.indexOf(element: UByte): Int {
    return storage.indexOf(element.toByte())
}

/**
 * Returns first index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.indexOf(element: UShort): Int {
    return storage.indexOf(element.toShort())
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUInt()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toULong()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUByte()) }
}

/**
 * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.indexOfFirst(predicate: (UShort) -> Boolean): Int {
    return storage.indexOfFirst { predicate(it.toUShort()) }
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.last(): UInt {
    return storage.last().toUInt()
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.last(): ULong {
    return storage.last().toULong()
}
```

---

**Open Source Used In DNAC Platform Antelope 13937**
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.last(): UByte {
    return storage.last().toUByte()!
}

/**
 * Returns the last element.
 * @throws [NoSuchElementException] if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.last(): UShort {
    return storage.last().toUShort()!
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.last(predicate: (UByte) -> Boolean): UByte {
    for (index in this.indices.reversed()) {
        val element = this[index]!
        if (predicate(element)) return element!
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")!
}

/**
 * Returns the last element matching the given [predicate].
 * @throws [NoSuchElementException] if no such element is found.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.last(predicate: (UShort) -> Boolean): UShort {
    for (index in this.indices.reversed()) {
        val element = this[index]!
        if (predicate(element)) return element!
    }
    throw NoSuchElementException("Array contains no element matching the predicate.")!
}

/**
 * Returns last index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.lastIndexOf(element: UInt): Int {
    return storage.lastIndexOf(element.toInt())
}

/**
 * Returns last index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.lastIndexOf(element: ULong): Int {
    return storage.lastIndexOf(element.toLong())
}

/**
 * Returns last index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.lastIndexOf(element: UByte): Int {
    return storage.lastIndexOf(element.toByte())
}

/**
 * Returns last index of [element], or -1 if the array does not contain element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.lastIndexOf(element: UShort): Int {
    return storage.lastIndexOf(element.toShort())
}

/**
 * Returns the last element, or `null` if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.lastOrNull(): UInt? {
    return if (isEmpty()) null else this[size - 1]!
}

/**
 * Returns the last element, or `null` if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.lastOrNull(): ULong? {
    return if (isEmpty()) null else this[size - 1]!
}

/**
 * Returns the last element, or `null` if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.lastOrNull(): UByte? {
    return if (isEmpty()) null else this[size - 1]!
}

/**
 * Returns the last element, or `null` if the array is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.lastOrNull(): UShort? {
    return if (isEmpty()) null else this[size - 1]!
}
UIntArray.lastOrNull(predicate: (UInt) -> Boolean): UInt? { n for (index in this.indices.reversed()) { n val element = this[index] n if (predicate(element)) return element n } return null} println("* Returns the last element matching the given [predicate], or ‘null’ if no such element was found.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun ULongArray.lastOrNull(predicate: (ULong) -> Boolean): ULong? { n for (index in this.indices.reversed()) { n val element = this[index] n if (predicate(element)) return element n } return null} println("* Returns the last element matching the given [predicate], or ‘null’ if no such element was found.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UByteArray.lastOrNull(predicate: (UByte) -> Boolean): UByte? { n for (index in this.indices.reversed()) { n val element = this[index] n if (predicate(element)) return element n } return null} println("* Returns the last element matching the given [predicate], or ‘null’ if no such element was found.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UShortArray.lastOrNull(predicate: (UShort) -> Boolean): UShort? { n for (index in this.indices.reversed()) { n val element = this[index] n if (predicate(element)) return element n } return null} println("* Returns the last element matching the given [predicate], or ‘null’ if no such element was found.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UIntArray.random(): UInt { n return random(Random) println("* Returns a random element from this array.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun ULongArray.random(): ULong { n return random(Random) println("* Returns a random element from this array.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UByteArray.random(): UByte { n return random(Random) println("* Returns a random element from this array.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UShortArray.random(): UShort { n return random(Random) println("* Returns a random element from this array.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UIntArray.randomOrNull(): UInt? { n return randomOrNull(Random) println("* Returns a random element from this array, or ‘null’ if this array is empty.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun ULongArray.randomOrNull(): ULong? { n return randomOrNull(Random) println("* Returns a random element from this array, or ‘null’ if this array is empty.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UByteArray.randomOrNull(): UByte? { n return randomOrNull(Random) println("* Returns a random element from this array, or ‘null’ if this array is empty.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UShortArray.randomOrNull(): UShort? { n return randomOrNull(Random) println("* Returns a random element from this array, or ‘null’ if this array is empty.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UIntArray.random(random: Random): UInt { n if (isEmpty()) throw NoSuchElementException("Array is empty.") println("* Returns a random element from this array using the specified source of randomness.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public fun ULongArray.random(random: Random): ULong { n if (isEmpty()) throw NoSuchElementException("Array is empty.") println("* Returns a random element from this array using the specified source of randomness.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public fun UByteArray.random(random: Random): UByte { n if (isEmpty()) throw NoSuchElementException("Array is empty.") println("* Returns a random element from this array using the specified source of randomness.

* @SinceKotlin("1.3") * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public fun UShortArray.random(random: Random): UShort { n if (isEmpty()) throw NoSuchElementException("Array is empty.") println("* Returns a random element from this array using the specified source of randomness.

* @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UIntArray.randomOrNull(random: Random): UInt? { n return randomNextInt(size) println("* Returns a random element from this array using the specified source of randomness.

* @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun ULongArray.randomOrNull(random: Random): ULong? { n return randomNextInt(size) println("* Returns a random element from this array using the specified source of randomness.

* @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UByteArray.randomOrNull(random: Random): UByte? { n return randomNextInt(size) println("* Returns a random element from this array using the specified source of randomness.

* @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public inline fun UShortArray.randomOrNull(random: Random): UShort? { n return randomNextInt(size) println("* Returns a random element from this array using the specified source of randomness.

* @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public fun UIntArray.randomOrNull(random: Random): UInt? { n return randomOrNotNull(random) println("* Returns a random element from this array, or ‘null’ if this array is empty.

* @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public fun ULongArray.randomOrNull(random: Random): ULong? { n return randomOrNotNull(random) println("* Returns a random element from this array, or ‘null’ if this array is empty.

* @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public fun UByteArray.randomOrNull(random: Random): UByte? { n return randomOrNotNull(random) println("* Returns a random element from this array, or ‘null’ if this array is empty.

* @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes

@kotlin.internal.InlineOnly
@public fun UShortArray.randomOrNull(random: Random): UShort? { n return randomOrNotNull(random) println("* Returns a random element from this array, or ‘null’ if this array is empty.
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public inline fun UByteArray.randomOrNull(): UByte? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array, or `null` if this array is empty.
*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public inline fun UShortArray.randomOrNull(): UShort? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public fun UIntArray.randomOrNull(random: Random): UInt? {
    if (isEmpty())
        return null
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public fun ULongArray.randomOrNull(random: Random): ULong? {
    if (isEmpty())
        return null
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public fun UByteArray.randomOrNull(random: Random): UByte? {
    if (isEmpty())
        return null
    return get(random.nextInt(size))
}

/**
 * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.
*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public fun UShortArray.randomOrNull(random: Random): UShort? {
    if (isEmpty())
        return null
    return get(random.nextInt(size))
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.single(): UInt {
    return storage.single().toUInt()
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.single(): ULong {
    return storage.single().toULong()
}

/**
 * Returns the single element, or throws an exception if the array is empty or has more than one element.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.single(): UByte {
    return storage.single().toUByte()
}

/**
 * Returns the single element, or throws an exception if there is no or more than one matching element.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.single(predicate: (UShort) -> Boolean): UShort {
    var single: UShort? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as UShort
}

/**
 * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.single(predicate: (UInt) -> Boolean): UInt {
    var single: UInt? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        } else if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    }
    if (!found) throw NoSuchElementException("Array contains no element matching the predicate.")
    @Suppress("UNCHECKED_CAST")
    return single as UInt
}
matching the given [predicate], or throws exception if there is no or more than one matching element.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.single(predicate: (UByte) -> Boolean): UByte {
    var single: UByte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no matching element matching the predicate.")
    return single
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.single(predicate: (UShort) -> Boolean): UShort {
    var single: UShort? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) throw IllegalArgumentException("Array contains more than one matching element.")
            single = element
            found = true
        }
    }
    if (!found) throw NoSuchElementException("Array contains no matching element matching the predicate.")
    return single
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UIntArray.singleOrNull(): UInt? {
    return if (size == 1) this[0] else null
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun ULongArray.singleOrNull(): ULong? {
    return if (size == 1) this[0] else null
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UByteArray.singleOrNull(): UByte? {
    return if (size == 1) this[0] else null
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UShortArray.singleOrNull(): UShort? {
    return if (size == 1) this[0] else null
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {
    var single: UInt? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun ULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {
    var single: ULong? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {
    var single: UByte? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public fun UShortArray.singleOrNull(predicate: (UShort) -> Boolean): UShort? {
    var single: UShort? = null
    var found = false
    for (element in this) {
        if (predicate(element)) {
            if (found) return null
            single = element
            found = true
        }
    }
    if (!found) return null
    return single
}
```

n).coerceAtLeast(0))

* Returns a list containing all elements except first [n] elements.

**Throws**:
IllegalArgumentException if [n] is negative.

*sample
collections.Collections.Transformations.drop

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.drop(n: Int): List<ULong> {

    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))

    * Returns a list containing all elements except first [n] elements.

**Throws**:
IllegalArgumentException if [n] is negative.

* sample
collections.Collections.Transformations.drop

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.drop(n: Int): List<UByte> {

    require(n >= 0) { "Requested element count $n is less than zero." }
    return takeLast((size - n).coerceAtLeast(0))

    * Returns a list containing all elements except last [n] elements.

**Throws**:
IllegalArgumentException if [n] is negative.

* sample
collections.Collections.Transformations.drop

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.dropLast(n: Int): List<UInt> {

    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))

    * Returns a list containing all elements except last [n] elements.

**Throws**:
IllegalArgumentException if [n] is negative.

* sample
collections.Collections.Transformations.drop

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.dropLast(n: Int): List<ULong> {

    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))

    * Returns a list containing all elements except last [n] elements.

**Throws**:
IllegalArgumentException if [n] is negative.

* sample
collections.Collections.Transformations.drop

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.dropLast(n: Int): List<UByte> {

    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))

    * Returns a list containing all elements except last [n] elements.

**Throws**:
IllegalArgumentException if [n] is negative.

* sample
collections.Collections.Transformations.drop

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.dropLast(n: Int): List<UShort> {

    require(n >= 0) { "Requested element count $n is less than zero." }
    return take((size - n).coerceAtLeast(0))

    * Returns a list containing all elements except last [n] elements.

**Throws**:
IllegalArgumentException if [n] is negative.

* sample
collections.Collections.Transformations.drop

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> {

    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> {

    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {

    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {

    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> {

    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {

    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()

* @SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {

    for (index in lastIndex downTo 0) {
        if (!predicate(this[index])) {
            return take(index + 1)
        }
    }
    return emptyList()
UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index]))
            return take(index + 1)
    return emptyList()
} * Returns a list containing all elements except last elements that satisfy the given [predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index]))
            return take(index + 1)
    return emptyList()
} * Returns a list containing all elements except last elements that satisfy the given
[predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index]))
            return take(index + 1)
    return emptyList()
} * Returns a list containing all elements except last elements that satisfy the given
[predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> {
    var yielding = false
    val list = ArrayList<UInt>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
} * Returns a list containing all elements except first elements that satisfy the given
[predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.dropWhile(predicate: (ULong) -> Boolean): List<ULong> {
    var yielding = false
    val list = ArrayList<ULong>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
} * Returns a list containing all elements except first elements that satisfy the given
[predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {
    var yielding = false
    val list = ArrayList<UByte>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
} * Returns a list containing all elements except first elements that satisfy the given
[predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {
    var yielding = false
    val list = ArrayList<UShort>()
    for (item in this)
        if (yielding)
            list.add(item)
        else if (!predicate(item)) {
            list.add(item)
            yielding = true
        }
    return list
} * Returns a list containing all elements except first elements that satisfy the given
[predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UIntArray.filter(predicate: (UInt) -> Boolean): List<UInt> {
    return filterTo(ArrayList<UInt>(), predicate)
} * Returns a list containing only elements matching the given [predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {
    return filterTo(ArrayList<ULong>(), predicate)
} * Returns a list containing only elements matching the given [predicate].

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {
    return filterTo(ArrayList<UByte>(), predicate)
} * Returns a list containing only elements matching the given [predicate].

@SinceKotlin("1.3")
@ ExperimentalUnsignedTypes
public inline fun UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {
    return filterTo(ArrayList<UShort>(), predicate)
} * Returns a list containing only elements matching the given [predicate].
evaluation on the element.

```kotlin
@sample samples.collections.Collections.Filtering.filterIndexed

@sample samples.collections.Collections.Filtering.filterIndexedTo

@sample samples.collections.Collections.Filtering.filterIndexedTo

@sample samples.collections.Collections.Filtering.filterIndexedTo

@sample samples.collections.Collections.Filtering.filterIndexedTo

@sample samples.collections.Collections.Filtering.filterIndexedTo

@sample samples.collections.Collections.Filtering.filterIndexedTo

@sample samples.collections.Collections.Filtering.filterIndexedTo
```

```

```
destination.add(element)
}  // Returns a list containing all elements not matching the given [predicate].

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.filterNot(predicate: (UInt) -> Boolean): List<UInt> {
    var destination = ArrayList<UInt>()
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.filterNot(predicate: (ULong) -> Boolean): List<ULong> {
    var destination = ArrayList<ULong>()
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.filterNot(predicate: (UByte) -> Boolean): List<UByte> {
    var destination = ArrayList<UByte>()
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Returns a list containing all elements not matching the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.filterNot(predicate: (UShort) -> Boolean): List<UShort> {
    var destination = ArrayList<UShort>()
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <C : MutableCollection<in UInt>> UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C {
    var destination = ArrayList<UInt>()
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <C : MutableCollection<in ULong>> ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C {
    var destination = ArrayList<ULong>()
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <C : MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {
    var destination = ArrayList<UByte>()
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun <C : MutableCollection<in UShort>> UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C {
    var destination = ArrayList<UShort>()
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UIntArray.filterNotTo(destination: MutableCollection<inUInt>, predicate: (UInt) -> Boolean): MutableCollection<inUInt> {
    var destination = destination.arrayList
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun ULongArray.filterNotTo(destination: MutableCollection<in ULong>, predicate: (ULong) -> Boolean): MutableCollection<in ULong> {
    var destination = destination.arrayList
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UByteArray.filterNotTo(destination: MutableCollection<in UByte>, predicate: (UByte) -> Boolean): MutableCollection<in UByte> {
    var destination = destination.arrayList
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}

/**
 * Appends all elements not matching the given [predicate] to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
@public inline fun UShortArray.filterNotTo(destination: MutableCollection<in UShort>, predicate: (UShort) -> Boolean): MutableCollection<in UShort> {
    var destination = destination.arrayList
    for (element in this) if (!predicate(element)) destination.add(element)
    return destination
}
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun <UByte> UByteArray.filterTo(destination: Collection<UByte>, predicate: (UByte) -> Boolean): Collection<UByte> {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun <UShort> UShortArray.filterTo(destination: Collection<UShort>, predicate: (UShort) -> Boolean): Collection<UShort> {
    for (element in this) if (predicate(element)) destination.add(element)
    return destination
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun List<UInt> slice(indices: IntRange): List<UInt> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun List<ULong> slice(indices: IntRange): List<ULong> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun List<UByte> slice(indices: IntRange): List<UByte> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun List<UShort> slice(indices: IntRange): List<UShort> {
    if (indices.isEmpty()) return listOf()
    return copyOfRange(indices.start, indices.endInclusive + 1).asList()
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun List<UInt> slice(indices: Collection<Int>): List<UInt> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<UInt>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun List<ULong> slice(indices: Collection<Int>): List<ULong> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<ULong>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun List<UByte> slice(indices: Collection<Int>): List<UByte> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<UByte>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun List<UShort> slice(indices: Collection<Int>): List<UShort> {
    val size = indices.collectionSizeOrDefault(10)
    if (size == 0) return emptyList()
    val list = ArrayList<UShort>(size)
    for (index in indices) {
        list.add(get(index))
    }
    return list
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sliceArray(indices: Collection<Int>): UIntArray {
    return UIntArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sliceArray(indices: Collection<Int>): ULongArray {
    return ULongArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sliceArray(indices: Collection<Int>): UByteArray {
    return UByteArray(storage.sliceArray(indices))
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sliceArray(indices: Collection<Int>): UShortArray {
    return UShortArray(storage.sliceArray(indices))
}
Returns a list containing last `[n]` elements.

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.takeLast(n: Int): List<UByte> {
    require(n >= 0) { "Requested element count Sn is less than zero." }
    if (n == 0) return emptyList()
    val size = this.size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<UByte>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.takeLast(n: Int): List<UShort> {
    require(n >= 0) { "Requested element count Sn is less than zero." }
    if (n == 0) return emptyList()
    val size = this.size
    if (n >= size) return toList()
    if (n == 1) return listOf(this[size - 1])
    val list = ArrayList<UShort>(n)
    for (index in size - n until size)
        list.add(this[index])
    return list
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt> {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    return toList()
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    return toList()
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    return toList()
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {
    for (index in lastIndex downTo 0)
        if (!predicate(this[index])) {
            return drop(index + 1)
        }
    return toList()
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.takeWhile(predicate: (UInt) -> Boolean): List<UInt> {
    val list = ArrayList<UInt>()
    for (item in this)
        if (!predicate(item))
            break
        list.add(item)
    return list
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.takeWhile(predicate: (ULong) -> Boolean): List<ULong> {
    val list = ArrayList<ULong>()
    for (item in this)
        if (!predicate(item))
            break
        list.add(item)
    return list
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.takeWhile(predicate: (UByte) -> Boolean): List<UByte> {
    val list = ArrayList<UByte>()
    for (item in this)
        if (!predicate(item))
            break
        list.add(item)
    return list
}
```

```kotlin
@Sample
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {
    val list = ArrayList<UShort>()
    for (item in this)
        if (!predicate(item))
            break
        list.add(item)
    return list
}
```
samples.collections.Collections.Transformations.take
*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {
    var list = ArrayList<UShort>()
    for (item in this) {
        if (!predicate(item))
            break
        list.add(item)
    }
    return list
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reverse(): Unit {
    storage.reverse()
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reverse(): Unit {
    storage.reverse()
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reverse(): Unit {
    storage.reverse()
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reverse(): Unit {
    storage.reverse()
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {
    storage.reverse(fromIndex, toIndex)
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reversed(): List<UInt> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reversed(): List<ULong> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reversed(): List<UByte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reversed(): List<UShort> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reverseOrder(): List<UInt> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reverseOrder(): List<ULong> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reverseOrder(): List<UByte> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}

@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reverseOrder(): List<UShort> {
    if (isEmpty()) return emptyList()
    val list = toMutableList()
    list.reverse()
    return list
}
**UIntArray.reversedArray():**

```kotlin
UIntArray.reversedArray(): UIntArray {
    return UIntArray(storage.reversedArray())
}
```

**Returns an array with elements of this array in reversed order.**

**UIntArray.shuffle():**

```kotlin
UIntArray.shuffle(): Unit {
    shuffle(Random)
}
```

**Randomly shuffles elements in this array in-place.**

**UIntArray.sortDescending():**

```kotlin
UIntArray.sortDescending(): Unit {
    if (size > 1) {
        sort()
        reverse()
    }
}
```

**Sorts elements in the array in-place descending according to their natural sort order.**

---

**Open Source Used In DNAC Platform Antelope 13950**
if (size > 1) {
    sort()
    reverse()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun UIntArray.sorted(): List<UInt> {
    return copyOf().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun ULongArray.sorted(): List<ULong> {
    return copyOf().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun UByteArray.sorted(): List<UByte> {
    return copyOf().apply { sort() }.asList()
}

/**
 * Returns a list of all elements sorted according to their natural sort order.
 */
public fun UShortArray.sorted(): List<UShort> {
    return copyOf().apply { sort() }.asList()
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun UIntArray.sortedArray(): UIntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun ULongArray.sortedArray(): ULongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun UByteArray.sortedArray(): UByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted according to their natural sort order.
 */
public fun UShortArray.sortedArray(): UShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sort() }
}

/**
 * Returns an array with all elements of this array sorted descending according to their natural sort order.
 */
public fun UIntArray.sortedArrayDescending(): UIntArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted descending according to their natural sort order.
 */
public fun ULongArray.sortedArrayDescending(): ULongArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted descending according to their natural sort order.
 */
public fun UByteArray.sortedArrayDescending(): UByteArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns an array with all elements of this array sorted descending according to their natural sort order.
 */
public fun UShortArray.sortedArrayDescending(): UShortArray {
    if (isEmpty()) return this
    return this.copyOf().apply { sortDescending() }
}

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun UIntArray.sortedDescending(): List<UInt> {
    return copyOf().apply { sort() }.reversed()
}

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun ULongArray.sortedDescending(): List<ULong> {
    return copyOf().apply { sort() }.reversed()
}

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun UByteArray.sortedDescending(): List<UByte> {
    return copyOf().apply { sort() }.reversed()
}

/**
 * Returns a list of all elements sorted descending according to their natural sort order.
 * 
 * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.
 */
public fun UShortArray.sortedDescending(): List<UShort> {
    return copyOf().apply { sort() }.reversed()
}

Open Source Used In DNAC Platform Antelope 13951
Returns an array of type `[ByteArray]`, which is a view of this array where each element is a signed reinterpretation \* of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArrayList.asByteArray(): ByteArrayList { return storage }  
```

Returns an array of type `[IntArray]`, which is a view of this array where each element is a signed reinterpretation \* of the corresponding element of this array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArrayList.asIntArray(): IntArrayList { return storage }  
```

Returns a `List` that wraps the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UIntArrayList.asList(): List<UInt>  
```

Returns a `List` that wraps the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun ULongArrayList.asList(): List<ULong>  
```

Returns a `List` that wraps the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public expect fun UByteArrayArrayList.asList(): List<UByte>  
```

Returns a `List` that wraps the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArrayList.asList(): UShortArrayList { return storage }  
```

Returns `true` if the two specified arrays are *structurally* equal to one another, \* i.e. contain the same number of the same elements in the same order.

```kotlin
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun UIntArrayList.contentEquals(other: UIntArrayList): Boolean { return this.contentEquals(other) }  
```

Returns `true` if the two specified arrays are *structurally* equal to one another, \* i.e. contain the same number of the same elements in the same order.

```kotlin
@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning.")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun ULongArrayList.contentEquals(other: ULongArrayList): Boolean { return this.contentEquals(other) }  
```

Returns `true` if the two specified arrays are *structurally* equal to one another, \* i.e. contain the same number of the same elements in the same order.
Open Source Used In DNAC Platform Antelope  13953
samples.collections.Arrays.ContentOperations.contentToString
} /*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning."")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.contentToString(): String {
    return this.contentToString()\n} /*\n@Deprecated("Use Kotlin compiler 1.4 to avoid deprecation warning."")\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentToString(): String {
    return this.contentToString()\n} /*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentToString(): String {
    return this?.contentToString() ?: \"null\"
} /*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray?.contentToString(): String {
    return this?.contentToString() ?: \"null\"
} /*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentToString(): String {
    return this?.contentToString() ?: \"null\"
} /*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray?.contentToString(): String {
    return this?.contentToString() ?: \"null\"
} /*\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UIntArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
} /*\n@SinceKotlin("1.3")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.copystorage(copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UIntArray {
    storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)
    return destination
} */
destination range.

*param destination the array to copy to.

*param destinationOffset the position in the [destination] array to copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

*throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*returns IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset].

* or when that index is out of the [destination] array indices range.

*return the [destination] array.

*An@SinceKotlin("1.3.0")@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly

public inline fun ULongArray.copyInto(destination: ULongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):

ULongArray

return destination

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] array to copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

*throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*returns IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],

* or when that index is out of the [destination] array indices range.

*return the [destination] array.

*An@SinceKotlin("1.3.0")@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly

public inline fun UByteArray.copyInto(destination: UByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):

UByteArray

/**
 * Copies this array or its subrange into the [destination] array and returns that array.
 * 
 * It's allowed to pass the same array in the [destination] array to copy to, 0 by default.

*param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.

*param endIndex the end (exclusive) of the subrange to copy, size of this array by default.

*throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.

*returns IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],

* or when that index is out of the [destination] array indices range.

*return the [destination] array.

*An@SinceKotlin("1.3.0")@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly

public inline fun UShortArray.copyInto(destination: UShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size):

UShortArray

/**
 * Returns new array which is a copy of the original array.
 * 
 * @sample samples.collections.Arrays.CopyOfOperations.copyOf

*/

@SinceKotlin("1.3.0")

public inline fun UIntArray.copyOf(): UIntArray

/**
 * Returns new array which is a copy of the original array, resized to the given [newSize].
 * The copy is either truncated or padded at

*An@SinceKotlin("1.3.0")@ExperimentalUnsignedTypes@n@kotlin.internal.InlineOnly

public inline fun UShortArray.copyOf(): UShortArray

*/

@SinceKotlin("1.3.0")

public inline fun UByteArray.copyOf(): UByteArray

*/

@SinceKotlin("1.3.0")

public inline fun ULongArray.copyOf(): ULongArray

*/

@SinceKotlin("1.3.0")

public inline fun UShortArray.copyOf(): UShortArray

*/

@SinceKotlin("1.3.0")

public inline fun UByteArray.copyOf(): UByteArray

*/

@SinceKotlin("1.3.0")

public inline fun UIntArray.copyOf(): UIntArray

*/
the end with zero values if necessary.
* - If [newSize] is less than the size of the original array, the copy array
is truncated to the [newSize].
* - If [newSize] is greater than the size of the original array, the extra elements in the
copy array are filled with zero values.

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\*\@kotlin.internal.InlineOnly\*\@public inline fun
*\UIntArray.copyOf(newSize: Int): UIntArray {\n    return UIntArray(storage.copyOf(newSize))\n}\n*\*\n* - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\*\@kotlin.internal.InlineOnly\*\@public inline fun
*\ULongArray.copyOf(newSize: Int): ULongArray {\n    return ULongArray(storage.copyOf(newSize))\n}\n*\*\n* - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\*\@kotlin.internal.InlineOnly\*\@public inline fun
*\UByteArray.copyOf(newSize: Int): UByteArray {\n    return UByteArray(storage.copyOf(newSize))\n}\n*\*\n* - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\*\@kotlin.internal.InlineOnly\*\@public inline fun
*\UShortArray.copyOf(newSize: Int): UShortArray {\n    return UShortArray(storage.copyOf(newSize))\n}\n*\*\n* - If [newSize] is less than the size of the
original array, the copy array is truncated to the [newSize].
* - If [newSize] is greater than the size of the original
array, the extra elements in the copy array are filled with zero values.

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\*\@kotlin.internal.InlineOnly\*\@public inline fun
*\UIntArray.copyOfRange(fromIndex: Int, toIndex: Int): UIntArray {
    return UIntArray(storage.copyOfRange(fromIndex, toIndex))\n}\n* - If [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.
* - Returns a new array which is a copy of the specified range of the original
array.
* - Throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\*\@kotlin.internal.InlineOnly\*\@public inline fun
*\ULongArray.copyOfRange(fromIndex: Int, toIndex: Int): ULongArray {
    return ULongArray(storage.copyOfRange(fromIndex, toIndex))\n}\n* - If [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.
* - Returns a new array which is a copy of the specified range of the original
array.
* - Throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\*\@kotlin.internal.InlineOnly\*\@public inline fun
*\UByteArray.copyOfRange(fromIndex: Int, toIndex: Int): UByteArray {
    return UByteArray(storage.copyOfRange(fromIndex, toIndex))\n}\n* - If [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.
* - Returns a new array which is a copy of the specified range of the original
array.
* - Throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.

*\@SinceKotlin("1.3")\*\@ExperimentalUnsignedTypes\*\@kotlin.internal.InlineOnly\*\@public inline fun
*\UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {
    return UShortArray(storage.copyOfRange(fromIndex, toIndex))\n}\n* - If [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.
* - Returns a new array which is a copy of the specified range of the original
array.
* - Throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this
array.

Open Source Used In DNAC Platform Antelope 13956
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun
UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {
    return
UShortArray(storage.copyOfRange(fromIndex, toIndex))
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.fill(element: UInt, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toInt(), fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.fill(element: ULong, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toLong(), fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.fill(element: UByte, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toByte(), fromIndex, toIndex)
}

/**
 * Fills this array or its subrange with the specified [element] value.
 * @param fromIndex the start of the range (inclusive) to fill, 0 by default.
 * @param toIndex the end of the range (exclusive) to fill, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.fill(element: UShort, fromIndex: Int = 0, toIndex: Int = size): Unit {
    storage.fill(element.toShort(), fromIndex, toIndex)
}

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UIntArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val ULongArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UByteArray.indices: IntRange
    get() = storage.indices

/**
 * Returns the range of valid indices for the array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline val UShortArray.indices: IntRange
    get() = storage.indices

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
operator fun UIntArray.plus(element: UInt): UIntArray {
    return UIntArray(storage + element.toInt())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
operator fun ULongArray.plus(element: ULong): ULongArray {
    return ULongArray(storage + element.toLong())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
operator fun UByteArray.plus(element: UByte): UByteArray {
    return UByteArray(storage + element.toByte())
}

/**
 * Returns an array containing all elements of the original array and then the given [element].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline
operator fun UShortArray.plus(element: UShort): UShortArray {
    return UShortArray(storage + element.toShort())
}
operator fun ULongArray.plus(element: ULong): ULongArray {
    return ULongArray(storage + element.toLong())
}

/**
 * Returns an array containing all elements of the original array and then the given
 * element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline operator fun UByteArray.plus(element: UByte): UByteArray {
    return UByteArray(storage + element.toByte())
}

/**
 * Returns an array containing all elements of the original array and then the given
 * element.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
    return UShortArray(storage + element.toShort())
}

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UIntArray.plus(elements: Collection<UInt>): UIntArray {
    var index = size
    val result = storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toInt()
    return UIntArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun ULongArray.plus(elements: Collection<ULong>): ULongArray {
    var index = size
    val result = storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toLong()
    return ULongArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UByteArray.plus(elements: Collection<UByte>): UByteArray {
    var index = size
    val result = storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toByte()
    return UByteArray(result)
}

/**
 * Returns an array containing all elements of the original array and then all elements
 * of the given [elements] collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public operator fun UShortArray.plus(elements: Collection<UShort>): UShortArray {
    var index = size
    val result = storage.copyOf(size + elements.size)
    for (element in elements) result[index++] = element.toShort()
    return UShortArray(result)
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.sort(): Unit {
    if (size > 1) sortArray(this, 0, size)
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.sort(): Unit {
    if (size > 1) sortArray(this, 0, size)
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.sort(): Unit {
    if (size > 1) sortArray(this, 0, size)
}

/**
 * Sorts the array in-place.
 * 
 * @sample samples.collections.Arrays.Sorting.sortArray
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.sort(): Unit {
    if (size > 1) sortArray(this, 0, size)
}
sortArray(this, 0, size)

/**
 * Sorts a range in the array in-place.
 * @param fromIndex the start of the range (inclusive) to sort, 0 by default.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)
    sortArray(this, fromIndex, toIndex)
}

/**
 * Sorts elements of the array in the specified range in-place.
 * The elements are sorted according to their natural sort order.
 * @param fromIndex the start of the range (inclusive) to sort.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    sort(fromIndex, toIndex)
    reverse(fromIndex, toIndex)
}

/**
 * Sorts elements of the array in the specified range in-place.
 * The elements are sorted according to their natural sort order.
 * @param fromIndex the start of the range (inclusive) to sort.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    sort(fromIndex, toIndex)
    reverse(fromIndex, toIndex)
}

/**
 * Sorts elements of the array in the specified range in-place.
 * The elements are sorted according to their natural sort order.
 * @param fromIndex the start of the range (inclusive) to sort.
 * @param toIndex the end of the range (exclusive) to sort, size of this array by default.
 * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.
 * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
 * @sample samples.collections.Arrays.Sorting.sortRangeOfArray
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {
    sort(fromIndex, toIndex)
    reverse(fromIndex, toIndex)
}
* @param fromIndex the start of the range (inclusive) to sort.* @param toIndex the end of the range
* (exclusive) to sort.* @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
* is greater than the size of this array.* @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].
*
*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {
    sort(fromIndex, toIndex)
    reverse(fromIndex, toIndex)
}

/**
 * Returns an array of type [ByteArray], which is a copy of this array where each element is a signed reinterpretation
* of the corresponding element of this array.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.toByteArray(): ByteArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [IntArray], which is a copy of this array where each element is a signed reinterpretation
* of the corresponding element of this array.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.toIntArray(): IntArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [LongArray], which is a copy of this array where each element is a signed reinterpretation
* of the corresponding element of this array.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.toLongArray(): LongArray {
    return storage.copyOf()
}

/**
 * Returns an array of type [ShortArray], which is a copy of this array where each element is a signed reinterpretation
* of the corresponding element of this array.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.toShortArray(): ShortArray {
    return storage.copyOf()
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.toTypedArray(): Array<UInt>
{  
    return Array(size) { index -> this[index] }
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.toTypedArray(): Array<ULong> {
    return Array(size) { index -> this[index] }
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ByteArray.toUByteArray(): UByteArray {
    return UByteArray(this.copyOf())
}

/**
 * Returns an array of type [UByteArray], which is a copy of this array where each element is an unsigned
* reinterpretation
* of the corresponding element of this array.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun IntArray.toUIntArray(): UIntArray {
    return UIntArray(this.copyOf())
}

/**
 * Returns an array of type [ULongArray], which is a copy of this array where each element is an unsigned
* reinterpretation
* of the corresponding element of this array.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.toUByteArray(): UByteArray {
    return UByteArray(this.copyOf())
}

/**
 * Returns a *typed* object array containing all of the elements of this primitive array.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.toUByteArray(): UByteArray {
    return UByteArray(this.copyOf())
}

/**
 * Returns an array of type [ByteArrayList], which is a copy of this array where each element is a signed reinterpretation
* of the corresponding element of this array.
*/
```kotlin
/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun LongArray.toULongArray(): ULongArray { return ULongArray(this.copyOf()) }

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes
public fun Array<out UShort>.toUShortArray(): UShortArray {
    return UShortArray(size) { index -> this[index] }
}

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly
public inline fun ShortArray.toUShortArray(): UShortArray {
    return UShortArray(this.copyOf())
}

/*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.associateWith(valueSelector: (UInt) -> V): Map<UInt, V> {
    val result = LinkedHashMap<UInt, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.associateWith(valueSelector: (ULong) -> V): Map<ULong, V> {
    val result = LinkedHashMap<ULong, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.associateWith(valueSelector: (UByte) -> V): Map<UByte, V> {
    val result = LinkedHashMap<UByte, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/*\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.associateWith(valueSelector: (UShort) -> V): Map<UShort, V> {
    val result = LinkedHashMap<UShort, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.toUShortArray(): UShortArray {
    return UShortArray(this.copyOf())
}

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.toULongArray(): ULongArray { return ULongArray(this.copyOf()) }

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.associateWith(valueSelector: (ULong) -> V): Map<ULong, V> {
    val result = LinkedHashMap<ULong, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.associateWith(valueSelector: (UByte) -> V): Map<UByte, V> {
    val result = LinkedHashMap<UByte, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}

/*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.associateWith(valueSelector: (UShort) -> V): Map<UShort, V> {
    val result = LinkedHashMap<UShort, V>(mapCapacity(size).coerceAtLeast(16))
    return associateWithTo(result, valueSelector)
}
```

Open Source Used In DNAC Platform Antelope 13961
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <V, M : MutableMap<in ULong, in V>> ULongArray.associateWithTo(destination: M, valueSelector: (ULong) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.
 */
@sample samples.collections.Collections.Transformations.associateWithTo

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <V, M : MutableMap<in UByte, in V>> UByteArray.associateWithTo(destination: M, valueSelector: (UByte) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array, where key is the element itself and value is provided by the [valueSelector] function applied to that key. If any two elements are equal, the last one overwrites the former value in the map.
 */
@sample samples.collections.Collections.Transformations.associateWithTo

@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <V, M : MutableMap<in UShort, in V>> UShortArray.associateWithTo(destination: M, valueSelector: (UShort) -> V): M {
    for (element in this) {
        destination.put(element, valueSelector(element))
    }
    return destination
}

/**
 * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.
 */
@sample samples.collections.Collections.Transformations.flatMap

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R> UIntArray.flatMap(transform: (UInt) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R> ULongArray.flatMap(transform: (ULong) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R> UByteArray.flatMap(transform: (UByte) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R> UShortArray.flatMap(transform: (UShort) -> Iterable<R>): List<R> {
    return flatMapTo(ArrayList<R>(), transform)
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R> UIntArray.flatMapIndexed(transform: (index: Int, UInt) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R> ULongArray.flatMapIndexed(transform: (index: Int, ULong) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R> UByteArray.flatMapIndexed(transform: (index: Int, UByte) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R> UShortArray.flatMapIndexed(transform: (index: Int, UShort) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}
```
ULongArray.flatMapIndexed(transform: (index: Int, ULong) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from 
 * [transform] function being invoked on each element * and its index in the original array.
 * @sample samples.collections.Transformations.flatMapIndexed
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>

UByteArray.flatMapIndexed(transform: (index: Int, UByte) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Returns a single list of all elements yielded from 
 * [transform] function being invoked on each element * and its index in the original array.
 * @sample samples.collections.Transformations.flatMapIndexed
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>

UShortArray.flatMapIndexed(transform: (index: Int, UShort) -> Iterable<R>): List<R> {
    return flatMapIndexedTo(ArrayList<R>(), transform)
}

/**
 * Appends all elements yielded from results of 
 * [transform] function being invoked on each element * and its index in the original array, to the given 
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C :

UIntArray.flatMapIndexedTo(destination: C, transform: (index: Int, UInt) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of 
 * [transform] function being invoked on each element * and its index in the original array, to the given 
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C :

UByteArray.flatMapIndexedTo(destination: C, transform: (index: Int, UByte) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}

/**
 * Appends all elements yielded from results of 
 * [transform] function being invoked on each element * and its index in the original array, to the given 
 * [destination].
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C :

UShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, UShort) -> Iterable<R>): C {
    var index = 0
    for (element in this) {
        val list = transform(index++, element)
        destination.addAll(list)
    }
    return destination
}
```
fun <R, C : MutableCollection<in R>> ULongArray.flatMapTo(destination: C, transform: (ULong) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

fun <R, C : MutableCollection<in R>> UByteArray.flatMapTo(destination: C, transform: (UByte) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

fun <R, C : MutableCollection<in R>> UShortArray.flatMapTo(destination: C, transform: (UShort) -> Iterable<R>): C {
    for (element in this) {
        val list = transform(element)
        destination.addAll(list)
    }
    return destination
}

fun <K> UIntArray.groupBy(keySelector: (UInt) -> K): Map<K, List<UInt>> {
    return groupByTo(LinkedHashMap<K, MutableList<UInt>>(), keySelector)
}

fun <K> ULongArray.groupBy(keySelector: (ULong) -> K): Map<K, List<ULong>> {
    return groupByTo(LinkedHashMap<K, MutableList<ULong>>(), keySelector)
}

fun <K> UByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {
    return groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)
}

fun <K, V> UIntArray.groupBy(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {
    return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)
}
the given `[keySelector]` function applied to the element
* and returns a map where each group key is associated
with a list of corresponding values.
* The returned map preserves the entry iteration order of the keys
produced from the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, V>
ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K, valueTransform: (ULong) -> V): Map<K, MutableList<V>> {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<V>()
        }
        list.add(valueTransform(element))
    }
    return destination
}
```

The returned map preserves the entry iteration order of the keys
produced from the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, M : MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<UInt>()
        }
        list.add(element)
    }
    return destination
}
```

The returned map preserves the entry iteration order of the keys
produced from the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, M : UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<UByte>()
        }
        list.add(element)
    }
    return destination
}
```

The returned map preserves the entry iteration order of the keys
produced from the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <K, M : UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K): M {
    for (element in this) {
        val key = keySelector(element)
        val list = destination.getOrPut(key) {
            ArrayList<UShort>()
        }
        list.add(element)
    }
    return destination
}
```

The returned map preserves the entry iteration order of the keys
produced from the original array.
MutableMap<in K, MutableList<UShort>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K):
M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrElse(key) {\n            ArrayList<UShort>()\n        }\n        list.add(element)\n    }\n    return destination\n} Groups values returned by the [valueTransform] function applied to each element of the original array\n* by the key returned by the given [keySelector] function applied to the element\n* and puts to the [destination] map each group key associated with a list of corresponding values.\n* @return The [destination] map\n* @sample samples.collections.Collections.Transformations.groupByKeysAndValues

MutableMap<in K, MutableList<UInt>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K, valueTransform: (UInt) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrElse(key) {\n            ArrayList<V>()\n        }\n        list.add(valueTransform(element))\n    }\n    return destination\n} Groups values returned by the [valueTransform] function applied to each element of the original array\n* by the key returned by the given [keySelector] function applied to the element\n* and puts to the [destination] map each group key associated with a list of corresponding values.\n* @return The [destination] map\n* @sample samples.collections.Collections.Transformations.groupByKeysAndValues

MutableMap<in K, MutableList<ULong>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K, valueTransform: (ULong) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrElse(key) {\n            ArrayList<V>()\n        }\n        list.add(valueTransform(element))\n    }\n    return destination\n} Groups values returned by the [valueTransform] function applied to each element of the original array\n* by the key returned by the given [keySelector] function applied to the element\n* and puts to the [destination] map each group key associated with a list of corresponding values.\n* @return The [destination] map\n* @sample samples.collections.Collections.Transformations.groupByKeysAndValues

MutableMap<in K, MutableList<UByte>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K, valueTransform: (UByte) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrElse(key) {\n            ArrayList<V>()\n        }\n        list.add(valueTransform(element))\n    }\n    return destination\n} Groups values returned by the [valueTransform] function applied to each element of the original array\n* by the key returned by the given [keySelector] function applied to the element\n* and puts to the [destination] map each group key associated with a list of corresponding values.\n* @return The [destination] map\n* @sample samples.collections.Collections.Transformations.groupByKeysAndValues

MutableMap<in K, MutableList<UShort>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K, valueTransform: (UShort) -> V): M {\n    for (element in this) {\n        val key = keySelector(element)\n        val list = destination.getOrElse(key) {\n            ArrayList<V>()\n        }\n        list.add(valueTransform(element))\n    }\n    return destination\n} Groups values returned by the [valueTransform] function applied to each element of the original array\n* by the key returned by the given [keySelector] function applied to the element\n* and puts to the [destination] map each group key associated with a list of corresponding values.\n* @return The [destination] map\n* @sample samples.collections.Collections.Transformations.groupByKeysAndValues

UIntArray.map(transform: (UInt) -> R): List<R>{\n    return mapTo(ArrayList<R>(size), transform)\n} Returns a list containing the results of applying the given [transform] function\n* to each element in the original array\n* @sample samples.collections.Collections.Transformations.map

ULongArray.map(transform: (ULong) -> R): List<R>{\n    return mapTo(ArrayList<R>(size), transform)\n} Returns a list containing the results of applying the given [transform] function\n* to each element in the original array\n* @sample samples.collections.Collections.Transformations.map

UByteArray.map(transform: (UByte) -> R): List<R>{\n    return mapTo(ArrayList<R>(size), transform)\n} Returns a list containing the results of applying the given [transform] function\n* to each element in the original array\n* @sample samples.collections.Collections.Transformations.map

UShortArray.map(transform: (UShort) -> R): List<R>{\n    return mapTo(ArrayList<R>(size), transform)\n} Returns a list containing the results of applying the given [transform] function\n* to each element in the original array\n* @sample samples.collections.Collections.Transformations.map

Open Source Used In DNAC Platform Antelope 13966
element in the original array.

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.map(transform: (UShort) -> R): List<R> { // ...
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.mapIndexed(transform: (Int, UInt) -> R): List<R> { // ...
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.mapIndexed(transform: (Int, ULong) -> R): List<R> { // ...
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.mapIndexed(transform: (Int, UByte) -> R): List<R> { // ...
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.mapIndexed(transform: (Int, UShort) -> R): List<R> { // ...
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.mapIndexedTo(destination: MutableCollection<R>, transform: (Int, UInt) -> R): MutableCollection<R> { // ...
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.mapIndexedTo(destination: MutableCollection<R>, transform: (Int, ULong) -> R): MutableCollection<R> { // ...
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.mapIndexedTo(destination: MutableCollection<R>, transform: (Int, UByte) -> R): MutableCollection<R> { // ...
```

```kotlin
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UShortArray.mapIndexedTo(destination: MutableCollection<R>, transform: (Int, UShort) -> R): MutableCollection<R> { // ...
```
/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Applies the given [transform] function to each element of the original array
 * and appends the results to the given [destination].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, C : MutableCollection<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {
    for (item in this)
        destination.add(transform(item))
    return destination
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.withIndex(): Iterable<IndexedValue<UInt>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.withIndex(): Iterable<IndexedValue<ULong>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.withIndex(): Iterable<IndexedValue<UByte>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns a lazy [Iterable] that wraps each element of the original array
 * into an [IndexedValue] containing the index of that element and the element itself.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.withIndex(): Iterable<IndexedValue<UShort>> {
    return IndexingIterable { iterator() }
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.all(predicate: (UInt) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.all(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.all(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}

/**
 * Returns `true` if all elements match the given [predicate].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.all(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (!predicate(element)) return false
    return true
}
* \n * @sample samples.collections.Collections.Aggregates.any

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.any(): Boolean {\n    return storage.any()\n} \n\n* Returns `true` if array has at least one element.\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.any

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.any(): Boolean {\n    return storage.any()\n} \n\n* Returns `true` if array has at least one element.\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.any

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.any(): Boolean {\n    return storage.any()\n} \n\n* Returns `true` if at least one element matches the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.any(predicate: (UInt) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true
    return false\n} \n\n* Returns `true` if at least one element matches the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.any(predicate: (ULong) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true
    return false\n} \n\n* Returns `true` if at least one element matches the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.any(predicate: (UByte) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true
    return false\n} \n\n* Returns `true` if at least one element matches the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.any(predicate: (UShort) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true
    return false\n} \n\n* Returns `true` if at least one element matches the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.count(predicate: (UInt) -> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count
    return count\n} \n\n* Returns the number of elements matching the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.count(predicate: (ULong) -> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count
    return count\n} \n\n* Returns the number of elements matching the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.count(predicate: (UByte) -> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count
    return count\n} \n\n* Returns the number of elements matching the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.count(predicate: (UShort) -> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count
    return count\n} \n\n* Returns the number of elements matching the given [predicate].\n* \n* \n* \n@sample samples.collections.Collections.Aggregates.anyWithPredicate

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n} \n\n* Returns the specified [initial] value if the array is empty.\n* \n* \n* \n@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n} \n\n* Returns the specified [initial] value if the array is empty.\n* \n* \n* \n@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

*\n*\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n} \n\n* Returns the specified [initial] value if the array is empty.\n* \n* \n* \n@param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
this) accumulator = operation(accumulator, element)  
    return accumulator
}  

// **
// * Accumulates value starting with [initial] value and applying [operation] from left to right * to current accumulator value and each element.
// * Returns the specified [initial] value if the array is empty.
// *
// * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
// *}
@SinceKotlin("1.3")  
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>  
UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {
    var accumulator = initial
    for (element in this) accumulator = operation(accumulator, element)
    return accumulator
}

// **
// * Accumulates value starting with [initial] value and applying [operation] from left to right * to current accumulator value and each element.
// * Returns the specified [initial] value if the array is empty.
// *
// * @param [operation] function that takes the index of an element, current accumulator value* and the element itself, and calculates the next accumulator value.
// *}
@SinceKotlin("1.3")  
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>  
UByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): R {
    var accumulator = initial
    var index = 0
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

// **
// * Accumulates value starting with [initial] value and applying [operation] from left to right * to current accumulator value and each element with its index in the original array.
// * Returns the specified [initial] value if the array is empty.
// *
// * @param [operation] function that takes the index of an element, current accumulator value* and the element itself, and calculates the next accumulator value.
// *}
@SinceKotlin("1.3")  
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>  
UIntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

// **
// * Accumulates value starting with [initial] value and applying [operation] from left to right * to current accumulator value and each element with its index in the original array.
// * Returns the specified [initial] value if the array is empty.
// *
// * @param [operation] function that takes the index of an element, current accumulator value* and the element itself, and calculates the next accumulator value.
// *}
@SinceKotlin("1.3")  
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>  
ULongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): R {
    var index = 0
    var accumulator = initial
    for (element in this) accumulator = operation(index++, accumulator, element)
    return accumulator
}

// **
// * Accumulates value starting with [initial] value and applying [operation] from right to left * to each element and current accumulator value.
// * Returns the specified [initial] value if the array is empty.
// *
// * @param [operation] function that takes the index of an element, current accumulator value* and the element itself, and calculates the next accumulator value.
// *}
@SinceKotlin("1.3")  
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>  
UShortArray.foldRight(initial: R, operation: (R, UShort) -> R): R {
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}
accumulator

Accumulates value starting with [initial] value and applying [operation] from right to left
* to each element and current accumulator value.

* Returns the specified [initial] value if the array is empty.

* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

*<R> ULongArray.foldRight(initial: R, operation: (ULong, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

Accumulates value starting with [initial] value and applying [operation] from right to left
* to each element and current accumulator value.

* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

*<R> UByteArray.foldRight(initial: R, operation: (UByte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
    }
    return accumulator
}

Accumulates value starting with [initial] value and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.

* @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.

*<R> UIntArray.foldRightIndexed(initial: R, operation: (index: Int, UInt, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

*<R> ULongArray.foldRightIndexed(initial: R, operation: (index: Int, ULong, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

Accumulates value starting with [initial] value and applying [operation] from right to left
* to each element with its index in the original array and current accumulator value.

* @param [operation] function that takes the index of an element, the element itself
* and current accumulator value, and calculates the next accumulator value.

*<R> UByteArray.foldRightIndexed(initial: R, operation: (index: Int, UByte, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun <R>
UShortArray.foldRightIndexed(initial: R, operation: (index: Int, UShort, acc: R) -> R): R {
    var index = lastIndex
    var accumulator = initial
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Performs the given [action] on each element.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun UIntArray.forEach(action: (UInt) -> Unit): Unit {
    for (element in this) action(element)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes the index of an element and the element itself and performs the action on the element.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun ULongArray.forEachIndexed(action: (index: Int, ULong) -> Unit): Unit {
    var index = 0
    for (item in this)
        action(index++, item)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes the index of an element and the element itself and performs the action on the element.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun UByteArray.forEachIndexed(action: (index: Int, UByte) -> Unit): Unit {
    var index = 0
    for (item in this)
        action(index++, item)
}

/**
 * Performs the given [action] on each element, providing sequential index with the element.
 * @param [action] function that takes the index of an element and the element itself and performs the action on the element.
 */
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
public inline fun UShortArray.forEachIndexed(action: (index: Int, UShort) -> Unit): Unit {
    var index = 0
    for (item in this)
        action(index++, item)
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.max(): UInt? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.max(): ULong? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.max(): UByte? {
    return maxOrNull()
}

@Deprecated("Use maxOrNull instead.", ReplaceWith("maxOrNull()"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.max(): UShort? {
    return maxOrNull()
}

@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntArray.maxByOrNull(selector: (UInt) -> Boolean): ULong? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongArray.maxByOrNull(selector: (ULong) -> Boolean): ULong? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UByteArray.maxByOrNull(selector: (UByte) -> Boolean): UByte? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.", ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UShortArray.maxByOrNull(selector: (UShort) -> Boolean): UShort? {
    return maxByOrNull(selector)
}
1.4")@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UIntArray.maxBy(selector: (UInt) -> R): UInt? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.",
ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.maxBy(selector: (ULong) -> R): ULong? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.",
ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte? {
    return maxByOrNull(selector)
}

@Deprecated("Use maxByOrNull instead.",
ReplaceWith("maxByOrNull(selector)"))
@DeprecatedSinceKotlin(warningSince = "1.4")@SinceKotlin("1.3")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort? {
    return maxByOrNull(selector)
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UIntArray.maxByOrNull(selector: (UInt) -> R): UInt? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.maxByOrNull(selector: (ULong) -> R): ULong? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.maxByOrNull(selector: (UByte) -> R): UByte? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the first element yielding the largest value of the given function or `null` if there are no elements.
 * 
 * @sample samples.collections.Collections.Aggregates.maxByOrNull
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.maxByOrNull(selector: (UShort) -> R): UShort? {
    if (isEmpty()) return null
    var maxElem = this[0]
    val lastIndex = this.lastIndex
    if (lastIndex == 0) return maxElem
    var maxValue = selector(maxElem)
    for (i in 1..lastIndex) {
        val e = this[i]
        val v = selector(e)
        if (maxValue < v) {
            maxElem = e
            maxValue = v
        }
    }
    return maxElem
}

/**
 * Returns the largest value among all values produced by [selector] function applied to each element in the array.
 * 
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes@kotlin.internal.InlineOnly
public inline fun UIntArray.maxOf(selector: (UInt) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
...
var maxValue = selector(this[0])
for (i in 1..lastIndex) {
    val v = selector(this[i])
    maxVal = maxOf(maxValue, v, true)
    return maxVal
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun ULongArray.maxOf(selector: (ULong) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v, true)
    }
    return maxValue
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun UByteArray.maxOf(selector: (UByte) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v, true)
    }
    return maxValue
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun UShortArray.maxOf(selector: (UShort) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v, true)
    }
    return maxValue
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun UIntArray.maxOf(selector: (UInt) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v, true)
    }
    return maxValue
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun ULongArray.maxOf(selector: (ULong) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v, true)
    }
    return maxValue
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun UByteArray.maxOf(selector: (UByte) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v, true)
    }
    return maxValue
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun UIntArray.maxOf(selector: (UInt) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v, true)
    }
    return maxValue
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun ULongArray.maxOf(selector: (ULong) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v, true)
    }
    return maxValue
}

@OptIn(ClassTypeInference::class)
@OverloadResolution
ByLambdaReturnType
public inline fun UByteArray.maxOf(selector: (UByte) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
public inline fun UShortArray.maxOf(selector: (UShort) -> Float): Float {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by 
 * [selector] function applied to each element in the 
 * array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun ULongArray.maxOf(selector: (ULong) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced 
 * by [selector] function applied to each element in the 
 * array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun UByteArray.maxOf(selector: (UByte) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced 
 * by [selector] function applied to each element in the 
 * array or `null` if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
public inline fun UCharArray.maxOf(selector: (UChar) -> Double): Double? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        maxValue = maxOf(maxValue, v)
    }
    return maxValue
}
values produced by [selector] function is `NaN`, the returned result is `NaN`.

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.maxOfOrNull(selector: (UByte) -> Double): Double? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.maxOfOrNull(selector: (UByte) -> Float): Float? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.maxOfOrNull(selector: (UShort) -> Double): Double? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.maxOfOrNull(selector: (UShort) -> Float): Float? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.maxOfOrNull(selector: (UInt) -> Float): Float? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.maxOfOrNull(selector: (UInt) -> R): R? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.maxOfOrNull(selector: (ULong) -> Double): Double? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.maxOfOrNull(selector: (ULong) -> Float): Float? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UIntArray.maxOfOrNull(selector: (UInt) -> R): R? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```

```kotlin
@Selector
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
@Inplace
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.maxOfOrNull(selector: (ULong) -> R): R? {
    if (isEmpty()) return null
    var max = selector(this[0])
    for (i in 1..lastIndex) {
        var v = selector(this[i])
        max = maxOf(max, v)
    }
    return max
}
```
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.maxOfOrNull(selector: (ULong) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.maxOfOrNull(selector: (UByte) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value among all values produced by [selector] function
 * applied to each element in the array or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.maxOfOrNull(selector: (UShort) -> R): R? {
    if (isEmpty()) return null
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (maxValue < v) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator]
 * among all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UIntArray.maxOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator]
 * among all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.maxOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator]
 * among all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.maxOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator]
 * among all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.maxOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator]
 * among all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.maxOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

/**
 * Returns the largest value according to the provided [comparator]
 * among all values produced by [selector] function applied to each element in the array.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.maxOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var maxValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(maxValue, v) < 0) {
            maxValue = v
        }
    }
    return maxValue
}

Open Source Used In DNAC Platform Antelope 13977
maxValue
} \n \n \n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]\n function applied to each element in the array or `null` if there are no elements.

 *\n @ Since Kotlin ("1.4") *\n @ OptIn (kotlin.experimental.ExperimentalTypeInference::class) *\n @ OverloadResolution
 By Lambda Return Type \n @ ExperimentalUnsignedTypes \n @ kotlin.internal.InlineOnly \n public inline fun <R> 
 UIntArray.maxOfOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? { \n \n if (is-empty()) return null \n \n var maxValue = selector(this[0]) \n \n for (i in 1..lastIndex) { \n \n val v = selector(this[i]) \n \n if (comparator.compare(maxValue, v) < 0) { \n \n maxValue = v \n \n } \n \n } \n \n return maxValue \n } \n
 *\n @ Since Kotlin ("1.4") *\n @ OptIn (kotlin.experimental.ExperimentalTypeInference::class) *\n @ OverloadResolution
 By Lambda Return Type \n @ ExperimentalUnsignedTypes \n @ kotlin.internal.InlineOnly \n public inline fun <R> 
 ULongArray.maxOfOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? { \n \n if (is-empty()) return null \n \n var maxValue = selector(this[0]) \n \n for (i in 1..lastIndex) { \n \n val v = selector(this[i]) \n \n if (comparator.compare(maxValue, v) < 0) { \n \n maxValue = v \n \n } \n \n } \n \n return maxValue \n } \n
 *\n @ Since Kotlin ("1.4") *\n @ OptIn (kotlin.experimental.ExperimentalTypeInference::class) *\n @ OverloadResolution
 By Lambda Return Type \n @ ExperimentalUnsignedTypes \n @ kotlin.internal.InlineOnly \n public inline fun <R> 
 UByteArray.maxOfOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? { \n \n if (is-empty()) return null \n \n var maxValue = selector(this[0]) \n \n for (i in 1..lastIndex) { \n \n val v = selector(this[i]) \n \n if (comparator.compare(maxValue, v) < 0) { \n \n maxValue = v \n \n } \n \n } \n \n return maxValue \n } \n
 *\n @ Since Kotlin ("1.4") *\n @ OptIn (kotlin.experimental.ExperimentalTypeInference::class) *\n @ OverloadResolution
 By Lambda Return Type \n @ ExperimentalUnsignedTypes \n @ kotlin.internal.InlineOnly \n public inline fun <R> 
 UShortArray.maxOfOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? { \n \n if (is-empty()) return null \n \n var maxValue = selector(this[0]) \n \n for (i in 1..lastIndex) { \n \n val v = selector(this[i]) \n \n if (comparator.compare(maxValue, v) < 0) { \n \n maxValue = v \n \n } \n \n } \n \n return maxValue \n } \n
 @ ExperimentalUnsignedTypes
 @ kotlin.internal.InlineOnly
 public inline fun <R> 
 UIntArray.maxOrNull(): UInt? { \n \n if (isEmpty()) return null \n \n var max = this[0] \n \n for (i in 1..lastIndex) { \n \n val e = this[i] \n \n if (max < e) max = e \n \n } \n \n return max \n } \n
 @ ExperimentalUnsignedTypes
 @ kotlin.internal.InlineOnly
 public inline fun <R> 
 ULongArray.maxOrNull(): ULong? { \n \n if (isEmpty()) return null \n \n var max = this[0] \n \n for (i in 1..lastIndex) { \n \n val e = this[i] \n \n if (max < e) max = e \n \n } \n \n return max \n } \n
 @ ExperimentalUnsignedTypes
 @ kotlin.internal.InlineOnly
 public inline fun <R> 
 UByteArray.maxOrNull(): UByte? { \n \n if (isEmpty()) return null \n \n var max = this[0] \n \n for (i in 1..lastIndex) { \n \n val e = this[i] \n \n if (max < e) max = e \n \n } \n \n return max \n } \n
 @ ExperimentalUnsignedTypes
 @ kotlin.internal.InlineOnly
 public inline fun <R> 
 UShortArray.maxOrNull(): UShort? { \n \n if (isEmpty()) return null \n \n var max = this[0] \n \n for (i in 1..lastIndex) { \n \n val e = this[i] \n \n if (max < e) max = e \n \n } \n \n return max \n }
ReplaceWith("maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.\nmaxWith(comparator: Comparator<in UByte>): UByte? \n{\n    return maxWithOrNull(comparator)\n}\n\n@Deprecated("Use maxWithOrNull instead.\n\nReplaceWith("maxWithOrNull(comparator)"))\n@DeprecatedSinceKotlin(warningSince = "1.4")\n@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.\nmaxWith(comparator: Comparator<in UShort>): UShort? \n{\n    return maxWithOrNull(comparator)\n}\n
@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.\nmaxWithOrNull(comparator: Comparator<in UInt>): UInt? {
    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {
        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.\nmaxWithOrNull(comparator: Comparator<in ULong>): ULong? \n{\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {
        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.\nmaxWithOrNull(comparator: Comparator<in UByte>): UByte? {
    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {
        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.\nmaxWithOrNull(comparator: Comparator<in UShort>): UShort? {
    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {
        val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e
    }
    return max
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.\nmin(): UInt? {
    return minOrNull()
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.\nmin(): ULong? {
    return minOrNull()
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.\nmin(): UByte? {
    return minOrNull()
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.\nmin(): UShort? {
    return minOrNull()
}

@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.\nminByOrNull(selector: (UInt) -> R): UInt? {
    return minByOrNull(selector)\n}\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.\nminByOrNull(selector: (ULong) -> R): ULong? {
    return minByOrNull(selector)\n}\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.\nminByOrNull(selector: (UByte) -> R): UByte? {
    return minByOrNull(selector)\n}\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.\nminByOrNull(selector: (UShort) -> R): UShort? {
    return minByOrNull(selector)\n}\n
@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun\ninline fun <R : Comparable<R>> UIntArray.\nminByOrNull(selector: (UInt) -> R): UInt? {
    return minByOrNull(selector)\n}\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.\nminByOrNull(selector: (ULong) -> R): ULong? {
    return minByOrNull(selector)\n}\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.\nminByOrNull(selector: (UByte) -> R): UByte? {
    return minByOrNull(selector)\n}\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.\nminByOrNull(selector: (UShort) -> R): UShort? {
    return minByOrNull(selector)\n}\n
@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun\ninline fun <R : Comparable<R>> UByteArray.\nminByOrNull(selector: (UByte) -> R): UByte? {
    return minByOrNull(selector)\n}\n\n@SinceKotlin("1.4")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.\nminByOrNull(selector: (UShort) -> R): UShort? {
    return minByOrNull(selector)\n}\n
Open Source Used In DNAC Platform Antelope 13979
ReplaceWith("minByOrNull(selector)\n")@DeprecatedSinceKotlin(warningSince = "1.4")@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.minBy(selector: (UShort) -> R): UShort? {
    return minByOrNull(selector)\n}/** * Returns the first element yielding the smallest value of the given function or ‘null’ if there are no elements.* * @sample samples.collections.Collections.Aggregates.minByOrNull */
@DeprecatedSinceKotlin(warningSince = "1.4")@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.minByOrNull(selector: (UInt) -> R): UInt? {
    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {
        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {
            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}/** * Returns the first element yielding the smallest value of the given function or ‘null’ if there are no elements.* * @sample samples.collections.Collections.Aggregates.minByOrNull */
@DeprecatedSinceKotlin(warningSince = "1.4")@SinceKotlin("1.3")@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.minByOrNull(selector: (ULong) -> R): ULong? {
    if (isEmpty()) return null\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex == 0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {
        val e = this[i]\n        val v = selector(e)\n        if (minValue > v) {
            minElem = e\n            minValue = v\n        }\n    }\n    return minElem\n}/** * Returns the first element yielding the smallest value of the given function or ‘null’ if there are no elements.* * @sample samples.collections.Collections.Aggregates.minByOrNull */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly\npublic inline fun UIntArray.minOf(selector: (UInt) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {
        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}/** * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.minOf(selector: (ULong) -> Double): Double {
    if (isEmpty()) throw NoSuchElementException\n    var minValue = selector(this[0])\n    for (i in 1..lastIndex) {
        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n    }\n    return minValue\n}/** * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * @throws NoSuchElementException if the array is empty.\n */
Optionalsuspend fun UByteArray.minOf(selector: (UByte) -> Double): Double {
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function applied to each element in the array.
 * If any of values produced by [selector] function
 * is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
Optionalsuspend fun UShortArray.minOf(selector: (UShort) -> Double): Double {
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function applied to each element in the array.
 * If any of values produced by [selector] function
 * is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
Optionalsuspend fun UIntArray.minOf(selector: (UInt) ->Float): Float {
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function applied to each element in the array.
 * If any of values produced by [selector] function
 * is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
Optionalsuspend fun ULongArray.minOf(selector: (ULong) ->Float): Float {
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced
 * by [selector] function applied to each element in the array.
 * If any of values produced by [selector] function
 * is `NaN`, the returned result is `NaN`.
 * @throws NoSuchElementException if the array is empty.
 */
Optionalsuspend fun <R : Comparable<R>> UByteArray.minOf(selector: (UByte) -> R): R {
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}
```kotlin
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@kotlin.experimental.InlineOnly
public inline fun <R : Comparable<R>> ULongArray.minOf(selector: (ULong) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R : Comparable<R>> UByteArray.minOf(selector: (UByte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun <R : Comparable<R>> UShortArray.minOf(selector: (UShort) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun UIntArray.minOfOrNull(selector: (UInt) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun ULongArray.minOfOrNull(selector: (ULong) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun UByteArray.minOfOrNull(selector: (UByte) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}

/**
 * Returns the smallest value among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.experimental.InlineOnly
public inline fun UShortArray.minOfOrNull(selector: (UShort) -> Double): Double? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        minValue = minOf(minValue, v)
    }
    return minValue
}
```

// return minValue
// * Returns the smallest value among all values produced by [selector] function
// * applied to each element in the array or `null` if there are no elements.
// * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.

/*@SinceKotlin("1.4")*/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
UIntArray.minOfOrNull(selector: (UInt) -> Float): Float? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) minValue = minOf(minValue, v)
}
return minValue

/*@SinceKotlin("1.4")*/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
ULongArray.minOfOrNull(selector: (ULong) -> Float): Float? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) minValue = minOf(minValue, v)
}
return minValue

/*@SinceKotlin("1.4")*/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
UByteArray.minOfOrNull(selector: (UByte) -> Float): Float? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) minValue = minOf(minValue, v)
}
return minValue

/*@SinceKotlin("1.4")*/@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
UShortArray.minOfOrNull(selector: (UShort) -> Float): Float? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) minValue = minOf(minValue, v)
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 * If any of values produced by [selector] function is 'NaN', the returned result is 'NaN'.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
 */
@SinceKotlin("1.4")@OptIn(kotlin.experimental.ExperimentalTypeInference::class)@OverloadResolution
ByLambdaReturnType@ExperimentalUnsignedTypes}@kotlin.internal.InlineOnly}/public inline fun
Comparable<R>.minOfOrNull(selector: (R) -> R): R? { if (isEmpty()) return null
var minValue = selector(this[0]) for (i in 1..lastIndex) { val v = selector(this[i]) if (minValue > v) {
minValue = v
}
}
return minValue

/**
 * Returns the smallest value among all values produced by [selector] function
 * applied to each element in the array or `null` if there are no elements.
```kotlin
public inline fun <R : Comparable<R>> UShortArray.minOfOrNull(selector: (UShort) -> R): R? {
    if (isEmpty()) return null
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (minValue > v) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
public inline fun <R> UIntArray.minOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
public inline fun <R> ULongArray.minOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
public inline fun <R> UByteArray.minOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}

/**
 * Returns the smallest value according to the provided [selector] function applied to each element in the array.
 * @throws NoSuchElementException if the array is empty.
 */
public inline fun <R> UShortArray.minOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {
    if (isEmpty()) throw NoSuchElementException()
    var minValue = selector(this[0])
    for (i in 1..lastIndex) {
        val v = selector(this[i])
        if (comparator.compare(minValue, v) > 0) {
            minValue = v
        }
    }
    return minValue
}
```

Open Source Used In DNAC Platform Antelope 13984
Open Source Used In DNAC Platform Antelope 13985

return null
var minValue = selector(this[0])
for (i in 1..lastIndex) {
  val v = selector(this[i])
  if (comparator.compare(minValue, v) > 0) {
    minValue = v
  }
}
return minValue

/**
 * Returns the smallest value according to the provided [comparator] among all values produced by [selector] function applied to each element in the array or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
@OptIn(kotlin.experimental.ExperimentalTypeInference::class)
@OverloadResolution
ByLambdaReturnType
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R>
UByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {
  if (isEmpty()) return null
  var minValue = selector(this[0])
  for (i in 1..lastIndex) {
    val v = selector(this[i])
    if (comparator.compare(minValue, v) > 0) {
      minValue = v
    }
  }
  return minValue
}

/**
 * Returns the smallest element or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.minOrNull(): UInt? {
  if (isEmpty()) return null
  var min = this[0]
  for (i in 1..lastIndex) {
    val e = this[i]
    if (min > e) min = e
  }
  return min
}

/**
 * Returns the smallest element or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.minOrNull(): ULong? {
  if (isEmpty()) return null
  var min = this[0]
  for (i in 1..lastIndex) {
    val e = this[i]
    if (min > e) min = e
  }
  return min
}

/**
 * Returns the smallest element or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.minOrNull(): UByte? {
  if (isEmpty()) return null
  var min = this[0]
  for (i in 1..lastIndex) {
    val e = this[i]
    if (min > e) min = e
  }
  return min
}

/**
 * Returns the smallest element or 'null' if there are no elements.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.minOrNull(): UShort? {
  if (isEmpty()) return null
  var min = this[0]
  for (i in 1..lastIndex) {
    val e = this[i]
    if (min > e) min = e
  }
  return min
}

@Deprecated("Use minWithOrNull instead.", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@ExperimentalUnsignedTypes
public fun UIntArray.minWith(comparator: Comparator<in UInt>): UInt? {
  return minWithOrNull(comparator)
}

@Deprecated("Use minWithOrNull instead.", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@ExperimentalUnsignedTypes
public fun ULongArray.minWith(comparator: Comparator<in ULong>): ULong? {
  return minWithOrNull(comparator)
}

@Deprecated("Use minWithOrNull instead.", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@ExperimentalUnsignedTypes
public fun UByteArray.minWith(comparator: Comparator<in UByte>): UByte? {
  return minWithOrNull(comparator)
}

@Deprecated("Use minWithOrNull instead.", ReplaceWith("minWithOrNull(comparator)"))
@DeprecatedSinceKotlin(warningSince = "1.4")
@ExperimentalUnsignedTypes
public fun UShortArray.minWith(comparator: Comparator<in UShort>): UShort? {
  return minWithOrNull(comparator)
}
@SinceKotlin("1.4")@ExperimentalUnsignedTypes
public fun ULongArray.minWithOrNull(comparator: Comparator<in ULong>): ULong? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes
public fun UByteArray.minWithOrNull(comparator: Comparator<in UByte>): UByte? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes
public fun UShortArray.minWithOrNull(comparator: Comparator<in UShort>): UShort? {
    if (isEmpty()) return null
    var min = this[0]
    for (i in 1..lastIndex) {
        val e = this[i]
        if (comparator.compare(min, e) > 0) min = e
    }
    return min
}

/**
 * Returns `true` if the array has no elements.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if the array has no elements.
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(): Boolean {
    return isEmpty()
}

/**
 * Returns `true` if no elements match the given [predicate].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.none(predicate: (ULong) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.none(predicate: (UByte) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Returns `true` if no elements match the given [predicate].
 */
@SinceKotlin("1.3")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.none(predicate: (UShort) -> Boolean): Boolean {
    for (element in this) if (predicate(element)) return false
    return true
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
fun ULongArray.onEach(action: (ULong) -> Unit): ULongArray {
    return apply { for (element in this)
        action(element)
    }
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
fun UByteArray.onEach(action: (UByte) -> Unit): UByteArray {
    return apply { for (element in this)
        action(element)
    }
}

/**
 * Performs the given [action] on each element and returns the array itself afterwards.
 */
@SinceKotlin("1.4")@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
fun UShortArray.onEach(action: (UShort) -> Unit): UShortArray {
    return apply { for (element in this)
        action(element)
    }
}

Open Source Used In DNAC Platform Antelope 13986
afterwards.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public

```

```kotlin
in fun UShortArray.onEachIndexed(action: (Int, UShort) -> Unit): UShortArray {
    return apply {
        forEachIndexed { index, element -> action(index, element) }
    }
}
```

```kotlin
/**
 * Performs the given [action] on each element, providing sequential index
 * and returns the array itself afterwards.
 * @param [action] function that takes the index of an element and
 * the element itself
 * and performs the action on the element.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public

```

```kotlin
in fun ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation]
 * from left to right
 * to current accumulator value and each element. It returns `null` when its
 * receiver is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public

```

```kotlin
in fun UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in 1..lastIndex) {
        accumulator = operation(accumulator, this[index])
    }
    return accumulator
}
```

```kotlin
/**
 * Accumulates value starting with the first element and applying [operation]
 * from left to right
 * to current accumulator value and each element. It returns `null` when its
 * receiver is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public

```

```kotlin
in fun UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {
    if (isEmpty())
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = this[0]
    for (index in
```kotlin
/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * @param [operation] function that takes current accumulator value and an element,
 * accumulates value from left to right.
 * @return the resulting accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return this.reduceIndexed(operation)
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * @param [operation] function that takes current accumulator value and an element,
 * accumulates value from left to right.
 * @return the resulting accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt {
    return this.reduceIndexedIndexed(operation)
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * @param [operation] function that takes current accumulator value and an element,
 * accumulates value from left to right.
 * @return the resulting accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceIndexedIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong {
    return this.reduceIndexedIndexedIndexed(operation)
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * @param [operation] function that takes current accumulator value and an element,
 * accumulates value from left to right.
 * @return the resulting accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduce
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceIndexedIndexedIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte {
    return this.reduceIndexedIndexedIndexedIndexed(operation)
}
```
throw UnsupportedOperationException("Empty array can't be reduced.");

var accumulator = this[0]
for (index in 1..lastIndex)
    accumulator = operation(index, accumulator, this[index])

return accumulator

* Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. * Returns `null` if the array is empty.

* * * @param [operation] function that takes the index of an element, current accumulator value and the element itself, * and calculates the next accumulator value.

* * * @sample samples.collections.Collections.Aggregates.reduceOrNonNull

* * * @SinceKotlin("1.4") * @ExperimentalUnsignedTypes * @kotlin.internal.InlineOnly * @public inline fun UIntArray.reduceIndexedOrNull(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt?

* * * if (isEmpty())
    return null

* * * var accumulator = this[0]

* * * for (index in 1..lastIndex)
    accumulator = operation(index, accumulator, this[index])

* * * return accumulator

* * * * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. * Returns `null` if the array is empty.

* * * * @param [operation] function that takes the index of an element, current accumulator value and the element itself, * and calculates the next accumulator value.

* * * * @sample samples.collections.Collections.Aggregates.reduceOrNonNull

* * * * @SinceKotlin("1.4") * @ExperimentalUnsignedTypes * @kotlin.internal.InlineOnly * @public inline fun ULongArray.reduceIndexedOrNull(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong?

* * * * if (isEmpty())
    return null

* * * * var accumulator = this[0]

* * * * for (index in 1..lastIndex)
    accumulator = operation(index, accumulator, this[index])

* * * * return accumulator

* * * * * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. * Returns `null` if the array is empty.

* * * * * @param [operation] function that takes the index of an element, current accumulator value and the element itself, * and calculates the next accumulator value.

* * * * * @sample samples.collections.Collections.Aggregates.reduceOrNonNull

* * * * * @SinceKotlin("1.4") * @ExperimentalUnsignedTypes * @kotlin.internal.InlineOnly * @public inline fun UByteArray.reduceIndexedOrNull(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte?

* * * * * if (isEmpty())
    return null

* * * * * var accumulator = this[0]

* * * * * for (index in 1..lastIndex)
    accumulator = operation(index, accumulator, this[index])

* * * * * return accumulator

* * * * * * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element with its index in the original array. * Returns `null` if the array is empty.

* * * * * * @param [operation] function that takes the index of an element, current accumulator value and the element itself, * and calculates the next accumulator value.

* * * * * * @sample samples.collections.Collections.Aggregates.reduceOrNonNull

* * * * * * @SinceKotlin("1.4") * @ExperimentalUnsignedTypes * @kotlin.internal.InlineOnly * @public inline fun UShortArray.reduceIndexedOrNull(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort?

* * * * * * if (isEmpty())
    return null

* * * * * * var accumulator = this[0]

* * * * * * for (index in 1..lastIndex)
    accumulator = operation(index, accumulator, this[index])

* * * * * * return accumulator

* * * * * * * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. * Returns `null` if the array is empty.

* * * * * * * * @param [operation] function that takes the index of an element, current accumulator value and the element itself, * and calculates the next accumulator value.

* * * * * * * * @sample samples.collections.Collections.Aggregates.reduceOrNonNull

* * * * * * * * @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes * @kotlin.internal.InlineOnly * @public inline fun UIntArray.reduceOrNull(operation: (acc: UInt, UInt) -> UInt): UInt?

* * * * * * * * if (isEmpty())
    return null

* * * * * * * * var accumulator = this[0]

* * * * * * * * for (index in 1..lastIndex)
    accumulator = operation(accumulator, this[index])

* * * * * * * * return accumulator

* * * * * * * * * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. * Returns `null` if the array is empty.

* * * * * * * * * @param [operation] function that takes the current accumulator value and an element, * and calculates the next accumulator value.

* * * * * * * * * @sample samples.collections.Collections.Aggregates.reduceOrNonNull

* * * * * * * * * @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes * @kotlin.internal.InlineOnly * @public inline fun ULongArray.reduceOrNull(operation: (acc: ULong, ULong) -> ULong): ULong?

* * * * * * * * * if (isEmpty())
    return null

* * * * * * * * * var accumulator = this[0]

* * * * * * * * * for (index in 1..lastIndex)
    accumulator = operation(accumulator, this[index])

* * * * * * * * * return accumulator

* * * * * * * * * * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. * Returns `null` if the array is empty.

* * * * * * * * * * @param [operation] function that takes the current accumulator value and an element, * and calculates the next accumulator value.

* * * * * * * * * * @sample samples.collections.Collections.Aggregates.reduceOrNonNull

* * * * * * * * * * @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes * @kotlin.internal.InlineOnly * @public inline fun UByteArray.reduceOrNull(operation: (acc: UByte, UByte) -> UByte): UByte?

* * * * * * * * * * * if (isEmpty())
    return null

* * * * * * * * * * * var accumulator = this[0]

* * * * * * * * * * * for (index in 1..lastIndex)
    accumulator = operation(accumulator, this[index])

* * * * * * * * * * * return accumulator

* * * * * * * * * * * * Accumulates value starting with the first element and applying [operation] from left to right to current accumulator value and each element. * Returns `null` if the array is empty.

* * * * * * * * * * * @param [operation] function that takes the current accumulator value and an element, * and calculates the next accumulator value.

* * * * * * * * * * * @sample samples.collections.Collections.Aggregates.reduceOrNonNull

* * * * * * * * * * * @SinceKotlin("1.4") * @WasExperimental(ExperimentalStdlibApi::class) * @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceOrNull(operation: (acc: ULong, ULong) -> ULong): ULong? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex)
        accumulator = operation(accumulator, this[index])
    return accumulator
}

/**
 * Accumulates value starting with the first element and applying [operation] from left to right
 * to current accumulator value and each element.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes current accumulator value and an element,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceOrNull
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceOrNull(operation: (acc: UByte, UByte) -> UByte): UByte? {
    if (isEmpty())
        return null
    var accumulator = this[0]
    for (index in 1..lastIndex)
        accumulator = operation(accumulator, this[index])
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0)
        accumulator = operation(get(index--), accumulator)
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceRight(operation: (acc: UByte, UByte) -> UByte): UByte {
    var index = lastIndex
    if (index < 0)
        throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0)
        accumulator = operation(get(index--), accumulator)
    return accumulator
}
accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to
 * left to each element and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes an element and current accumulator value and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort): UShort {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(accumulator, get(index--))
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to
 * left to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceRightIndexed(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) { accolator = operation(index, get(index), accumulator)
    }
    --index
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to
 * left to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceRightIndexed(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to
 * left to each element with its index in the original array and current accumulator value.
 * Throws an exception if this array is empty. If the array can be empty in an expected way,
 * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.reduceRight
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceRightIndexed(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
    }
    return accumulator
}
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull

 * @SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceRightIndexedOrNull(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull

 * @SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceRightIndexedOrNull(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull

 * @SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceRightIndexedOrNull(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull

 * @SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceRightOrNull(operation: (UInt, acc: UInt) -> UInt): UInt? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
        --index
    }
    return accumulator
}

/*\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceRightIndexed(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort {
    var index = lastIndex
    if (index < 0) throw UnsupportedOperationException("Empty array can't be reduced.")
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull

 * @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceRightIndexedOrNull(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element with its index in the original array and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes the index of an element, the element itself and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull

 * @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.reduceRightIndexedOrNull(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull

 * @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.reduceRightIndexedOrNull(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(index, get(index), accumulator)
        --index
    }
    return accumulator
}

/**
 * Accumulates value starting with the last element and applying [operation] from right to left
 * to each element and current accumulator value.
 * Returns `null` if the array is empty.
 * @param [operation] function that takes an element and current accumulator value,
 * and calculates the next accumulator value.
 * @sample samples.collections.Aggregates.reduceRightOrNull

 * @ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.reduceRightOrNull(operation: (UInt, acc: UInt) -> UInt): UInt? {
    var index = lastIndex
    if (index < 0) return null
    var accumulator = get(index--)
    while (index >= 0) {
        accumulator = operation(get(index--), accumulator)
        --index
    }
    return accumulator
}*/

Open Source Used In DNAC Platform Antelope 13992
Accumulates value starting with the last element and applying [operation] from right to left * to each element and current accumulator value.

* `null` if the array is empty.
* `@param [operation] function that takes an element and current accumulator value.
* and calculates the next accumulator value.

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.reduceRightOrNull

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.reduceRightOrNull

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold

* `null` if the array is empty.

* `@param samples.collections.Collections.Aggregates.runningFold
element, and calculates the next accumulator value.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public inline fun <R> UByteArray.runningFold(initial: R, operation: (acc: R, UByte) -> R): List<R> {
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
    }
    return listOf(accumulator)
}
```

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal:inlineOnly
    var accumulator = initial
    for (element in this) {
        accumulator = operation(accumulator, element)
    }
    return listOf(accumulator)
}
```

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal:inlineOnly
public inline fun <R> UIntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
    }
    return listOf(accumulator)
}
```

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal:inlineOnly
public inline fun <R> ULongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
    }
    return listOf(accumulator)
}
```

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
current accumulator value that starts with [initial] value.

Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.

function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

@sample samples.collections.Collections.Aggregates.runningFold

```kotlin
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    if (isEmpty()) return listOf(initial)
    val result = ArrayList<R>(size + 1).apply { add(initial) }
    var accumulator = initial
    for (index in indices) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
```

```kotlin
/**
 * Returns a list containing successive accumulation values generated by applying
 * operation from left to right to each element and current accumulator value that
 * starts with the first element of this array.

 * Note that `acc` value passed to [operation] function should not be mutated;
 * otherwise it would affect the previous value in resulting list.

 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.
 * @sample samples.collections.Collections.Aggregates.runningReduce

 * @SinceKotlin("1.4")
 * @ExperimentalUnsignedTypes
 * @kotlin.internal.InlineOnly
 * if (isEmpty()) return emptyList()
 * var accumulator = this[0]
 * val result = ArrayList<UInt>(size).apply { add(accumulator) }
 * for (index in 1 until size) {
 *     accumulator = operation(accumulator, this[index])
 *     result.add(accumulator)
 * }
 * return result
 * }
```
/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right \(n\) * to each element, its index in the original array and current accumulator value that
 * starts with the first element of this array.\(n\) * \(n\) * Note that \`acc\` value passed to [operation] function should not be mutated;\(n\) * otherwise it would affect the previous value in resulting list.\(n\) * \(n\) * \@param [operation] function that takes the index of an element, current accumulator value\(n\) * and the element itself, and calculates the next accumulator value.\(n\) * \(n\) * \@sample

samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

UIntArray.runningReduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): List<UInt> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UInt>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right \(n\) * to each element, its index in the original array and current accumulator value that
 * starts with the first element of this array.\(n\) * \(n\) * Note that \`acc\` value passed to [operation] function should not be mutated;\(n\) * otherwise it would affect the previous value in resulting list.\(n\) * \(n\) * \@param [operation] function that takes the index of an element, current accumulator value\(n\) * and the element itself, and calculates the next accumulator value.\(n\) * \(n\) * \@sample

samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

ULongArray.runningReduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): List<ULong> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<ULong>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right \(n\) * to each element, its index in the original array and current accumulator value that
 * starts with the first element of this array.\(n\) * \(n\) * Note that \`acc\` value passed to [operation] function should not be mutated;\(n\) * otherwise it would affect the previous value in resulting list.\(n\) * \(n\) * \@param [operation] function that takes the index of an element, current accumulator value\(n\) * and the element itself, and calculates the next accumulator value.\(n\) * \(n\) * \@sample

samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

UByteArray.runningReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {
    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UByte>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right \(n\) * to each element, its index in the original array and current accumulator value that
 * starts with the first element of this array.\(n\) * \(n\) * Note that \`acc\` value passed to [operation] function should not be mutated;\(n\) * otherwise it would affect the previous value in resulting list.\(n\) * \(n\) * \@param [operation] function that takes the index of an element, current accumulator value\(n\) * and the element itself, and calculates the next accumulator value.\(n\) * \(n\) * \@sample

samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UShort>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}

/**
 * Returns a list containing successive accumulation values generated by applying
 * [operation] from left to right \(n\) * to each element, its index in the original array and current accumulator value that
 * starts with the first element of this array.\(n\) * \(n\) * Note that \`acc\` value passed to [operation] function should not be mutated;\(n\) * otherwise it would affect the previous value in resulting list.\(n\) * \(n\) * \@param [operation] function that takes the index of an element, current accumulator value\(n\) * and the element itself, and calculates the next accumulator value.\(n\) * \(n\) * \@sample

samples.collections.Collections.Aggregates.runningReduce

*/
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun

    if (isEmpty())
        return emptyList()
    var accumulator = this[0]
    val result = ArrayList<UShort>(size).apply {
        add(accumulator)
    }
    for (index in 1 until size) {
        accumulator = operation(index, accumulator, this[index])
        result.add(accumulator)
    }
    return result
}
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.scan(initial: R, operation: (acc: R, UInt) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. 

 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.scan(initial: R, operation: (acc: R, ULong) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. 

 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.scan(initial: R, operation: (acc: R, UByte) -> R): List<R> {
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. 

 * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.

 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return runningFold(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. 

 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UIntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. 

 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> ULongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. 

 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value. Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list. 

 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.

 * @sample samples.collections.Collections.Aggregates.scan
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return runningFoldIndexed(initial, operation)
}
resulting list.

* @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
  
  * @sample samples.collections.Aggregates.scan

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R> UByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {
    return runningFoldIndexed(initial, operation)
}

/**
 * Returns a list containing successive accumulation values generated by applying [operation] from left to right to each element, its index in the original array and current accumulator value that starts with [initial] value.
 * 
 * Note that `acc` value passed to [operation] function should not be mutated; otherwise it would affect the previous value in resulting list.
 * 
 * @param [operation] function that takes the index of an element, current accumulator value and the element itself, and calculates the next accumulator value.
  
  * @sample samples.collections.Aggregates.scan

*/
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return runningFoldIndexed(initial, operation)
}

@Deprecated("Use runningReduce instead.", ReplaceWith("runningReduce(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntArray.scanReduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): List<UInt> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.", ReplaceWith("runningReduceIndexed(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongArray.scanReduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): List<ULong> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.", ReplaceWith("runningReduceIndexed(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UByteArray.scanReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {
    return runningReduceIndexed(operation)
}

@Deprecated("Use runningReduceIndexed instead.", ReplaceWith("runningReduceIndexed(operation)"), level = DeprecationLevel.ERROR)
@SinceKotlin("1.3")
@ExperimentalStdlibApi
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    return runningReduceIndexed(operation)
}

/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.

Open Source Used In DNAC Platform Antelope 13998
*An@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sumBy(selector: (UInt) -> UInt): UInt {\n  var sum: UInt = 0u
  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sumBy(selector: (ULong) -> UInt): UInt {\n  var sum: UInt = 0u
  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.sumBy(selector: (UByte) -> UInt): UInt {\n  var sum: UInt = 0u
  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.sumBy(selector: (UShort) -> UInt): UInt {\n  var sum: UInt = 0u
  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sumByDouble(selector: (UInt) -> Double): Double {\n  var sum: Double = 0.0\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sumByDouble(selector: (ULong) -> Double): Double {\n  var sum: Double = 0.0\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")
@kotlin.jvm.JvmName("sumOfDouble")
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sumOf(selector: (UInt) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sumOf(selector: (ULong) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.sumOf(selector: (UByte) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.3")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType
@Suppress("INAPPLICABLE_JVM_NAME")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.sumOf(selector: (UShort) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UIntArray.sumOff(selector: (UInt) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.sumOff(selector: (ULong) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.sumOff(selector: (UByte) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/

*An@SinceKotlin("1.4")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress("INAPPLICABLE_JVM_NAME")\n@kotlin.jvm.JvmName("sumOfDouble")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.sumOff(selector: (UShort) -> Double): Double {\n  var sum: Double = 0.toDouble()\n  for (element in this) {\n    sum += selector(element)\n  }\n  return sum\n}\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n*/
*\n  @SinceKotlin("1.4")\n  @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n  @OverloadResolution\n  ByValLambdaReturnType\n  @Suppress("INAPPLICABLE_JVM_NAME")\n  @kotlin.jvm.JvmName("sumOfDouble")\n  @ExperimentalUnsignedTypes\n  @kotlin.internal.InlineOnly\n  public inline fun UShortArray.sumOf(selector: (UShort) -> Double): Double {
      var sum = Double(0.0)\n      for (element in this) {
          sum += selector(element)\n      }\n      return sum\n  }\n
** Returns the sum of all values produced by [selector] function applied to each element in the array.

*\n  @SinceKotlin("1.4")\n  @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n  @OverloadResolution\n  ByValLambdaReturnType\n  @Suppress("INAPPLICABLE_JVM_NAME")\n  @kotlin.jvm.JvmName("sumOfInt")\n  @ExperimentalUnsignedTypes\n  @kotlin.internal.InlineOnly\n  public inline fun UIntArray.sumOf(selector: (UInt) -> Int): Int {
      var sum = Int(0)\n      for (element in this) {
          sum += selector(element)\n      }\n      return sum\n  }\n
** Returns the sum of all values produced by [selector] function applied to each element in the array.

*\n  @SinceKotlin("1.4")\n  @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n  @OverloadResolution\n  ByValLambdaReturnType\n  @Suppress("INAPPLICABLE_JVM_NAME")\n  @kotlin.jvm.JvmName("sumOfInt")\n  @ExperimentalUnsignedTypes\n  @kotlin.internal.InlineOnly\n  public inline fun ULongArray.sumOf(selector: (ULong) -> Int): Int {
      var sum = Int(0)\n      for (element in this) {
          sum += selector(element)\n      }\n      return sum\n  }\n
** Returns the sum of all values produced by [selector] function applied to each element in the array.

*\n  @SinceKotlin("1.4")\n  @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n  @OverloadResolution\n  ByValLambdaReturnType\n  @Suppress("INAPPLICABLE_JVM_NAME")\n  @kotlin.jvm.JvmName("sumOfInt")\n  @ExperimentalUnsignedTypes\n  @kotlin.internal.InlineOnly\n  public inline fun UByteArray.sumOf(selector: (UByte) -> Int): Int {
      var sum = Int(0)\n      for (element in this) {
          sum += selector(element)\n      }\n      return sum\n  }\n
** Returns the sum of all values produced by [selector] function applied to each element in the array.

*\n  @SinceKotlin("1.4")\n  @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n  @OverloadResolution\n  ByValLambdaReturnType\n  @Suppress("INAPPLICABLE_JVM_NAME")\n  @kotlin.jvm.JvmName("sumOfLong")\n  @ExperimentalUnsignedTypes\n  @kotlin.internal.InlineOnly\n  public inline fun ULongArray.sumOf(selector: (ULong) -> Long): Long {
      var sum = Long(0)\n      for (element in this) {
          sum += selector(element)\n      }\n      return sum\n  }\n
** Returns the sum of all values produced by [selector] function applied to each element in the array.

*\n  @SinceKotlin("1.4")\n  @OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n  @OverloadResolution\n  ByValLambdaReturnType\n  @Suppress("INAPPLICABLE_JVM_NAME")\n  @kotlin.jvm.JvmName("sumOfLong")\n  @ExperimentalUnsignedTypes\n  @kotlin.internal.InlineOnly\n  public inline fun UByteArray.sumOf(selector: (UByte) -> Long): Long {
      var sum = Long(0)\n      for (element in this) {
          sum += selector(element)\n      }\n      return sum\n  }\n
** Returns the sum of all values produced by [selector] function applied to each element in the array.
*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@ Suppress("INAPPLICABLE_JVM_NAME")\@ kotlin.jvm.JvmName("sumOfLong")
\@ ExperimentalUnsignedTypes\@ kotlin.internal.InlineOnly\@ public inline fun UShortArray.sumOf(selector:
(UShort) -> Long): Long { \n \n var sum: Long = 0.toLong() \n \n for (element in this) { \n \n \n sum += selector(element) \n } \n return sum \n }\n/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@ Suppress("INAPPLICABLE_JVM_NAME")\@ kotlin.jvm.JvmName("sumOfUInt")
\@ ExperimentalUnsignedTypes\@ kotlin.internal.InlineOnly\@ public inline fun UIntArray.sumOf(selector:
(UInt) -> UInt): UInt { \n \n var sum: UInt = 0.toUInt() \n \n for (element in this) { \n \n \n sum += selector(element) \n } \n return sum \n }\n/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@ Suppress("INAPPLICABLE_JVM_NAME")\@ kotlin.jvm.JvmName("sumOfUInt")
\@ ExperimentalUnsignedTypes\@ kotlin.internal.InlineOnly\@ public inline fun ULongArray.sumOf(selector:
(ULong) -> ULong): ULong { \n \n var sum: ULong = 0.toULong() \n \n for (element in this) { \n \n \n sum += selector(element) \n } \n return sum \n }\n/**
 * Returns the sum of all values produced by [selector] function applied to each element in the array.
 */

*\@SinceKotlin("1.4")\@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\@OverloadResolution
ByLambdaReturnType\@ Suppress("INAPPLICABLE_JVM_NAME")\@ kotlin.jvm.JvmName("sumOfULong")
\@ ExperimentalUnsignedTypes\@ kotlin.internal.InlineOnly\@ public inline fun UByteArray.sumOf(selector:
(UByte) -> ULong): ULong { \n \n var sum: ULong = 0.toULong() \n \n for (element in this) { \n \n \n sum += selector(element) \n } \n return sum \n }
UShortArray.zip(other: Array<out R>, transform: (a: UShort, b: R) -> V): List<V> { 
    val size = minOf(size, other.size)
    val list = ArrayList<V>(size)
    for (i in 0 until size) {
        list.add(transform(this[i], other[i]))
    }
    return list
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 * @param samples.collections.Iterables.Operations.zipIterable
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UIntArray.zip(other: Iterable<R>): List<Pair<UInt, R>> { 
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 * @param samples.collections.Iterables.Operations.zipIterable
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> ULongArray.zip(other: Iterable<R>): List<Pair<ULong, R>> { 
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 * @param samples.collections.Iterables.Operations.zipIterable
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UByteArray.zip(other: Iterable<R>): List<Pair<UByte, R>> { 
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.
 * The returned list has length of the shortest collection.
 * @param samples.collections.Iterables.Operations.zipIterable
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public infix fun <R> UShortArray.zip(other: Iterable<R>): List<Pair<UShort, R>> { 
    return zip(other) { t1, t2 -> t1 to t2 }
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index.
 * The returned list has length of the shortest collection.
 * @param samples.collections.Iterables.Operations.zipIterableWithTransform
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, V> UIntArray.zip(other: Iterable<R>, transform: (a: UInt, b: R) -> V): List<V> { 
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index.
 * The returned list has length of the shortest collection.
 * @param samples.collections.Iterables.Operations.zipIterableWithTransform
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, V> ULongArray.zip(other: Iterable<R>, transform: (a: ULong, b: R) -> V): List<V> { 
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index.
 * The returned list has length of the shortest collection.
 * @param samples.collections.Iterables.Operations.zipIterableWithTransform
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun <R, V> UByteArray.zip(other: Iterable<R>, transform: (a: UByte, b: R) -> V): List<V> { 
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}

/**
 * Returns a list of values built from the elements of `this` array and the [other] collection with the same index.
 * The returned list has length of the shortest collection.
 * @param samples.collections.Iterables.Operations.zipIterableWithTransform
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
    val arraySize = size
    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))
    var i = 0
    for (element in other) {
        if (i >= arraySize) break
        list.add(transform(this[i++], element))
    }
    return list
}
returned list has length of the shortest collection.

@sample samples.collections.Operations.zipIterable

@SinceKotlin("1.3") @ExperimentalUnsignedTypes
fun <V> UIntArray.zip(other: UIntArray): List<Pair<UInt, UInt>>
val other = other.min
val list = ArrayList<Pair<UInt, UInt>>(minOf(this.size, other.size))
for (i in 0 until min)
list.add(zip(this[i], other[i]))
return list

@SinceKotlin("1.3") @ExperimentalUnsignedTypes
fun <V> ULongArray.zip(other: ULongArray): List<Pair<ULong, ULong>>
val other = other.min
val list = ArrayList<Pair<ULong, ULong>>(minOf(this.size, other.size))
for (i in 0 until min)
list.add(zip(this[i], other[i]))
return list

@SinceKotlin("1.3") @ExperimentalUnsignedTypes
fun <V> UByteArray.zip(other: UByteArray): List<Pair<UByte, UByte>>
val other = other.min
val list = ArrayList<Pair<UByte, UByte>>(minOf(this.size, other.size))
for (i in 0 until min)
list.add(zip(this[i], other[i]))
return list

@SinceKotlin("1.3") @ExperimentalUnsignedTypes
val other = other.min
val list = ArrayList<Pair<UShort, UShort>>(minOf(this.size, other.size))
for (i in 0 until min)
list.add(zip(this[i], other[i]))
return list

@kotlin.jvm.JvmName("sumOfUInt")
public fun <V> Array<out UInt>.sum(): UInt
val sum = this.fold(0u, { t1, t2 -> t1 + t2 })
return sum
Array<out ULong>.sum(): ULong {
    var sum: ULong = 0uL
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun Array<out UByte>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun Array<out UShort>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements in the array.
 */
public fun UIntArray.sum(): UInt {
    return storage.sum().toUInt()
}

/**
 * Returns the sum of all elements in the array.
 */
public fun ULongArray.sum(): ULong {
    return storage.sum().toULong()
}

/**
 * Returns the sum of all elements in the array.
 */
public inline fun UByteArray.sum(): UInt {
    return sumBy { it.toUInt() }
}

/**
 * Returns the sum of all elements in the array.
 */
public inline fun UShortArray.sum(): UInt {
    return sumBy { it.toUInt() }
}

/**
 * Returns the sum of all elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UInt>.toUIntArray(): UIntArray {
    val result = UIntArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns the sum of all elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<ULong>.toULongArray(): ULongArray {
    val result = ULongArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns the sum of all elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UByte>.toUByteArray(): UByteArray {
    val result = UByteArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns the sum of all elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UShort>.toUShortArray(): UShortArray {
    val result = UShortArray(size)
    var index = 0
    for (element in this)
        result[index++] = element
    return result
}

/**
 * Returns the sum of all elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UInt>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<ULong>.sum(): ULong {
    var sum: ULong = 0uL
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UByte>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}

/**
 * Returns the sum of all elements of this collection.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Collection<UShort>.sum(): UInt {
    var sum: UInt = 0u
    for (element in this) {
        sum += element
    }
    return sum
}
@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("UComparisonsKt")

package kotlin.comparisons

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

import kotlin.random.*

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
public fun maxOf(a: UInt, b: UInt): UInt {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
public fun maxOf(a: ULong, b: ULong): ULong {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
public fun maxOf(a: UByte, b: UByte): UByte {
    return if (a >= b) a else b
}

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.3")
public fun maxOf(a: UShort, b: UShort): UShort {
    return if (a >= b) a else b
}

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun maxOf(a: UInt, b: UInt, c: UInt): UInt {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun maxOf(a: ULong, b: ULong, c: ULong): ULong {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun maxOf(a: UByte, b: UByte, c: UByte): UByte {
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the greater of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
    return maxOf(a, maxOf(b, c))
}

/**
 * Returns the lesser of two values.
 */
@SinceKotlin("1.3")
public fun minOf(a: UInt, b: UInt): UInt {
    return if (a <= b) a else b
}

/**
 * Returns the lesser of two values.
 */
@SinceKotlin("1.3")
public fun minOf(a: ULong, b: ULong): ULong {
    return if (a <= b) a else b
}

/**
 * Returns the lesser of two values.
 */
@SinceKotlin("1.3")
public fun minOf(a: UByte, b: UByte): UByte {
    return if (a <= b) a else b
}

/**
 * Returns the lesser of two values.
 */
@SinceKotlin("1.3")
public fun minOf(a: UShort, b: UShort): UShort {
    return if (a <= b) a else b
}

/**
 * Returns the lesser of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun minOf(a: UInt, b: UInt, c: UInt): UInt {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the lesser of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun minOf(a: ULong, b: ULong, c: ULong): ULong {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the lesser of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public inline fun minOf(a: UByte, b: UByte, c: UByte): UByte {
    return minOf(a, minOf(b, c))
}

/**
 * Returns the lesser of three values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
    return minOf(a, minOf(b, c))
}

/**
 * Returns the lesser of the given values.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UInt, vararg other: UInt): UInt {
    var max = a
    for (e in other) max = minOf(max, e)
    return max
}

/**
 * Returns the lesser of the given values.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: ULong, vararg other: ULong): ULong {
    var max = a
    for (e in other) max = minOf(max, e)
    return max
}

/**
 * Returns the lesser of the given values.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UByte, vararg other: UByte): UByte {
    var max = a
    for (e in other) max = minOf(max, e)
    return max
}

/**
 * Returns the lesser of the given values.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UShort, vararg other: UShort): UShort {
    var max = a
    for (e in other) max = minOf(max, e)
    return max
}
UInt, vararg other: UInt): UInt {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UInt, vararg other: UInt): UInt {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UByte, vararg other: UByte): UByte {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/**
 * Returns the smaller of the given values.
 */
@SinceKotlin("1.4")
@ExperimentalUnsignedTypes
public fun minOf(a: UShort, vararg other: UShort): UShort {
    var min = a
    for (e in other) min = minOf(min, e)
    return min
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("URangesKt")

package kotlin.ranges

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun UIntRange.random(): UInt {
    return random(Random)
}

/**
 * Returns a random element from this range.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun ULongRange.random(): ULong {
    return random(Random)
}

/**
 * Returns a random element from this range using the specified source of randomness.
 * 
 * @throws IllegalArgumentException if this range is empty.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntRange.random(random: Random): UInt {
    try {
        return random.nextUInt(this)
    } catch(e: IllegalArgumentException) {
        throw NoSuchElementException(e.message)
    }
}

/**
 * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public fun UIntRange.randomOrNull(): UInt? {
    return randomOrNull(Random)
}

/**
 * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@ExperimentalUnsignedTypes
public fun ULongRange.randomOrNull(): ULong? {
    if (isEmpty())
        return null
    return random.nextULong(this)
}

/**
 * Returns `true` if this range contains the specified [element].
 * 
 * Always returns `false` if the [element] is `null`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun UIntRange.contains(element: UInt?): Boolean {
    return element != null && contains(element)
}

/**
 * Returns `true` if this range contains the specified [element].
 * 
 * Always returns `false` if the [element] is `null`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun ULongRange.contains(element: ULong?): Boolean {
    return element != null &&

Open Source Used In DNAC Platform Antelope 14007
contains(value: UByte): Boolean { return contains(value.toUInt()) } * Checks if the specified [value] belongs to this range.

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3") @ExperimentalUnsignedTypes public operator fun UIntRange.contains(value: UByte): Boolean {
    return contains(value.toUInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3") @ExperimentalUnsignedTypes public operator fun ULongRange.contains(value: ULong): Boolean {
    return contains(value.toULong())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3") @ExperimentalUnsignedTypes public operator fun UIntRange.contains(value: ULong): Boolean {
    return contains(value.toUInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3") @ExperimentalUnsignedTypes public operator fun UIntRange.contains(value: UShort): Boolean {
    return contains(value.toUInt())
}

/**
 * Checks if the specified [value] belongs to this range.
 */
@SinceKotlin("1.3") @ExperimentalUnsignedTypes public operator fun ULongRange.contains(value: UShort): Boolean {
    return contains(value.toULong())
}

/**
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
 * Returns a progression from this value down to the specified [to] value with the step -1.
value.\n * @\n
 If the [to] value is less than or equal to \`this\` value, then the returned range is empty.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun UInt.until(to: UInt): UIntRange {
    if (to <= UInt.MIN_VALUE) return UIntRange.EMPTY\n    return this .. (to - 1u).toUInt()
}

/**
 * Returns a range from this value up to but excluding the specified \[to\] value.\n * @\n * If the \[to\] value is less than or equal to \`this\` value, then the returned range is empty.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic infix fun ULong.until(to: ULong): ULongRange {
    if (to <= ULong.MIN_VALUE) return ULongRange.EMPTY\n    return this.toULong() .. (to - 1u).toULong()
}

/**
 * Ensures that this value is not less than the specified [minimumValue].\n * @\n * if it's greater than or equal to the [minimumValue] or the [maximumValue] otherwise.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UInt.coerceAtLeast(minimumValue: UInt): UInt {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue].\n * @\n * if it's less than or equal to the [maximumValue] or the [minimumValue] otherwise.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UInt.coerceAtMost(maximumValue: UInt): UInt {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue].\n * @\n * if it's less than or equal to the [maximumValue] or the [minimumValue] otherwise.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByte.coerceAtMost(maximumValue: UByte): UByte {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue].\n * @\n * if it's less than or equal to the [maximumValue] or the [minimumValue] otherwise.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShort.coerceAtMost(maximumValue: UShort): UShort {
    return if (this > maximumValue) maximumValue else this
}

/**
 * Ensures that this value is not less than the specified [minimumValue].\n * @\n * if it's greater than or equal to the [minimumValue] or the [maximumValue] otherwise.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByte.coerceAtLeast(minimumValue: UByte): UByte {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not less than the specified [minimumValue].\n * @\n * if it's greater than or equal to the [minimumValue] or the [maximumValue] otherwise.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShort.coerceAtLeast(minimumValue: UShort): UShort {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue].\n * @\n * if it's less than or equal to the [maximumValue] or the [minimumValue] otherwise.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UByte.coerceAtLeast(maximumValue: UByte): UByte {
    return if (this < minimumValue) minimumValue else this
}

/**
 * Ensures that this value is not greater than the specified [maximumValue].\n * @\n * if it's less than or equal to the [maximumValue] or the [minimumValue] otherwise.\n
 */
@SinceKotlin("1.3")\n@ExperimentalUnsignedTypes\npublic fun UShort.coerceAtLeast(minimumValue: UShort): UShort {
    return if (this < minimumValue) minimumValue else this
}
```kotlin
/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * public fun UShort.coerceAtMost(maximumValue: UShort) {  
 *     return if (this > maximumValue) this else maximumValue
 * } /* Ensures that this value lies in the specified range [minimumValue]..[maximumValue]. */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * public fun Uint.coerceIn(minimumValue: Uint, maximumValue: Uint) {  
 *     if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
 *     if (this < minimumValue) return minimumValue
 *     if (this > maximumValue) return maximumValue
 * } /* Ensures that this value lies in the specified range [minimumValue]..[maximumValue]. */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * public fun ULong.coerceIn(minimumValue: ULong, maximumValue: ULong) {  
 *     if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
 *     if (this < minimumValue) return minimumValue
 *     if (this > maximumValue) return maximumValue
 * } /* Ensures that this value lies in the specified range [minimumValue]..[maximumValue]. */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * public fun UByte.coerceIn(minimumValue: UByte, maximumValue: UByte) {  
 *     if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
 *     if (this < minimumValue) return minimumValue
 *     if (this > maximumValue) return maximumValue
 * } /* Ensures that this value lies in the specified range [minimumValue]..[maximumValue]. */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * public fun UShort.coerceIn(minimumValue: UShort, maximumValue: UShort) {  
 *     if (minimumValue > maximumValue) throw IllegalArgumentException("Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.")
 *     if (this < minimumValue) return minimumValue
 *     if (this > maximumValue) return maximumValue
 * } /* Ensures that this value lies in the specified range [minimumValue]..[maximumValue]. */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * public fun UInt.coerceIn(range: ClosedRange<UInt>): UInt {  
 *     if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
 *     return when {
 *         this < range.start -> range.start
 *         this > range.endInclusive -> range.endInclusive
 *         else -> this
 *     }
 * } /* Ensures that this value lies in the specified range [range]. */

/*
 * @SinceKotlin("1.3")
 * @ExperimentalUnsignedTypes
 * public fun ULong.coerceIn(range: ClosedRange<ULong>): ULong {  
 *     if (range.isEmpty()) throw IllegalArgumentException("Cannot coerce value to an empty range: $range.")
 *     return when {
 *         this < range.start -> range.start
 *         this > range.endInclusive -> range.endInclusive
 *         else -> this
 *     }
 * } /* Ensures that this value lies in the specified range [range]. */
```
range.endInclusive -> range.endInclusive

else -> this

}"*/

* * * * *

* Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* *

* Note: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt file. See:

https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

* *

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* *

* *

* *

* *

* *

* *

* *

* *
DeprecationLevel.ERROR

public expect open class NoWhenBranchMatchedException : RuntimeException {

    constructor()
    constructor(message: String?)
    constructor(message: String?, cause: Throwable?)
}

@Deprecated("This exception type is not supposed to be thrown or caught in common code and will be removed from kotlin-stdlib-common soon.", level = DeprecationLevel.ERROR)
public expect class UninitializedPropertyAccessException : RuntimeException {

    constructor()
    constructor(message: String?)
    constructor(message: String?, cause: Throwable?)
    constructor(cause: Throwable?)
}

/**
 * Thrown after invocation of a function or property that was expected to return `Nothing`, but returned something instead.
 */
@SinceKotlin("1.4.")
public
internal class KotlinNothingValueException : RuntimeException {

    constructor() : super()
    constructor(message: String?) : super(message)
    constructor(message: String?, cause: Throwable?) : super(message, cause)
}

/**
 * Returns the detailed description of this throwable with its stack trace.
 * @return The detailed description includes:
 * - the short description (see [Throwable.toString]) of this throwable;
 * - the complete stack trace;
 * - detailed descriptions of the exceptions that were suppressed
 * in order to deliver this exception;
 * - the detailed description of each throwable in the [Throwable.cause] chain.
 */
@SinceKotlin("1.4.")
public expect fun Throwable.stackTraceToString(): String

/**
 * Prints the detailed description [Throwable.stackTraceToString] of this throwable to the standard output or standard error output.
 */
@SinceKotlin("1.4.")
public expect fun Throwable.printStackTrace(): Unit

/**
 * When supported by the platform, adds the specified exception to the list of exceptions that were suppressed in order to deliver this exception.
 */
@SinceKotlin("1.4.")
public expect fun Throwable.addSuppressed(exception: Throwable)

/**
 * Returns a list of all exceptions that were suppressed in order to deliver this exception.
 */
@SinceKotlin("1.4.")
public expect val Throwable.suppressedExceptions: List<Throwable>
Classes that inherit from this interface can be represented as a sequence of elements that can be iterated over. The iterator is covariant in its element type. Classes that inherit from this interface can be represented as a sequence of elements that can be iterated over and that supports removing elements during iteration. The mutable iterator is invariant in its element type.

```kotlin
public interface List<out E> : Collection<E> {

    // Access Operations
    public fun contains(element: @UnsafeVariance E): Boolean
    public fun indexOf(element: @UnsafeVariance E): Int
    public fun get(index: Int): E

    // Search Operations
    public fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean
    public fun retainAll(elements: Collection<E>): Boolean
    public fun remove(element: E): Boolean
    public fun add(element: E): Boolean

    // Bulk Operations
    public fun addAll(elements: Collection<E>): Boolean
    public fun removeAll(elements: Collection<E>): Boolean
    public fun retainAll(elements: Collection<E>): Boolean
    public fun removeAll(elements: Collection<E>): Boolean

    // Positional Access
    override val size: Int
    override fun isEmpty(): Boolean
    override fun iterator(): Iterator<E>
}
```

Classes that support removing elements also support adding elements.

```kotlin
override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean
```

The collection is covariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

The list is invariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

Classes that support removing elements also support adding elements.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

The list is invariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

Classes that support removing elements also support adding elements.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

The list is invariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

Classes that support removing elements also support adding elements.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

The list is invariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

Classes that support removing elements also support adding elements.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

The list is invariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

Classes that support removing elements also support adding elements.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

The list is invariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

Classes that support removing elements also support adding elements.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

The list is invariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

Classes that support removing elements also support adding elements.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

The list is invariant in its element type. The list is covariant in its element type.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```

Classes that support removing elements also support adding elements.

```kotlin
public interface MutableList<@UnsafeVariance E> : List<@UnsafeVariance E> {

    override fun contains(element: @UnsafeVariance E): Boolean
    override fun get(index: Int): @UnsafeVariance E
}
```
lastIndexOf(element: @UnsafeVariance E): Int
    // List Iterators
    /**
     * Returns a list iterator over the elements in this list (in proper sequence).
     */
    public fun listIterator(): ListIterator<@UnsafeVariance E>
    /**
     * Returns a list iterator over the elements in this list (in proper sequence), starting at the specified [index].
     */
    public fun listIterator(index: Int): ListIterator<@UnsafeVariance E>

    // View
    /**
     * Returns a view of the portion of this list between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).
     */
    public fun subList(fromIndex: Int, toIndex: Int): SubList<@UnsafeVariance E, @UnsafeVariance E>

    // Positional Access Operations
    /**
     * Returns `true` if the list is empty.
     */
    public fun isEmpty(): Boolean

    // List Iterators
    /**
     * Returns a list iterator over the elements in this list (in proper sequence).
     */
    public fun listIterator(): ListIterator<@UnsafeVariance E>
    /**
     * Returns a list iterator over the elements in this list (in proper sequence), starting at the specified [index].
     */
    public fun listIterator(index: Int): ListIterator<@UnsafeVariance E>

    // View
    /**
     * Returns a view of the portion of this list between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).
     */
    public fun subList(fromIndex: Int, toIndex: Int): SubList<@UnsafeVariance E, @UnsafeVariance E>

    // Positional Access Operations
    /**
     * Returns `true` if the list is empty.
     */
    public fun isEmpty(): Boolean

    // List Iterators
    /**
     * Returns a list iterator over the elements in this list (in proper sequence).
     */
    public fun listIterator(): ListIterator<@UnsafeVariance E>
    /**
     * Returns a list iterator over the elements in this list (in proper sequence), starting at the specified [index].
     */
    public fun listIterator(index: Int): ListIterator<@UnsafeVariance E>

    // View
    /**
     * Returns a view of the portion of this list between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).
     */
    public fun subList(fromIndex: Int, toIndex: Int): SubList<@UnsafeVariance E, @UnsafeVariance E>
(contains no elements), "false" otherwise.

public fun isEmpty(): Boolean

public fun containsKey(key: K): Boolean

public fun containsValue(value: V): Boolean

Returns the value corresponding to the given [key], or `null` if such a key is not present in the map.

public fun get(key: K): V?

Returns the value corresponding to the given [key], or [defaultValue] if such a key is not present in the map.

@UnsafeVariance V

public fun getOrDefault(key: K, defaultValue: V): V

Returns the value corresponding to the given [key], or the specified [defaultValue] if such a key is not present in the map.

public fun put(key: K, value: V): V?

Updates this map with key/value pairs from the specified map [from].

public fun clear(): Unit

Removes all elements from this map.

public fun remove(key: K): V?

Removes the entry for the specified key only if it is mapped to the specified value.

public fun remove(key: K, value: V): Boolean

Removes the entry for the specified key only if it is mapped to the specified value.

public fun containsValue(value: V): Boolean

Returns `true` if the map contains one or more keys to the specified [value].

public fun containsKey(key: K): Boolean

Returns `true` if the map contains the specified [key].

public fun putAll(map: Map<K, V>): Unit

Updates this map with key/value pairs from the specified map [from].

public fun remove(key: K): V?

Removes the entry for the specified key only if it is mapped to the specified value.

public fun containsKey(key: K): Boolean

Returns `true` if the map contains the specified [key].
*An abstract class LongIterator : Iterator<Long> {
    override final fun next() = nextLong()
    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextLong(): Long
}

An iterator over a sequence of values of type `Float`.
*An abstract class FloatIterator : Iterator<Float> {
    override final fun next() = nextFloat()
    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextFloat(): Float
}

An iterator over a sequence of values of type `Double`.
*An abstract class DoubleIterator : Iterator<Double> {
    override final fun next() = nextDouble()
    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextDouble(): Double
}

An iterator over a sequence of values of type `Boolean`.
*An abstract class BooleanIterator : Iterator<Boolean> {
    override final fun next() = nextBoolean()
    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextBoolean(): Boolean
}

An iterator over a progression of values of type `Char`.
* @property step the number by which the value is incremented on each step.
* An iterator over a progression of values of type `Int`.
* @property step the number by which the value is incremented on each step.
* An iterator over a progression of values of type `Long`.
* @property step the number by which the value is incremented on each step.
The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step]. In order to go backwards the [step] must be negative. [step] must be greater than `Long.MIN_VALUE` and not equal to zero. 

A progression of values of type `Long` open class LongProgression internal constructor {

    * The first element in the progression.

    }
}

/**
 * A progression of values of type `Int`.
 */
public class IntProgression

    companion object {
        /**
         * Creates IntProgression within the specified bounds of a closed range.

         * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].

        */
    }

    /**
     * The last element in the progression.
     */
    public val last: Int = endInclusive

    /**
     * The step of the progression.
     */
    public val step: Int = step

    override fun iterator(): IntIterator =

        IntProgressionIterator(first, last, step)

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean =

        if (step == 0) start > last else start < last

    override fun equals(other: Any?): Boolean =

        if (other is LongProgression && isEmpty() && other.isEmpty()) {
            first == other.first && last == other.last &&
            step == other.step
        } else {
            getProgressionLastElement(start, toLong(), endInclusive.toLong(), step).toInt() ==
            other.getProgressionLastElement(start, toLong(), endInclusive.toLong(), step).toInt()
        }

    override fun hashCode(): Int =

        if (isEmpty()) -1 else 31 * (31 * (first xor (first ushr 32)) + (last xor (last ushr 32))) + (step xor (step ushr 32)).toInt()

    companion object {
        /**
         * Creates IntProgression within the specified bounds of a closed range.

         * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].

        */
    }

    /**
     * The first element in the progression.
     */
    public val first: Int = start

    /**
     * The step of the progression.
     */
    public val step: Int = step

    override fun iterator(): IntIterator =

        IntProgressionIterator(first, last, step)

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean =

        if (step == 0) first > last else first < last

    override fun equals(other: Any?): Boolean =

        if (other is IntProgression && isEmpty() && other.isEmpty()) {
            first == other.first && last == other.last &&
            step == other.step
        } else {
            getProgressionLastElement(start, toInt(), endInclusive.toInt(), step).toInt() ==
            other.getProgressionLastElement(start, toInt(), endInclusive.toInt(), step).toInt()
        }

    override fun hashCode(): Int =

        if (isEmpty()) -1 else 31 * (31 * first + last) + (step xor (step ushr 32)).toInt()

    companion object {
        /**
         * Creates IntProgression within the specified bounds of a closed range.

         * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].

        */
    }

    /**
     * The first element in the progression.
     */
    public val first: Int = start

    /**
     * The step of the progression.
     */
    public val step: Int = step

    override fun iterator(): IntIterator =

        IntProgressionIterator(first, last, step)

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean =

        if (step == 0) first > last else first < last

    override fun equals(other: Any?): Boolean =

        if (other is IntProgression && isEmpty() && other.isEmpty()) {
            first == other.first && last == other.last &&
            step == other.step
        } else {
            getProgressionLastElement(start, toInt(), endInclusive.toInt(), step).toInt() ==
            other.getProgressionLastElement(start, toInt(), endInclusive.toInt(), step).toInt()
        }

    override fun hashCode(): Int =

        if (isEmpty()) -1 else 31 * (31 * first + last) + (step xor (step ushr 32)).toInt()

    companion object {
        /**
         * Creates IntProgression within the specified bounds of a closed range.

         * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].

        */
    }

    /**
     * The first element in the progression.
     */
    public val first: Char = rangeStart

    /**
     * The step of the progression.
     */
    public val step: Int = step

    override fun iterator(): CharIterator =

        CharProgressionIterator(first, last, step)

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean =

        if (step == 0) first > last else first < last

    override fun equals(other: Any?): Boolean =

        if (other is CharProgression && isEmpty() && other.isEmpty()) {
            first == other.first && last == other.last &&
            step == other.step
        } else {
            getProgressionLastElement(start, toChar(), endInclusive.toChar(), step).toChar() ==
            other.getProgressionLastElement(start, toChar(), endInclusive.toChar(), step).toChar()
        }

    override fun hashCode(): Int =

        if (isEmpty()) -1 else 31 * (31 * first + last) + (step xor (step ushr 32)).toInt()

    companion object {
        /**
         * Creates CharProgression within the specified bounds of a closed range.

         * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].

        */
    }

    /**
     * The first element in the progression.
     */
    public val first: Char = rangeStart

    /**
     * The step of the progression.
     */
    public val step: Int = step

    override fun iterator(): CharIterator =

        CharProgressionIterator(first, last, step)

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean =

        if (step == 0) first > last else first < last

    override fun equals(}
Boolean = value <=> start && value <= endInclusive

/**
 * Checks whether the range is empty.
 */

public fun isEmpty(): Boolean = start > endInclusive

/*
 * Copyright 2010-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 */

package kotlin

/**
 * The type with only one value: the `Unit` object. This type corresponds to the `void` type in Java.
 */
public object Unit {
    override fun toString() = "kotlin.Unit"
}

/*
 * Copyright 2010-2015 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 */

package kotlin.annotation

import kotlin.annotation.AnnotationTarget.*

/**
 * Contains the list of code elements which are the possible annotation targets
 */
public enum class AnnotationTarget {
    /**
     * Class, interface or object, annotation class is also included
     */
    CLASS,
    /**
     * Property
     */
    PROPERTY,
    /** Field, including property's backing field
     */
    FIELD,
    /** Local variable
     */
    LOCAL_VARIABLE,
    /** Constructor only (primary or secondary)
     */
    CONSTRUCTOR,
    /** Property getter only
     */
    PROPERTY_GETTER,
    /** Type usage
     */
    TYPE,
    /** Any expression
     */
    EXPRESSION,
    /** File
     */
    FILE,
    /** Type alias
     */
    TYPEALIAS
}

/**
 * Contains the list of possible annotation's retentions.
 */
public enum class AnnotationRetention {
    /**
     * Annotation isn't stored in binary output
     */
    SOURCE,
    /**
     * Annotation is stored in binary output, but invisible for reflection
     */
    BINARY,
    /**
     * Annotation is stored in binary output and visible for reflection (default retention)
     */
    RUNTIME
}

/**
 * This meta-annotation indicates the kinds of code elements which are possible targets of an annotation.
 */
@Target(AnnotationTarget.ANNOTATION_CLASS)
@MustBeDocumented
public annotation class Target(vararg val allowedTargets: AnnotationTarget)

/**
 * This meta-annotation determines whether an annotation is applicable twice or more on a single code element
 */
@Target(AnnotationTarget.ANNOTATION_CLASS)
public annotation class Repeatable

/**
 * This meta-annotation determines that an annotation is a part of public API and therefore should be included in the generated documentation for the element to which the annotation is applied.
 */
@Target(AnnotationTarget.ANNOTATION_CLASS)
@MustBeDocumented
public annotation class Repeatable

/*
 * Copyright 2010-2016 JetBrains s.r.o.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 */
software * distributed under the License is distributed on an "AS IS" BASIS, either express or implied. * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

For example, MutableMap.remove(K, V) available if corresponding * method 'java/util/Map.remove(Ljava/lang/Object;Ljava/lang/Object;Z) is defined in JDK (i.e. for major versions >= 8)

* Use of this source code is governed by the Apache 2.0 license that can be found in the LICENSE.txt file.

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 */

@JsName("arrayIterator")
internal fun arrayIterator(array: dynamic, type: String?) = when (type) {
    null -> {
        val arr: Array<dynamic> = array
        object : Iterator<dynamic> {
            var index = 0
            override fun hasNext() = index < arr.size
            override fun next() = if (index < arr.size) arr[index++] else throw NoSuchElementException("$index")
        }
    }
    "BooleanArray" -> booleanArrayIterator(array)
    "ByteArray" -> byteArrayIterator(array)
    "ShortArray" -> shortArrayIterator(array)
    "CharArray" -> charArrayIterator(array)
    "IntArray" -> intArrayIterator(array)
    "LongArray" -> longArrayIterator(array)
    "FloatArray" -> floatArrayIterator(array)
    "DoubleArray" -> doubleArrayIterator(array)
    else -> throw IllegalArgumentException("Unsupported type argument for arrayIterator: $type")
}

@JsName("booleanArrayIterator")
internal fun booleanArrayIterator(array: BooleanArray) =
result.length += collection.size
copyArrayType(array, result)

var index: Int = array.length
for (element in collection)
result[index++] = element

return result
}

internal fun <T> fillFromCollection(dst: dynamic, startIndex: Int, collection: Collection<T>): dynamic {
var index = startIndex
for (element in collection)
dst[index++] = element
return dst
}

internal inline fun copyArrayType(from: dynamic, to: dynamic) {
if (from.`$type$` !== undefined) {
    to.`$type$` = from.`$type$`
}
}

internal inline fun jsIsType(obj: dynamic, jsClass: dynamic) = js("Kotlin").isType(obj, jsClass)

package kotlin

import kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED

@SinceKotlin("1.3")
@JsName("CoroutineImpl")
internal abstract class CoroutineImpl(private val resultContinuation: Continuation<Any?>) : Continuation<Any?> {
    protected var state = 0
    protected var exceptionState = 0
    protected var result: Any? = null
    protected var exception: Throwable? = null
    protected var finallyPath: Array<Int>? = null

    public override val context: CoroutineContext = resultContinuation.context

    private var intercepted_: Continuation<Any?>? = null

    public fun intercepted(): Continuation<Any?> =
        intercepted_
            ?: (context[ContinuationInterceptor]?.interceptContinuation(this) ?: this).also { intercepted_ = it }

    override fun resumeWith(result: Result<Any?>) {
        var current = this
        var currentResult: Any? = result.getOrNull()
        var currentException: Throwable? = result.exceptionOrNull()

        while (true) {
            with(current) {
                val completion = resultContinuation

                if (currentException == null) {
                    this.result = currentResult
                } else {
                    state = exceptionState
                    exception = currentException
                }

                try {
                    val outcome = doResume()
                    if (outcome === COROUTINE_SUSPENDED)
                        return
                    currentResult = outcome
                    currentException = null
                } catch (exception: dynamic) {
                    currentResult = null
                    currentException = exception.unsafeCast<Throwable>()
                }

                releaseIntercepted() // this state machine instance is terminating

                if (completion is CoroutineImpl) {
                    current = completion
                } else {
                    // top-level completion reached -- invoke and return
                    currentException?.let {
                        completion.resumeWithException(it)
                    } ?: completion.resume(currentResult)
                    return
                }
            }
        }
    }

    private fun releaseIntercepted() {
        val intercepted = intercepted_
        if (intercepted != null) {
            context[ContinuationInterceptor]!!.releaseInterceptedContinuation(intercepted)
        }
        this.intercepted_ = CompletedContinuation // just in case
    }

    protected abstract fun doResume(): Any?
}

internal object CompletedContinuation : Continuation<Any?> {
    override val context: CoroutineContext
        get() = error("This continuation is already complete")

    override fun resumeWith(result: Result<Any?>) {
        error("This continuation is already complete")
    }

    override fun toString(): String = "This continuation is already complete"
}

@file:Suppress("UNCHECKED_CAST", "RedundantVisibilityModifier")

package kotlin

import kotlin.contracts.*
import kotlin.internal.InlineOnly
import kotlin.jvm.JvmField
import kotlin.jvm.JvmName

/**
 * A discriminated union that encapsulates a successful outcome with a value of type [T] or a failure with an arbitrary [Throwable]
 */
@Suppress("NON_PUBLIC_PRIMARY_CONSTRUCTOR_OF_INLINE_CLASS")
public inline class Result<out T> @PublishedApi internal constructor(value: Any?) {
    // discovery

    /**
     * Returns `true` if this instance represents a successful outcome.
     */
    val isSuccess: Boolean
        get() = value is Success

    /**
     * Returns `false` if this instance represents a failed outcome.
     */
    val isFailure: Boolean
        get() = value is Failure

    // inline

    inline operator fun invoke() = value

    inline fun <reified R> as(): R
        where R : Any = value

    inline fun <T, R> transform(transformer: (T) -> R): Result<R> = when (this)
        where this is Success
            -> Success(transformer.invoke(value))
        else
            -> Failure(transformer.invoke(exception))

    inline fun <T> fold(failure: (T) -> R, success: (T) -> R): R
        where R : Any = when (this)
            where this is Success
                -> success.invoke(value)
            else
                -> failure.invoke(exception)

    inline fun <T> foldIf(failure: (T) -> Unit, success: (T) -> Unit): Unit
        where T : Any = when (this)
            where this is Success
                -> success.invoke(value)
            else
                -> failure.invoke(exception)

    // capture lambda

    inline fun <T> capture(f: (T) -> Unit): Resultlime<
        where T : Any = when (this)
            where this is Success
                -> Success(f.invoke(value))
            else
                -> Failure(f.invoke(exception))

    // check lambda

    inline fun <T> check(f: (T) -> Unit): Result<
        where T : Any = when (this)
            where this is Success
                -> Success(f.invoke(value))
            else
                -> Failure(f.invoke(exception))

    // run lambda

    inline fun <R> run(f: (T) -> R): R
        where R : Any = when (this)
            where this is Success
                -> success.invoke(value)
            else
                -> failure.invoke(exception)

    // implicit conversion

    fun <R> as(): R
        where R : Any = value

    // specializations

    @PublishedApi
    inline fun <T : Any> Success(value: T): T
    {"Open Source Used In DNAC Platform Antelope 14022"}
This document contains a large amount of code, primarily in Java, with comments and annotations. The code appears to be related to a library or framework for handling asynchronous operations and exceptions. The comments and annotations are in English, and the code seems to be implementing or extending functionality related to the 'result' pattern, which is a common pattern in functional programming languages.

The code includes functions and methods for handling results of computations, including success, failure, and error handling. It seems to be part of a library that provides a way to handle and propagate errors in a functional manner, possibly for frameworks that support asynchronous operations.

The specific features include:
- A class `Result` that encapsulates values, with methods to handle success, failure, and exceptions.
- Functions to handle exceptions, such as `getOrElse` and `getOrThrow`, which return values or throw exceptions based on the success or failure of the computation.
-Generics and type safety are used extensively to ensure that the operations are type-safe and can handle various types of results.

Overall, this code is part of a larger library or framework designed to handle asynchronous operations in a functional and error-safe manner.
function is a shorthand for `getOrElse { defaultValue }` (see [getOrElse]).

```kotlin
@InlineOnly
@SinceKotlin("1.3")
public inline fun <R, T : R> Result<T>.getOrElse(defaultValue: R): R {
    if (isFailure) return defaultValue
    return value as T
}
```

*/

```kotlin
/**
 * Returns the result of [transform] function applied to the encapsulated value
 * or the *original encapsulated [Throwable] exception if it is [failure][Result.isFailure].
 * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.
 */
@InlineOnly
@SinceKotlin("1.3")
public inline fun <R, T> Result<T>.fold(onSuccess: (value: T) -> R, onFailure: (exception: Throwable) -> R): R {
    return when (val exception = exceptionOrNull()) {
        null -> onSuccess(value as T)
        else -> onFailure(exception)
    }
}
```

```kotlin
/**
 * Returns the encapsulated result of the given [transform] function applied to the encapsulated value
 * or the *original encapsulated [Throwable] exception if it is [failure][Result.isFailure].
 * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.
 */
@InlineOnly
@SinceKotlin("1.3")
public inline fun <R, T> Result<T>.map(transform: (value: T) -> R): Result<R> {
    return when {
        isSuccess -> Result.success(transform(value as T))
        else -> Result(value)
    }
}
```

```kotlin
/**
 * Returns the encapsulated result of the given [transform] function applied to the encapsulated value
 * or the *original encapsulated [Throwable] exception if it is [failure][Result.isFailure].
 * Note, that this function rethrows any [Throwable] exception thrown by [transform] function.
 */
@InlineOnly
@SinceKotlin("1.3")
public inline fun <R, T : R> Result<T>.recover(transform: (exception: Throwable) -> R): Result<R> {
    return when (val exception = exceptionOrNull()) {
        null -> this
        else -> Result.success(transform(exception))
    }
}
```

```kotlin
/**
 * Performs the given [action] on the encapsulated [Throwable] exception if this instance represents
 * [failure][Result.isFailure]. Returns the original `Result` unchanged.
 */
@InlineOnly
@SinceKotlin("1.3")
public inline fun <T> Result<T>.onFailure(action: (exception: Throwable) -> Unit): Result<T> {
    return when (val exception = exceptionOrNull()) {
        null -> this
        else -> runCatching { transform(exception) }
    }
}
```

```kotlin
/**
 * Performs the given [action] on the encapsulated [Throwable] exception if this instance represents
 * [failure][Result.isFailure]. Returns the original `Result` unchanged.
 */
@InlineOnly
@SinceKotlin("1.3")
public inline fun <T> Result<T>.onSuccess(action: (value: T) -> Unit): Result<T> {
    return when (val exception = exceptionOrNull()) {
        null -> this
        else -> runCatching { transform(exception) }
    }
}
```

```kotlin
// -------------------
```
s.r.o. and Kotlin Programming Language contributors. Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

---

```kotlin
package kotlin.coroutines

import kotlin.contracts.*
import kotlin.coroutines.intrinsics.*
import kotlin.internal.InlineOnly

/**
 * Interface representing a continuation after a suspension point that returns a value of type `T`.
 */
@SinceKotlin("1.3")
public interface Continuation<in T> {
    /**
     * The context of the coroutine that corresponds to this continuation.
     */
    public val context: CoroutineContext

    /**
     * Resumes the execution of the corresponding coroutine passing a successful or failed [result] as the value of the last suspension point.
     */
    public fun resumeWith(result: Result<T>)
}

/**
 * Classes and interfaces marked with this annotation are restricted when used as receivers for extension `suspend` functions. These `suspend` extensions can only invoke other member or extension `suspend` functions on this particular receiver and are restricted from calling arbitrary suspension functions.
 */
@SinceKotlin("1.3")
@Target(AnnotationTarget.CLASS)
@Retention(AnnotationRetention.BINARY)
public annotation class RestrictsSuspension

/**
 * Resumes the execution of the corresponding coroutine passing [value] as the return value of the last suspension point.
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun <T> Continuation<T>.resume(value: T): Unit =
    resumeWith(Result.success(value))

/**
 * Resumes the execution of the corresponding coroutine so that the [exception] is re-thrown right after the last suspension point.
 */
@SinceKotlin("1.3")
@InlineOnly
public inline fun <T> Continuation<T>.resumeWithException(exception: Throwable): Unit =
    resumeWith(Result.failure(exception))

/**
 * Creates a [Continuation] instance with the given [context] and implementation of [resumeWith] method.
 */
@SinceKotlin("1.3")
public inline fun <T>
    Continuation(
        context: CoroutineContext,
        crossinline resumeWith: (Result<T>) -> Unit
    ): Continuation<T> =
    object : Continuation<T> {
        override val context: CoroutineContext
            get() = context

        override fun resumeWith(result: Result<T>) =
            resumeWith(result)
    }

/**
 * Starts a coroutine without a receiver and with result type [T].
 * This function creates and starts a new, fresh instance of suspendable computation every time it is invoked.
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <T> (suspend () -> T).startCoroutine(
    completion: Continuation<T>
) {
    createCoroutineUnintercepted(completion).intercepted().resume(Unit)
}

/**
 * Starts a coroutine with receiver type [R] and result type [T].
 * This function creates and starts a new, fresh instance of suspendable computation every time it is invoked.
 */
@SinceKotlin("1.3")
@Suppress("UNCHECKED_CAST")
public fun <R, T> (suspend R.() -> T).startCoroutine(
    receiver: R,
    completion: Continuation<T>
) {
    createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)
}
```

---

Open Source Used In DNAC Platform Antelope 14025
createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit) // Obtains the current continuation instance inside suspend functions and suspends * the currently running coroutine.

In this function both [Continuation.resume] and [Continuation.resumeWithException] can be used either synchronously inUnit * the same stack-frame where the suspension function is run or asynchronously later in the same thread or in a different thread of execution. Subsequent invocation of any resume function will produce an [IllegalStateException].

suspendCoroutine(crossinline block: (Continuation<T>) -> Unit): T { // T \n  contract { callsInPlace(block, InvocationKind.EXACTLY_ONCE) } \n  return suspendCoroutineUninterceptedOrReturn { c: Continuation<T> - \n  } val safe = SafeContinuation(c.intercepted())\n  block(safe)\n  safe.getOrThrow()\n} \n
// Returns the context of the current coroutine.

suspendCoroutineUninterceptedOrReturn { \n  completion: Continuation<T>\n} Any? = this.asDynamic(completion, false) // Starts an unintercepted coroutine without a receiver and with result type [T] and executes it until its first suspension.

suspendCoroutineUninterceptedOrReturn { \n  receiver: R, \n  completion: Continuation<T>\n} Any? = this.asDynamic(receiver, completion, false) // Starts an unintercepted coroutine with receiver type [R] and result type [T] and executes it until its first suspension.

suspendCoroutineUninterceptedOrReturn { \n  receiver: R, \n  param: P, \n  completion: Continuation<T>\n} Any? = this.asDynamic(receiver, param, completion, false) // Creates unintercepted coroutine without receiver and with result type [T].

This function creates a new, fresh instance of suspendable computation every time it is invoked. InUnit * To start executing the created coroutine, invoke `resume(Unit)` on the returned [Continuation] instance.

The [completion] continuation is invoked when coroutine completes with a result or an exception. In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception. In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception. In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception. In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception.

Note that [completion] of this function may get invoked in an arbitrary context inUnit * [Continuation.intercepted] can be used to acquire the intercepted continuation. InUnit * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of both the coroutine and [completion] happens in the invocation context established byUnit * [ContinuationInterceptor]. InUnit * Repeated invocation of any resume function
on the resulting continuation corrupts the state machine of the coroutine and may result in arbitrary behaviour or exception.

* If (this.asDynamic().length == 2) {
    // When `suspended` is true the continuation is created, but not executed
    when (completion) {
        // Kotlin/JS suspend lambdas have an extra parameter
        if (this.asDynamic().length == 3) {
            // When `suspended` is true the continuation is created, but not executed
            this.asDynamic()(receiver, completion, true)
        } else {
            createCoroutineFromSuspendFunction(completion) {
                this.asDynamic()(receiver, completion)
            }
        }
    }
}

private inline fun <T> createCoroutineFromSuspendFunction(completion: Continuation<T>, crossinline block: () -> Any?): Continuation<Unit> {
    @Suppress("UNCHECKED_CAST")
    return object : CoroutineImpl(completion as Continuation<Any?>) {
        override fun doResume(): Any? {
            exception?.let { throw it }
            return block()
        }
    }
}

// Mirrors signature from JS IR
@library
@JsName("imulEmulated")
internal fun imul(x: Int, y: Int): Int = definedExternally

internal inline fun isArrayish(o: dynamic) =
    js("Kotlin").isArrayish(o)
constructor(message: String?, cause: Throwable?) : Throwable(message, cause ?: null) {
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class IllegalArgumentException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class IllegalStateException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class IndexOutOfBoundsException actual constructor(message: String?) : RuntimeException(message) {
    actual constructor() : this(null)
}

public actual open class ConcurrentModificationException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class UnsupportedOperationException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class NumberFormatException actual constructor(message: String?) : IllegalArgumentException(message) {
    actual constructor() : this(null)
}

public actual open class NullPointerException actual constructor(message: String?) : RuntimeException(message) {
    actual constructor() : this(null)
}

public actual open class ClassCastException actual constructor(message: String?) : RuntimeException(message) {
    actual constructor() : this(null)
}

public actual open class AssertionError actual constructor(message: String?, cause: Throwable?) : Error(message, cause) {
    actual constructor() : this(null)
    constructor(message: String?) : this(message, null)
    actual constructor(message: Any?) : this(message.toString(), message as? Throwable)
}

public actual open class NoSuchElementException actual constructor(message: String?) : RuntimeException(message) {
    actual constructor() : this(null)
}

public actual open class ArithmeticException actual constructor(message: String?) : RuntimeException(message) {
    actual constructor() : this(null)
}

public actual open class NoWhenBranchMatchedException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}

public actual open class UninitializedPropertyAccessException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {
    actual constructor() : this(null, null)
    actual constructor(message: String?) : this(message, null)
    actual constructor(cause: Throwable?) : this(cause)
}
this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

*Apache* Used In DNAC Platform Antelope 14029

*/

package kotlin

/**
 * Returns a bit representation of the specified floating-point value as [Long] according to the IEEE 754 floating-point "double format" bit layout.
 */
@SinceKotlin("1.2")
@library("doubleToBits")
public actual fun Double.toBits(): Long = definedExternally

/**
 * Returns a bit representation of the specified floating-point value as [Long] according to the IEEE 754 floating-point "double format" bit layout, preserving NaN values exact layout.
 */
@SinceKotlin("1.2")
@library("doubleToRawBits")
public actual fun Double.toRawBits(): Long = definedExternally

/**
 * Returns the [Double] value corresponding to a given bit representation.
 */
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly

/**
 * Returns a bit representation of the specified floating-point value as [Int] according to the IEEE 754 floating-point "single format" bit layout.
 */
@SinceKotlin("1.2")
@library("floatToBits")
public actual fun Float.toBits(): Int = definedExternally

/**
 * Returns a bit representation of the specified floating-point value as [Int] according to the IEEE 754 floating-point "single format" bit layout, preserving NaN values exact layout.
 */
@SinceKotlin("1.2")
@library("floatToRawBits")
public actual fun Float.toRawBits(): Int = definedExternally

/**
 * Returns the [Float] value corresponding to a given bit representation.
 */
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual inline fun Float.Companion.fromBits(bits: Int): Float = js("Kotlin").floatFromBits(bits).unsafeCast<Float>()

internal inline fun Long(low: Int, high: Int) = js("Kotlin").Long.fromBits(low, high).unsafeCast<Long>()
internal inline val Long.low: Int get() = this.asDynamic().getLowBits().unsafeCast<Int>()
internal inline val Long.high: Int get() = this.asDynamic().getHighBits().unsafeCast<Int>()

internal inline fun <T : Annotation> KClass<T>.findAssociatedObject(annotationClass: KClass<T>): Any? {
    // This API is not supported in js-v1. Return `null` to be source-compatible with js-ir. Return null
    return null
}

package kotlin.reflect

import kotlin.reflect.KClass

@PublishedApi
internal fun <T : Annotation> KClass<T>.findAssociatedObject(annotationClass: KClass<T>): Any? {
    // This API is not supported in js-v1. Return `null` to be source-compatible with js-ir. Return null
    return null
}

import kotlin.reflect.KClass

@PublishedApi
internal fun <T : Annotation> KClass<T>.findAssociatedObject(annotationClass: KClass<T>): Any? {
    // This API is not supported in js-v1. Return `null` to be source-compatible with js-ir. Return null
    return null
}

package kotlin.text

// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib

import kotlin.js.*

/**
 * Returns a character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char sequence.
 */
public actual fun

Open Source Used In DNAC Platform Antelope 14029
CharSequence.elementAt(index: Int): Char {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, length: $length") }
}

/*
 * Copyright 2010-2020 JetBrains
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

//
// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt
// See:
https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib/\n\nimport kotlin.collections.Elements

public actual fun UIntArray.elementAt(index: Int): UInt {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

public actual fun ULongArray.elementAt(index: Int): ULong {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

public actual fun UByteArray.elementAt(index: Int): UByte {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

public actual fun UShortArray.elementAt(index: Int): UShort {
    return elementAtOrElse(index) { throw IndexOutOfBoundsException("index: $index, size: $size") }
}

public actual fun UIntArray.asList(): List<UInt> {
    return abstractList().randomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: UInt): Boolean = this@asList.contains(element)
        override fun get(index: Int): UInt {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: UInt): Int {
            if ((element as Any?) !is UInt) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: UInt): Int {
            if ((element as Any?) !is UInt) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}

public actual fun ULongArray.asList(): List<ULong> {
    return abstractList().randomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: ULong): Boolean = this@asList.contains(element)
        override fun get(index: Int): ULong {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: ULong): Int {
            if ((element as Any?) !is ULong) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: ULong): Int {
            if ((element as Any?) !is ULong) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}

public actual fun UByteArray.asList(): List<UByte> {
    return abstractList().randomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: UByte): Boolean = this@asList.contains(element)
        override fun get(index: Int): UByte {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: UByte): Int {
            if ((element as Any?) !is UByte) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: UByte): Int {
            if ((element as Any?) !is UByte) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}


a [List] that wraps the original array. 
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public actual fun UShortArray.asList(): List<UShort> {
    return object : AbstractList<UShort>(), RandomAccess {
        override val size: Int get() = this@asList.size
        override fun isEmpty(): Boolean = this@asList.isEmpty()
        override fun contains(element: UShort): Boolean = this@asList.contains(element)
        override fun get(index: Int): UShort {
            AbstractList.checkElementIndex(index, size)
            return this@asList[index]
        }
        override fun indexOf(element: UShort): Int {
            if ((element as Any?) !is UShort) return -1
            return this@asList.indexOf(element)
        }
        override fun lastIndexOf(element: UShort): Int {
            if ((element as Any?) !is UShort) return -1
            return this@asList.lastIndexOf(element)
        }
    }
}

package jquery.ui

import jquery.JQuery
import kotlin.js.Json

@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery'
package instead.", level = DeprecationLevel.ERROR)
public inline fun JQuery.buttonset(): JQuery = asDynamic().buttonset()

@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery'
package instead.", level = DeprecationLevel.ERROR)
public inline fun JQuery.dialog(): JQuery = asDynamic().dialog()

@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery'
package instead.", level = DeprecationLevel.ERROR)
public inline fun JQuery.dialog(params: Json): JQuery = asDynamic().dialog(params)

@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery'
package instead.", level = DeprecationLevel.ERROR)
public inline fun JQuery.button(): JQuery = asDynamic().button()

@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery'
package instead.", level = DeprecationLevel.ERROR)
public inline fun JQuery.accordion(): JQuery = asDynamic().accordion()

@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery'
package instead.", level = DeprecationLevel.ERROR)
public inline fun JQuery.draggable(params: Json): JQuery = asDynamic().draggable(params)

@Deprecated("Use declarations from 'https://bintray.com/kotlin/js-externals/kotlin-js-jquery'
package instead.", level = DeprecationLevel.ERROR)
public inline fun JQuery.selectable(): JQuery = asDynamic().selectable()
are writing a JavaScript library in Kotlin. The compiler produces mangled names for functions with parameters, which is unnatural for usual JavaScript developer. You can put `@JsName(...)` on functions you want to be available from JavaScript. For some reason you want to rename declaration, e.g. there's common term in JavaScript for a concept provided by the declaration, which is uncommon in Kotlin. Example:
```kotlin
class Person(val name: String) {
    fun hello() {
        println("Hello $name!")
    }

    @JsName("helloWithGreeting")
    fun hello(greeting: String) {
        println("$greeting $name!")
    }
}
```
It's required to denote a valid JavaScript identifier. The compiler produces the code relevant for the target module system, for example, in case of CommonJS, it will import the declaration via the `require(...)`. The annotation can be used on top-level external declarations (classes, properties, functions) and files. In case of file (which can't be `external`) the following rule applies: all the declarations in the file must be `external`. By applying `@JsModule(...)` on a file you tell the compiler to import a JavaScript object that contain all the declarations from the file. Some JavaScript libraries are distributed both as a standalone downloadable piece of JavaScript and as a module available as an npm package. To tell the Kotlin compiler to accept both cases, you can augment with the `@JsNonModule` annotation: `@JsModule("jquery") @JsNonModule
@JsName("$")
external abstract class JQuery() {
    // some declarations here
}
```
Some JavaScript libraries are distributed as an npm package, which requires importing the library via `require(...)`. The `@JsModule` annotation can be applied on top-level declarations (classes, properties, functions). Some JavaScript libraries are distributed both as a standalone downloadable piece of JavaScript and as a module. The compiler turns references to `external` declarations either to plain unprefixed names (in case of `plain` modules) or to plain imports. However, by applying `@JsModule` annotation you can make a declaration unavailable to `plain` module system. Some JavaScript libraries are distributed as a module available as an npm package. This annotation can be applied to either files or top-level declarations. It is currently prohibited to export the following kinds of declarations:
- `expect` declarations
- inline functions with reified type parameters
- suspended functions
- secondary constructors without `@JsName`
- extension properties
- enum classes
- annotation classes
- `Signature of exported declarations must only contain `Exportable!` types

Open Source Used in DNAC Platform Antelope 14032
`dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`, `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`, `Array<exportable-type>`, `*` Function types with exportable parameters and return types

* `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`

* `Array<exportable-type>`

* Function types with exportable parameters and return types

* `external` or `@JsExport` classes and interfaces

* Nullable counterparts of types above

* `Unit` return type. Must not be nullable

* This annotation is experimental, meaning that restrictions mentioned above are subject to change.

*/

@ExperimentalJsExport
@Retention(AnnotationRetention.BINARY)
@Target(CLASS, PROPERTY, FUNCTION, FILE)

public actual annotation class JsExport

@Target(AnnotationTarget.FIELD)
@Retention(AnnotationRetention.SOURCE)

public actual annotation class Volatile

@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY_SETTER)
@Retention(AnnotationRetention.SOURCE)

public actual annotation class Synchronized

/**
 * Provides a skeletal implementation of the [MutableCollection] interface. This is not
 * an actual abstract override func add(element: E):
 * Boolean
 * actual override fun remove(element: E): Boolean
 * checkIsMutable() val iterator =
 * iterator.next()() { () return true
 * return false
 * actual override fun addAll(elements:
 * Collection<E>): Boolean
 * if (add(element)) modified = true
 * return modified
 */

@JsName("toJSON")
open fun toJSON(): Any = this.toArray()
actual override fun iterator(): MutableListIterator<E> = IteratorImpl()

actual override fun contains(element: E): Boolean = indexOf(element) !=-1

actual override fun indexOf(element: E): Int {
    for (index in 0..lastIndex) {
        if (get(index) == element) return index
    }
    return -1
}

actual override fun lastIndexOf(element: E): Int {
    for (index in lastIndex downTo 0) {
        if (get(index) == element) return index
    }
    return -1
}

actual override fun removeRange(fromIndex: Int, toIndex: Int) {
    iterator.next() iterator.remove() return 0
}

actual override fun removeAt(index: Int): E {
    AbstractList.checkElementIndex(index, _size)

    return list.removeAt(index) _size-- return result
}

actual override fun set(index: Int, element: E) {
    AbstractList.checkElementIndex(index, _size)

    list.add(index, element)
    _size++ return result
}

actual override fun add(index: Int, element: E) {
    AbstractList.checkPositionIndex(index, _size)

    if (get(index) == element) return index
    return -1
}

actual override fun lastIndexOf(element: E): Int {
    for (index in lastIndex downTo 0) {
        if (get(index) == element) return index
    }
    return -1
}

actual override fun indexOf(element: E): Int {
    for (index in 0..lastIndex) {
        if (get(index) == element) return index
    }
    return -1
}

actual override fun iterator(): MutableListIterator<E> = IteratorImpl()

actual override fun listIterator(): MutableListIterator<E> = listIterator(0)

actual override fun listIterator(index: Int): MutableListIterator<E> = listIterator(index)

actual override fun subList(fromIndex: Int, toIndex: Int): MutableList<E> = SubList(this, fromIndex, toIndex)

/**
 * Based on GWT AbstractMap
 * Copyright 2007 Google Inc.
 */

/*
 * The implementor is required to implement [entries] property, which should return mutable set of map entries, and
 * put] function.
 * @param K the type of map keys. The map is invariant in its key type.
 * @param V the type of map values. The map is invariant in its value type.
 */
}
Open Source Used In DNAC Platform Antelope
throw UnsupportedOperationException()
    }

    private fun rangeCheck(index: Int) = index.apply {
        AbstractList.checkElementIndex(index, size)
    }

    private fun insertionRangeCheck(index: Int) = index.apply {
        AbstractList.checkPositionIndex(index, size)
    }

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

internal fun <T> sortArrayWith(array: Array<out T>, comparison: (T, T) -> Int) {
    if (getStableSortingIsSupported()) {
        array.asDynamic().sort(comparison)
    } else {
        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, Comparator(comparison))
    }
}

internal fun <T> sortArrayWith(array: Array<out T>, comparator: Comparator<in T>) {
    if (getStableSortingIsSupported()) {
        val comparison = { a: T, b: T -> comparator.compare(a, b) }
        array.asDynamic().sort(comparison)
    } else {
        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, comparator)
    }
}

internal fun <T> sortArrayWith(array: Array<out T>, fromIndex: Int, toIndex: Int, comparator: Comparator<in T>) {
    if (fromIndex < toIndex - 1) {
        mergeSort(array.unsafeCast<Array<T>>(), fromIndex, toIndex, comparator)
    }
}

internal fun <T : Comparable<T>> sortArray(array: Array<out T>) {
    if (getStableSortingIsSupported()) {
        val comparison = { a: T, b: T -> a.compareTo(b) }
        array.asDynamic().sort(comparison)
    } else {
        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, naturalOrder())
    }
}

private var _stableSortingIsSupported: Boolean? = null
private fun getStableSortingIsSupported(): Boolean {
    _stableSortingIsSupported?.let { return it }
    _stableSortingIsSupported = false
    _stableSortingIsSupported = false
    var result = 1
    for (i in 1 until array.size) {
        if (array[i] == null) continue
        if (array[i] == array[i - 1]) continue
        result *= array[i].contentHashCode()
    }
    return result
}

private fun <T> mergeSort(array: Array<T>, start: Int, end: Int, comparator: Comparator<in T>): Array<T> {
    if (start == end) {
        return array
    }

    val median = (start + end) / 2
    val left = mergeSort(array, start, median, comparator)
    val right = mergeSort(array, median + 1, end, comparator)

    var target = if (left === buffer) array else buffer

    // Merge.
    var leftIndex = start
    var rightIndex = median + 1
    for (i in start..end) {
        if (comparator.compare(left, right) <= 0) {
            target[i] = left
            leftIndex++
        } else {
            target[i] = right
            rightIndex++
        }
    }

    return target
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

@OptIn(ExperimentalUnsignedTypes::class)
@SinceKotlin("1.3")
@kotlin.js.JsName("contentDeepHashCodeImpl")
internal fun <T> Array<out T>?.contentDeepHashCodeImpl(): Int {
    if (this == null) return 0
    var result = 1
    for (element in this) {
        val elementHash = when {
            element == null -> 0
            isArrayish(element) -> (element.unsafeCast<Array<*>>()).contentDeepHashCodeImpl()

            element is UByteArray   -> element.contentHashCode()
            element is UShortArray  -> element.contentHashCode()
            element is UIntArray    -> element.contentHashCode()
            element is ULongArray   -> element.contentHashCode()

            else                    -> element.hashCode()
        }

        result = 31 * result + elementHash
    }
    return result
}
Kotlin Programming Language contributors.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```
package kotlin.collections

internal interface EqualityComparator {
  /**
   * Subclasses must override to return a value indicating whether or not two keys or values are equal.
   */
  abstract fun equals(value1: Any?, value2: Any?): Boolean

  /**
   * Subclasses must override to return the hash code of a given key.
   */
  abstract fun getHashCode(value: Any?): Int

  object HashCode : EqualityComparator {
    override fun equals(value1: Any?, value2: Any?): Boolean = value1 == value2
    override fun getHashCode(value: Any?): Int = value?.hashCode() ?: 0
  }
}
```

```
/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

/*
 * Based on GWT AbstractHashMap
 * Copyright 2008 Google Inc.
 */

package kotlin.collections

import kotlin.collections.MutableMap.MutableEntry

/**
 * Hash table based implementation of the [MutableMap] interface.
 */

actual class HashMap<K, V> : AbstractMutableMap<K, V>, MutableMap<K, V> {

    private inner class EntrySet :

      override fun add(element: MutableEntry<K, V>): Boolean = throw UnsupportedOperationException("Add is not supported on entries")
      override fun clear() {
        this@HashMap.clear()
      }
      override operator fun contains(element: MutableEntry<K, V>): Boolean = containsEntry(element)
      override operator fun iterator(): MutableIterator<MutableEntry<K, V>> = internalMap.iterator()
      override fun remove(element: MutableEntry<K, V>): Boolean {
        if (contains(element)) {
          this@HashMap.remove(element.key)
          return true
        }
        return false
      }
      override val size: Int get() = this@HashMap.size
    }

    /**
     * Internal implementation of the map: either string-based or hashcode-based.
     */
    private val internalMap:

    private val equality: EqualityComparator

    internal constructor(internalMap:
      actual constructor(initialCapacity: Int, loadFactor: Float = 0.0f) : this(initialCapacity, 0.0f)

    actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)

    /**
     * Constructs an empty [HashMap] instance.
     */
    actual constructor() : this(InternalHashCodeMap(EqualityComparator.HashCode))

    private var _entries: MutableSet<MutableMap.MutableEntry<K, V>>? = null
    actual override val entries: MutableSet<MutableMap.MutableEntry<K, V>>
        get() {
            if (_entries == null) {
                _entries = createEntrySet()
            }
            return _entries!!
        }

    protected open fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()

    actual override operator fun get(key: K): V? = internalMap.get(key)
    actual override fun put(key: K, value: V): V? = internalMap.put(key, value)
    actual override fun remove(key: K): V? = internalMap.remove(key)
    actual override val size: Int get() = internalMap.size

    actual override fun putAll(pairs: Pair<String, V>): HashMap<String, V> {
        return HashMap<String, V>(InternalStringMap(EqualityComparator.HashCode)).apply { putAll(pairs) }
    }
}
```

---

Open Source Used In DNAC Platform Antelope 14038
package kotlin.collections

/**
 * The implementation of the [MutableSet] interface, backed by a [HashMap] instance.
 * Classes that extend HashSet and implement `build()` (freezing) operation must have to make sure mutating methods check `checkIsMutable`
 * 
 */

public actual open class HashSet<E> : AbstractMutableSet<E>, MutableSet<E> {

    internal val map: HashMap<E, Any>

    /**
     * Constructs a new empty [HashSet].
     */
    actual constructor() {
        map = HashMap<E, Any>()
    }

    /**
     * Constructs a new [HashSet] filled with the elements of the specified collection.
     */
    actual constructor(elements: Collection<E>) {
        map = HashMap<E, Any>(elements.size)
        addAll(elements)
    }

    /**
     * Constructs a new empty [HashSet].
     *
     * @param initialCapacity the initial capacity
     * @param loadFactor the load factor
     *
     * @throws IllegalArgumentException if the initial capacity or load factor are negative
     */
    @Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
    actual constructor(initialCapacity: Int, loadFactor: Float = 0.0f) {
        map = HashMap<E, Any>(initialCapacity, loadFactor)
    }

    actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)

    internal constructor(map: HashMap<E, Any>) {
        this.map = map
    }

    actual override fun add(element: E): Boolean {
        val old = map.put(element, this)
        return old == null
    }

    actual override fun clear() {
        map.clear()
    }

    actual override fun contains(element: E): Boolean = map.containsKey(element)

    actual override fun isEmpty(): Boolean = map.isEmpty()

    actual override fun iterator(): MutableIterator<E> = map.keys.iterator()

    actual override fun remove(element: E): Boolean = map.remove(element) != null

    actual override val size: Int get() = map.size

}

/**
 * Creates a new instance of the specialized implementation of [HashSet] with the specified [String] elements,
 * which elements the keys as properties of JS object without hashing them.
 */
public fun stringSetOf(vararg elements: String): HashSet<String> {
    return HashSet(stringMapOf<Any>()).apply { addAll(elements) }
}

import kotlin.collections.MutableMap.MutableEntry
import kotlin.collections.AbstractMutableMap.SimpleEntry

/**
 * A simple wrapper around JavaScriptObject to provide [java.util.Map]-like semantics for any type.
 * 
 * Implementation notes:
 * 
 * A key's hashCode is the index in backingMap which should contain that key. Since several keys may have the same hash,
 * each value in hashCodeMap is actually an array containing all entries whose key shares the same hash.
 * The entries in the array are not ordered.
 */
internal class InternalHashCodeMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {

    private var backingMap: dynamic = createJsMap()
    override val size: Int = 0
        private set

    override fun put(key: K, value: V): V? {
        val hashCode = equality.getHashCode(key)
        val chainOrEntry = getChainOrEntryOrNull(hashCode) ?: return null
        actual override fun clone(): Any {
            return HashSet<E>(this)
        }
    }

    override fun remove(key: K): V? {
        val hashCode = equality.getHashCode(key)
        val chainOrEntry = getChainOrEntryOrNull(hashCode) ?: return null
    }

    override fun contains(key: K): Boolean = equality.equals(backingMap[hashCode], key)

    override fun isEmpty(): Boolean = backingMap.size == 0

    override fun entrySet(): Set<Map.Entry<K, V>> = backingMap.entrySet()

    override fun equals(other: Any?): Boolean {
        return equality.equals(this, other)
    }

    override fun hashCode(): Int {
        return equality.getHashCode(this)
    }

    override fun structureChanged(host) {
        // This is a new chain, put it to the map.
        backingMap[hashCode] = SimpleEntry(key, value)
        else {
            if (entryOrEntry != Array<*>()) {
                // It is an entry
                val entry = SimpleEntry<K, V>(key, value)
                size++
            } else {
                // Chain already exists, perhaps key also exists.
                val chain = Array<MutableEntry<K, V>>()
                val entry = chain.findEntryInChain(key)
                if (entry == null) {
                    backingMap[hashCode] = arrayOf(entry)
                    size++
                } else {
                    backingMap[hashCode] = arrayOf(entry, SimpleEntry<K, V>())
                    size++
                }
            }
        }
        return null
    }

    override fun contains(key: K): Boolean = equality.equals(backingMap[hashCode], key)

    override fun isEmpty(): Boolean = backingMap.size == 0

    override fun entrySet(): Set<Map.Entry<K, V>> = backingMap.entrySet()

    override fun equals(other: Any?): Boolean {
        return equality.equals(this, other)
    }

    override fun hashCode(): Int {
        return equality.getHashCode(this)
    }

    override fun structureChanged(host) {
        // This is a new chain, put it to the map.
        backingMap[hashCode] = SimpleEntry(key, value)
        else {
            if (entryOrEntry != Array<*>()) {
                // It is an entry
                val entry = SimpleEntry<K, V>(key, value)
                size++
            } else {
                // Chain already exists, perhaps key also exists.
                val chain = Array<MutableEntry<K, V>>()
                val entry = chain.findEntryInChain(key)
                if (entry == null) {
                    backingMap[hashCode] = arrayOf(entry)
                    size++
                } else {
                    backingMap[hashCode] = arrayOf(entry, SimpleEntry<K, V>())
                    size++
                }
            }
        }
        return null
    }

    override fun contains(key: K): Boolean = equality.equals(backingMap[hashCode], key)

    override fun isEmpty(): Boolean = backingMap.size == 0

    override fun entrySet(): Set<Map.Entry<K, V>> = backingMap.entrySet()

    override fun equals(other: Any?): Boolean {
        return equality.equals(this, other)
    }

    override fun hashCode(): Int {
        return equality.getHashCode(this)
    }

    override fun structureChanged(host) {
        // This is a new chain, put it to the map.
        backingMap[hashCode] = SimpleEntry(key, value)
        else {
            if (entryOrEntry != Array<*>()) {
                // It is an entry
                val entry = SimpleEntry<K, V>(key, value)
                size++
            } else {
                // Chain already exists, perhaps key also exists.
                val chain = Array<MutableEntry<K, V>>()
                val entry = chain.findEntryInChain(key)
                if (entry == null) {
                    backingMap[hashCode] = arrayOf(entry)
                    size++
                } else {
                    backingMap[hashCode] = arrayOf(entry, SimpleEntry<K, V>())
                    size++
                }
            }
        }
        return null
    }

    override fun contains(key: K): Boolean = equality.equals(backingMap[hashCode], key)

    override fun isEmpty(): Boolean = backingMap.size == 0

    override fun entrySet(): Set<Map.Entry<K, V>> = backingMap.entrySet()
(equality.equals(entry.key, key)) {\n    jsDeleteProperty(backingMap, hashCode)\n    size--\n    return entry.value\n} else {\n    return null\n} else {\n    val chain: Array<MutableEntry<K, V>> = chainOrEntry\n    for (index in chain.indices) {\n        val entry = chain[index]\n        if (equality.equals(entry.key, key)) {\n            if (chain.size == 1) {\n                chain.asDynamic().length = 0\n                // remove the whole array\n                jsDeleteProperty(backingMap, hashCode)\n            } else {\n                // splice out the entry we're removing\n                chain.asDynamic().splice(index, 1)\n            }\n            size--\n        }\n    }\n    return null\n} else {\n    val chain: Array<MutableEntry<K, V>> = chainOrEntry\n    for (index in chain.indices) {\n        val entry = chain[index]\n        if (equality.equals(entry.key, key)) {\n            if (chain.size == 1) {\n                chain.asDynamic().splice(index, 1)\n            } else {\n                // splice out the entry we're removing\n                chain.asDynamic().splice(index, 1)\n            }\n            size--\n        }\n    }\n    return null\n} else {\n    function override fun remove(key: K): V? {\n        return getEntry(key)\n    }\n}

override fun clear() {\n    backingMap = createJsMap()\n    size = 0\n}

override fun contains(key: K): Boolean = getEntry(key) != null\n
override fun get(key: K): V? = getEntry(key)?.value\n
private fun getEntry(key: K): MutableEntry<K, V>? {\n    val chainOrEntry = getChainOrEntryOrNull(equality.getHashCode(key)) ?: return null\n    if (chainOrEntry !is Array<*>) {\n        val entry: MutableEntry<K, V> = chainOrEntry\n        if (equality.equals(entry.key, key)) {\n            return entry\n        } else {\n            return null\n        }
    } else {\n        val chain: Array<MutableEntry<K, V>> = chainOrEntry\n        return chain.findEntryInChain(key)\n    }
}

private fun Array<MutableEntry<K, V>>.findEntryInChain(key: K): MutableEntry<K, V>? =\n    firstOrNull { entry ->\n        equality.equals(entry.key, key)\n    }

override fun iterator(): MutableIterator<MutableEntry<K, V>> {\n    return object : MutableIterator<MutableEntry<K, V>> {\n        var state = -1 // -1 not ready, 0 - ready, 1 - done\n        var chainOrEntry: dynamic = null\n        var isChain = false\n        var itemIndex = 0\n        var lastEntry: MutableEntry<K, V>? = null\n
        private fun computeNext(): Int {\n            if (chainOrEntry != null && isChain) {\n                val chainSize: Int = chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>().size\n                if (++itemIndex < chainSize) return 0\n            }\n
            if (++keyIndex < keys.size) {\n                chainOrEntry = backingMap[keys[keyIndex]]\n                isChain = chainOrEntry is Array\n                itemIndex = 0\n            } else {\n                chainOrEntry = null\n                return 1\n            }\n        }

        override fun hasNext(): Boolean {
            if (state == -1)\n                state = computeNext()
            return state == 0\n        }

        override fun next(): MutableEntry<K, V> {\n            if (!hasNext()) throw NoSuchElementException()\n            val lastEntry = if (isChain) {\n                chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>().itemIndex\n            } else {\n                chainOrEntry.unsafeCast<MutableEntry<K, V>>().itemIndex\n            }\n            var state = -1\n            return lastEntry\n        }

        override fun remove() {\n            checkNotNull(lastEntry)\n            this@InternalHashCodeMap.remove(lastEntry!!.key)\n        }
    }
}

private fun getChainOrEntryOrNull(hashCode: Int): dynamic {\n    val chainOrEntry = backingMap[hashCode]\n    return if (chainOrEntry === undefined) null else chainOrEntry\n}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.collections

/**
 * A simple wrapper around JavaScript Map for key type is string.
 */

package kotlin.collections

import kotlin.collections.MutableMap.MutableEntry

/**
 * A simple wrapper around JavaScript Map for key type is string.
 * Though this map is instantiated only with
 */

Open Source Used In DNAC Platform Antelope 14040
K=String, the K type is not fixed to String statically,
* because we want to have it erased to Any? in order not to
generate type-safe override bridges for
* [get], [contains], [remove] etc, if they ever are generated.
* A mod count to track 'value' replacements in map to ensure that the 'value' that we have in the
* iterator entry is guaranteed to be still correct.
* This is to optimize for the common scenario where the values are not modified
during
* iterations where the entries are never stale.
* private var backingMap: dynamic = createJsMap()
* override operator fun contains(key: K): Boolean
* { return false
* }
* override operator fun get(key: K): V?
* { return null
* }
* override fun put(key: K, value: V): V?
* { return null
* }
* override fun remove(key: K): V?
* { return null
* }
* override fun clear()
* { this@InternalStringMap.backingMap = createJsMap()
* }
* override fun iterator()
* { return object : MutableIterator<MutableEntry<K, V>> {
*     private val keys: Array<String> =
*         js("Object").keys(this@InternalStringMap.backingMap)
*     private val iterator = keys.iterator()
*     private var lastKey: String? =
*         null
*     override fun hasNext(): Boolean = iterator.hasNext()
*     override fun next():
*        MutableEntry<K, V> {
*         val key = iterator.next()
*         lastKey = key
*         @Suppress("UNCHECKED_CAST")
*         return newMapEntry(key as K)
*     }
*     override fun remove()
*     { this@InternalStringMap.remove(checkNotNull(lastKey) as K)
*     }
*     override fun equals(other: Any?): Boolean =
*         AbstractMap.entryEquals(this, other)
*     override fun hashCode(): Int =
*         AbstractMap.entryHashCode(this)
* }
* }
* }
* class InternalStringMap<K, V>(override val equality: EqualityComparator) :
* InternalMap<K, V> {
*     private var backingMap: dynamic = createJsMap()
*     override val override var size: Int = 0
*     private set
*     /*
*     * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
*     * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
*     */
*     /*
*     * Based on GWT LinkedHashMap
*     * Copyright 2008 Google Inc.
*     */
*     import kotlin.collections.MutableMap
*     import kotlin.collections.MutableMap.Entry
*     public actual open class LinkedHashMap<K, V> : HashMap<K, V>,
*         MutableMap<K, V> {
*         /*
*         * The entry we use includes next/prev pointers for a doubly-linked circular
*         * list with a head node. This reduces the special cases we have to deal with
*         * in the list operations.
*         * Note that we duplicate the key from the underlying hash map so we can find
*         * the eldest entry. The alternative would have been to modify HashMap so
*         * more
*         * of the code was directly usable here, but this would have added some
*         * overhead to HashMap, or to reimplement most of the Hashmap code here within
*         * small modifications. Paying a small storage cost only if you use
*         * LinkedHashSet and minimizing code size seemed like a better tradeoff
*         */
*         private inner class ChainEntry<K, V>(key: K, value: V) :
*             AbstractMutableMap.SimpleEntry<K, V>(key, value)
*         { this@LinkedHashMap.entrySet() =
*             LinkedHashMap蹇
*         }
*         private inner class EntrySet :
*             AbstractMutableSet<MutableEntry<K, V>> {
*             override fun remove()
*             { this@LinkedHashMap.entrySet() =
*         }
private var last: ChainEntry<K, V>? = null

private var next: ChainEntry<K, V>? = null

init {
    next = head
//                recordLastKnownStructure(map, this)
}

override fun hasNext(): Boolean {
    return next !== null
}

override fun next(): MutableEntry<K, V> {
//                checkStructuralChange(map, this)
                if (!hasNext()) throw NoSuchElementException()

                val current = next!!
                last = current
                next = current.next.takeIf { it !== head }!!
                return current

            }

override fun remove() {
    check(last != null)
//                recordLastKnownStructure(map, this)

                last!!.remove()
//                recordLastKnownStructure(map, this)
                last = null
            }

    }

    override fun add(element: MutableEntry<K, V>): Boolean = throw UnsupportedOperationException("Add is not supported on entries")

    override fun clear() {
        this@LinkedHashMap.clear()
    }

    override operator fun contains(element: MutableEntry<K, V>): Boolean = containsEntry(element)

    override operator fun iterator(): MutableIterator<MutableEntry<K, V>> = EntryIterator()

    override fun remove(element: MutableEntry<K, V>): Boolean {
        checkIsMutable()
            if (contains(element)) {
                this@LinkedHashMap.remove(element.key)
                return true
            } else return false
        }

    override val size: Int get() = this@LinkedHashMap.size

    override fun checkIsMutable(): Unit = this@LinkedHashMap.checkIsMutable()
}

private var head: ChainEntry<K, V>? = null

private fun ChainEntry<K, V>.addToEnd() {
    // This entry is not in the list.
    check(next == null && prev == null)

    val _head = head
    if (_head == null) {
        head = this
        next = this
        prev = this
    } else {
        // Chain is valid.
        val _tail = checkNotNull(_head.prev)
        // Update me.
        prev = _tail
        next = _head
        // Update my new siblings: current head and old tail
        _head.prev = this
        _tail.next = this
    }
}

private fun ChainEntry<K, V>.remove() {
    if (this.next === this) {
        // if this is single element, remove head
        head = null
    } else {
        if (head === this) {
            // if this is first element, move head to next
            head = next
        }
        next!!.prev = prev
        prev!!.next = next
    }
}

/*
 * The hashmap that keeps track of our entries and the chain.
 * Duplicate the key here to eliminate changes to HashMap and minimize the code
 * here, at the expense of additional space.
 */
private val map: HashMap<K, ChainEntry<K, V>>

private var isReadOnly: Boolean = false

actual constructor() : super() {
    map = HashMap<K, ChainEntry<K, V>>()
}

private constructor(backingMap: HashMap<K, Any>) : super() {
    @Suppress("UNCHECKED_CAST") // expected to work due to erasure
//        map = backingMap as HashMap<K, ChainEntry<K, V>>
    map = backingMap as HashMap<K, ChainEntry<K, V>>
}

private constructor(initialCapacity: Int, loadFactor: Float = 0.0f) : super(initialCapacity, loadFactor) {
    this.map = HashMap<K, ChainEntry<K, V>>()
}

actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)

actual constructor(original: Map<out K, V>) {

    this.putAll(original)
//    @PublishedApi internal fun build(): Map<K, V> {
//        checkIsMutable() // Nullable
//        map.clear() // Nullable
//        override fun clone(): Any {
//            return LinkedHashMap(this)
//        }
//    }
//    override fun clone(): Any {
//        return LinkedHashMap(this)

    }

}
(node.value == value) { return true

    node = node.next!!
}

while (node != head) {
  if (old == null) {
    newEntry = ChainEntry(key, value)
    map.put(key, newEntry)
    newEntryaddToEnd()
    return null
  } else {
    return old.setValue(value)
  }
}

return null


override fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()

override fun get(key: K): V? = map.get(key)

实际 override fun put(key: K, value: V): V? {
  checkIsMutable()

  val old = map.get(key)
  if (old == null) {
    val newEntry = ChainEntry(key, value)
    map.put(key, newEntry)
    newEntry.addToEnd()
    return null
  } else {
    return old.setValue(value)
  }
}

实际 override fun remove(key: K): V? {
  checkIsMutable()

  val entry = map.remove(key)
  if (entry != null) {
    entry.remove()
    return entry.value
  }
  return null
}

实际 override val size: Int get() = map.size

internal override fun checkIsMutable() {
  if (isReadOnly) throw UnsupportedOperationException()
}

/**
 * Constructs the specialized implementation of
 * [LinkedHashMap] with [String] keys, which stores the keys as properties of
 * JS object without housing them.
 */
public fun <V> linkedStringMapOf(vararg pairs: Pair<String, V>): LinkedHashMap<String, V> {
  return LinkedHashMap<String, V>(stringMapOf<Any>()).apply { putAll(pairs) }
}

/**
 * Creates a new instance of the specialized implementation of
 * [LinkedHashSet] with [String] elements, which stores the keys as properties of
 * JS object without housing them.
 */
public fun linkedStringSetOf(vararg elements: String): LinkedHashSet<String> {
  return LinkedHashSet(linkedStringMapOf<Any>()).apply { addAll(elements) }
}

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin

import kotlin.contracts.*

@kotlin.internal.InlineOnly
public actual inline fun <R> synchronized(lock: Any, block: () -> R): R {
  contract {
    callsInPlace(block, InvocationKind.EXACTLY_ONCE)
  }
  return block()
}

/**
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.io

internal abstract class BaseOutput {
  open fun println() {
    print("\n")
  }

  open fun println(message: Any?) {
    print(message)
    println()
  }

  abstract fun print(message: Any?)

  open fun flush() {}
}

/** JsName used to make the declaration available outside of module to test it*/
@JsName("NodeJsOutput")
internal class NodeJsOutput(val outputStream: dynamic) : BaseOutput() {
  override fun print(message: Any?) {
    val messageString = String(message)
    outputStream.write(messageString)
  }
}

/** JsName used to make the declaration available outside of module to test it*/
@JsName("OutputToConsoleLog")
internal class OutputToConsoleLog : BaseOutput() {
  override fun println(message: Any?) {
    console.log(message)
  }
}


```kotlin
console.log(message) { println(message) }

override fun println() {
    console.log("")
}

override fun println() {
    console.log("")
}

/** JsName used to make the declaration available outside of module to test it */
@JsName("BufferedOutput")
internal open class BufferedOutput : BaseOutput() {
    override fun print(message: Any?) {
        buffer += String(message)
        flush()
    }

    override fun flush() {
        buffer = ""
    }
}

/** JsName used to make the declaration available outside of module to test it */
@JsName("BufferedOutputToConsoleLog")
internal class BufferedOutputToConsoleLog : BufferedOutput() {
    override fun print(message: Any?) {
        var s = String(message)
        val i = s.nativeLastIndexOf("\n", 0)
        if (i >= 0) {
            buffer += s.substring(0, i)
            flush()
            s = s.substring(i + 1)
        }
        buffer += s
    }

    override fun flush() {
        console.log(buffer)
        buffer = ""
    }
}

/** JsName used to make the declaration available outside of module to test it */
@JsName("output")
internal var output = run {
    val isNode: Boolean = js("typeof process !== 'undefined' && process.versions && !!process.versions.node")
    if (isNode) NodeJsOutput(js("process.stdout")) else BufferedOutputToConsoleLog()
}

@kotlin.internal.InlineOnly
private inline fun String(value: Any?): String = js("String")(value)

/** Prints the line separator to the standard output stream. */
public actual fun println() {
    output.println()
}

/** Prints the given [message] and the line separator to the standard output stream. */
public actual fun println(message: Any?) {
    output.println(message)
}

/** Prints the given [message] to the standard output stream. */
public actual fun print(message: Any?) {
    output.print(message)
}
```

---

The document contains code snippets and comments written in Kotlin, a programming language. It seems to be part of a larger source code base, possibly related to a platform named Antelope or DNAC. The code includes classes and functions related to data handling and output operations, with some annotations for JavaScript compatibility. The document also references a license notice and copyright information, indicating that the code is open-source and governed by the Apache 2.0 license.
Kotlin official documentation
Element.appendElement(name: String, noinline init: Element.() -> Unit): Element = this.newAppendElement(name, init)*/

"/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.dom

import org.w3c.dom.Element

/**
 * Returns true if the element has the given CSS class style in its 'class' attribute
 */
@LowPriorityInOverloadResolution
@Deprecated(
    message = "This API is moved to another package, use 'kotlinx.dom.hasClass' instead.",
    replaceWith = ReplaceWith("this.hasClass(cssClass)"),
    warningSince = "1.4.4")
inline fun Element.hasClass(cssClass: String): Boolean = this.newHasClass(cssClass)

/**
 * Adds CSS class to element.
 * Has no effect if all specified classes are already in class attribute of the element
 */
@LowPriorityInOverloadResolution
@Deprecated(
    message = "This API is moved to another package, use 'kotlinx.dom.addClass' instead.",
    replaceWith = ReplaceWith("this.addClass(cssClasses)"),
    warningSince = "1.4.4")
inline fun Element.addClass(vararg cssClasses: String): Boolean = this.newAddClass(*cssClasses)

/**
 * Removes all [cssClasses] from element.
 * Has no effect if all specified classes are missing in class attribute of the element
 */
@LowPriorityInOverloadResolution
@Deprecated(
    message = "This API is moved to another package, use 'kotlinx.dom.removeClass' instead.",
    replaceWith = ReplaceWith("this.removeClass(cssClasses)"),
    warningSince = "1.4.4")
inline fun Element.removeClass(vararg cssClasses: String): Boolean = this.newRemoveClass(*cssClasses)

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.dom

import org.w3c.dom.Element

/**
 * Gets a value indicating whether this node is a TEXT_NODE or a CDATA_SECTION_NODE.
 */
@LowPriorityInOverloadResolution
@Deprecated(
    message = "This API is moved to another package, use 'kotlinx.dom.isText' instead.",
    replaceWith = ReplaceWith("this.isText"),
    warningSince = "1.4.4")
public val Node.isText: Boolean
    inline get() = this.newIsText

/**
 * Gets a value indicating whether this node is an [Element].
 */
@LowPriorityInOverloadResolution
@Deprecated(
    message = "This API is moved to another package, use 'kotlinx.dom.isElement' instead.",
    replaceWith = ReplaceWith("this.isElement"),
    warningSince = "1.4.4")
public val Node.isElement: Boolean
    inline get() = this.newIsElement

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.w3c.dom.events

public fun EventListener(handler: (Event) -> Unit): EventListener = EventListenerHandler(handler)

private class EventListenerHandler(private val handler: (Event) -> Unit) : EventListener {
    public override fun handleEvent(event: Event) {
        handler(event)
    }

    public override fun toString(): String = "EventListenerHandler($handler)"
}

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.w3c.dom

public external interface ItemArrayLike<out T> {
    val length: Int
    fun item(index: Int): T?
}

/**
 * Returns the view of this `ItemArrayLike<T>` collection as `List<T>`
 */
public fun <T> ItemArrayLike<T>.asList(): List<T> = object : AbstractList<T>() {

    override val size: Int get() = this@asList.length

    override fun get(index: Int): T = when (index)
        { in 0..lastIndex -> this@asList.item(index).unsafeCast<T>()
        else -> throw IndexOutOfBoundsException("index $index is not in range [0..$lastIndex]"
        )
    }
}

Open Source Used In DNAC Platform Antelope 14046
Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
by attaching `keySelector: (T) -> K` function to a source of elements. To get an instance of `Grouping` use one of `groupingBy` extension functions:

```kotlin
* - [Iterable.groupingBy](#extension-functions)
* - [Sequence.groupingBy](#extension-functions)
* - [Array.groupingBy](#extension-functions)
* - [CharSequence.groupingBy](#extension-functions)
```

For the list of group-and-fold operations available, see the [extension functions](#extension-functions) for `Grouping`.

```kotlin
Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially. passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
```

```kotlin
fun sourceIterator(): Iterator<T> /** Returns an [Iterator] over the elements of the source of this grouping. */

@SinceKotlin("1.1")
public interface Grouping<T, out K> {
    /** Returns an [Iterator] over the elements of the source of this grouping. */
    fun sourceIterator(): Iterator<T>
    /** Extracts the key of an [element]. */
    fun keyOf(element: T): K
}

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially.
 * passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
 * The key for each element is provided by the [Grouping.keyOf] function.
 * @param operation function is invoked on each element with the following parameters:
 *  - `key`: the key of the group this element belongs to;
 *  - `accumulator`: the current value of the accumulator of the group, can be `null` if it's the first `element` encountered in the group;
 *  - `element`: the element from the source being accumulated;
 * @param first: indicates whether it's the first `element` encountered in the group.
 * @return a [Map] associating the key of each group with the result of aggregation of the group elements.
 * @sample samples.collections.Grouping.aggregateByRadix
 */
@SinceKotlin("1.1")
public inline fun <T, K, R> Grouping<T, K>.aggregate(
    operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R
): Map<K, R> {
    return aggregateTo(mutableMapOf<K, R>(), operation)
}

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in the given [destination] map.
 * The key for each element is provided by the [Grouping.keyOf] function.
 * If the [destination] map already has a value corresponding to some key, then the elements being aggregated for that key are never considered as `first`.
 * @param destination a function that provides an initial value of accumulator for each group.
 *  It's invoked with parameters:
 *  - `key`: the key of the group;
 *  - `element`: the first element being encountered in that group.
 * @param operation a function that is invoked on each element with the following parameters:
 *  - `key`: the key of the group this element belongs to;
 *  - `accumulator`: the current value of the accumulator of the group;
 *  - `element`: the element from the source being accumulated;
 * @return the [destination] map associating the key of each group with the result of aggregation of the group elements.
 * @sample samples.collections.Grouping.aggregateByRadixTo
 */
@SinceKotlin("1.1")
public inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.aggregateTo(
    destination: M,
    operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R
): M {
    for (e in this.sourceIterator()) {
        val key = keyOf(e)
        val accumulator = destination[key]
        destination[key] = operation(key, accumulator, e, accumulator == null && !destination.containsKey(key))
    }
    return destination
}

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in a new map.
 * An initial value of accumulator is provided by [initialValueSelector] function.
 * @param initialValueSelector a function that provides an initial value of accumulator for each group.
 *  It's invoked with parameters:
 *  - `key`: the key of the group;
 *  - `element`: the first element being encountered in that group.
 * @param operation a function that is invoked on each element with the following parameters:
 *  - `key`: the key of the group this element belongs to;
 *  - `accumulator`: the current value of the accumulator of the group;
 *  - `element`: the element from the source being accumulated.
 * @return a [Map] associating the key of each group with the result of accumulating the group elements.
 * @sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue
 */
@SinceKotlin("1.1")
public inline fun <T, K, R : Any> Grouping<T, K>.fold(
    initialValueSelector: (key: K, element: T) -> R,
    operation: (key: K, accumulator: R, element: T) -> R
): Map<K, R> =
    @Suppress("UNCHECKED_CAST")
    aggregate { key, acc, e -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }
}

```
provides an initial value of accumulator for each group. It's invoked with parameters:
- `key`: the key of the group;
- `element`: the first element being encountered in that group.

If the [destination] map already has a value corresponding to some key, that value is used as an initial value of the accumulator for that group and the [initialValueSelector] function is not called for that group.

The current value of the accumulator of the group is the element from the source being accumulated.

@return the [destination] map associating the key of each group with the result of accumulating the group elements.

@sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValueTo


    aggregateTo(destination) { key, acc, e, first -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially, passing the previously accumulated value and the current element as arguments, and stores the results in a new map. An initial value of accumulator is the same for each group.
 * The current value of the accumulator of the group is the element from the source being accumulated.
 * @return a [Map] associating the key of each group with the result of accumulating the group elements.
 */
@sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue


    aggregate { _, acc, e, first -> operation(if (first) initialValue else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group sequentially starting from the second element of the group, passing the previously accumulated value and the current element as arguments, and stores the results in a new map. An initial value of accumulator is the first element of the group.
 * The current value of the accumulator of the group is the element from the source being accumulated.
 * @return a [Map] associating the key of each group with the result of accumulating the group elements.
 */
@sample samples.collections.Grouping.reduceByMaxVowels

@public inline fun <S, T : S, K> Grouping<T, K>.reduce(operation: (key: K, accumulator: S, element: T) -> S): Map<K, S> =

    aggregate { key, acc, e, first -> operation(key, if (first) e else operation(key, acc as S, e)) }


    aggregateTo(destination) { key, acc, e, first -> operation(key, if (first) initialValue else acc as R, e) }

/**
 * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group sequentially starting from the second element of the group, passing the previously accumulated value and the current element as arguments, and stores the results in a new map. An initial value of accumulator is the first element of the group.
 * The current value of the accumulator of the group is the element from the source being accumulated.
 * @return a [Map] associating the key of each group with the result of accumulating the group elements.
 */
accumulator is the first element of the group.\n *\n * If the [destination] map already has a value corresponding to
the key of some group,\n * that value is used as an initial value of the accumulator for that group and the first
element of that group is also\n * subjected to the [operation].\n\n * @param operation a function that is invoked on
each subsequent element of the group with the following parameters:\n * - `accumulator`: the current value of the
accumulator of the group;\n * - `element`: the element from the source being folded;\n *\n * @return the
[destination] map associating the key of each group with the result of accumulating the group elements.\n *
@sample samples.collections.Grouping.reduceByMaxVowelsTo\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <S,
T : S, K, M : MutableMap<in K, S>> Grouping<T, K>.reduceTo(\n destination: M,\n operation: (key: K,
accumulator: S, element: T) -> S\n): M =\n aggregateTo(destination) { key, acc, e, first ->\n
@Suppress(\"UNCHECKED_CAST\")\n
if (first) e else operation(key, acc as S, e)\n }\n\n\n/**\n * Groups
elements from the [Grouping] source by key and counts elements in each group to the given [destination] map.\n *\n
* If the [destination] map already has a value corresponding to the key of some group,\n * that value is used as an
initial value of the counter for that group.\n *\n * @return the [destination] map associating the key of each group
with the count of elements in the group.\n *\n * @sample samples.collections.Grouping.groupingByEachCount\n
*/\n@SinceKotlin(\"1.1\")\npublic fun <T, K, M : MutableMap<in K, Int>> Grouping<T,
K>.eachCountTo(destination: M): M =\n foldTo(destination, 0) { acc, _ -> acc + 1 }\n\n/*\n/**\n * Groups
elements from the [Grouping] source by key and sums values provided by the [valueSelector] function for elements
in each group\n * to the given [destination] map.\n *\n *\n * If the [destination] map already has a value
corresponding to the key of some group,\n * that value is used as an initial value of the sum for that group.\n *\n *
@return the [destination] map associating the key of each group with the sum of elements in the group.\n
*/\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K, M : MutableMap<in K, Int>> Grouping<T,
K>.eachSumOfTo(destination: M, valueSelector: (T) -> Int): M =\n
foldTo(destination, 0) { acc, e -> acc +
valueSelector(e)}\n*/\n\n\n/*\n// TODO: sum by long and by double overloads\n\npublic inline fun <T, K, M :
MutableMap<in K, Long>> Grouping<T, K>.sumEachByLongTo(destination: M, valueSelector: (T) -> Long): M
=\n
foldTo(destination, 0L) { acc, e -> acc + valueSelector(e)}\n\npublic inline fun <T, K> Grouping<T,
K>.sumEachByLong(valueSelector: (T) -> Long): Map<K, Long> =\n
fold(0L) { acc, e -> acc +
valueSelector(e)}\n\npublic inline fun <T, K, M : MutableMap<in K, Double>> Grouping<T,
K>.sumEachByDoubleTo(destination: M, valueSelector: (T) -> Double): M =\n
foldTo(destination, 0.0) { acc, e
-> acc + valueSelector(e)}\n\npublic inline fun <T, K> Grouping<T, K>.sumEachByDouble(valueSelector: (T) ->
Double): Map<K, Double> =\n
fold(0.0) { acc, e -> acc + valueSelector(e)}\n*/\n","/*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Exposes
to Kotlin.\n */\n@Deprecated(\"Use top-level functions from kotlin.math package instead.\", level =
DeprecationLevel.ERROR)\npublic external object Math {\n @Deprecated(\"Use kotlin.math.PI instead.\",
ReplaceWith(\"PI\", \"kotlin.math.PI\"), level = DeprecationLevel.ERROR)\n public val PI: Double\n
@Deprecated(\"Use Random.nextDouble instead\", ReplaceWith(\"kotlin.random.Random.nextDouble()\",
\"kotlin.random.Random\"), level = DeprecationLevel.ERROR)\n public fun random(): Double\n
@Deprecated(\"Use kotlin.math.abs instead.\", ReplaceWith(\"abs(value)\", \"kotlin.math.abs\"), level =
DeprecationLevel.ERROR)\n public fun abs(value: Double): Double\n @Deprecated(\"Use kotlin.math.acos
instead.\", ReplaceWith(\"acos(value)\", \"kotlin.math.acos\"), level = DeprecationLevel.ERROR)\n public fun
acos(value: Double): Double\n @Deprecated(\"Use kotlin.math.asin instead.\", ReplaceWith(\"asin(value)\",
\"kotlin.math.asin\"), level = DeprecationLevel.ERROR)\n public fun asin(value: Double): Double\n
@Deprecated(\"Use kotlin.math.atan instead.\", ReplaceWith(\"atan(value)\", \"kotlin.math.atan\"), level =
DeprecationLevel.ERROR)\n public fun atan(value: Double): Double\n @Deprecated(\"Use kotlin.math.atan2
instead.\", ReplaceWith(\"atan2(y, x)\", \"kotlin.math.atan2\"), level = DeprecationLevel.ERROR)\n public fun
atan2(y: Double, x: Double): Double\n @Deprecated(\"Use kotlin.math.cos instead.\",
ReplaceWith(\"cos(value)\", \"kotlin.math.cos\"), level = DeprecationLevel.ERROR)\n public fun cos(value:

Open Source Used In DNAC Platform Antelope 14050


Open Source Used In DNAC Platform Antelope 14051

```kotlin
@PublishedApi
internal fun trunc(value: Number): Double
@PublishedApi
internal fun sign(value: Number): Double

@PublishedApi
internal fun sinh(value: Double): Double
@PublishedApi
internal fun cosh(value: Double): Double
@PublishedApi
internal fun tanh(value: Double): Double
@PublishedApi
internal fun asinh(value: Double): Double
@PublishedApi
internal fun acosh(value: Double): Double

@PublishedApi
internal fun hypot(x: Double, y: Double): Double
@PublishedApi
internal fun expm1(value: Double): Double
@PublishedApi
internal fun log10(value: Double): Double
@PublishedApi
internal fun log2(value: Double): Double
@PublishedApi
internal fun log1p(value: Double): Double

@PublishedApi
internal fun clz32(value: Int): Int

/**
 * Returns the smaller of two values.
 */
@Suppress("DEPRECATION_ERROR")
@Deprecated("Use minOf or kotlin.math.min instead", ReplaceWith("minOf(a, b)"), level = DeprecationLevel.ERROR)
public fun Math.min(a: Long, b: Long): Long = if (a <= b) a else b

/**
 * Returns the greater of two values.
 */
@Suppress("DEPRECATION_ERROR")
@Deprecated("Use maxOf or kotlin.math.max instead", ReplaceWith("maxOf(a, b)"), level = DeprecationLevel.ERROR)
public fun Math.max(a: Long, b: Long): Long = if (a >= b) a else b
```

---

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
 * contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file.
 */

package kotlin.js

/**
 * An interface for indexing access to a collection of
 * key-value pairs, where type of key is [String] and type of value is [Any?][Any].
 */
public external interface Json
```
public fun json(vararg pairs: Pair<String, Any?>): Json {
    return this.map { (k, v) -> v.toString() } as Json
}

/**
 * Adds key-value pairs from 
[other] to
this.  
 * Returns the original receiver.
 */
public fun Json.add(other: Json): Json {
    keys.forEach { key ->
        this[key] = other[key] as Any
    }
    return this
}

/**
 * Exposes the JavaScript [JSON

 * @param o an object
 * @param replacer a function that is called with key and value
 * @param space a string
 */
@file:Suppress("NOT_DOCUMENTED")
package kotlin

import kotlin.internal.InlineOnly
import kotlin.js.Math as nativeMath

// region ================ Double Math

/** Computes the sine of the angle [x] given in radians.
 * Special cases:
 *   - `sin(NaN)` is `NaN`
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sin(x: Double): Double =
nativeMath.sin(x)

/** Computes the cosine of the angle [x] given in radians.
 * Special cases:
 *   - `cos(NaN)` is `NaN`
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun cos(x: Double): Double =
nativeMath.cos(x)

/** Computes the tangent of the angle [x] given in radians.
 * Special cases:
 *   - `tan(NaN)` is `NaN`
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun tan(x: Double): Double =
nativeMath.tan(x)

/**
 * Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond
 * to the rectangular coordinates `(x, y)` by computing the arc tangent of the value
 * `y`/`x`. The returned value is an angle in the range from `-PI/2` to `PI/2` radians.
 * Special cases:
 *   - `atan2(0, x)` is `PI/2` for `x < 0`.
 *   - `atan2(0, x)` is `-PI/2` for `x > 0`.
 *   - `atan2(0, 0)` is `0`.
 *   - `atan2(y, 0)` is `PI/2` for `y > 0`.
 *   - `atan2(y, 0)` is `-PI/2` for `y < 0`.
 *   - `atan2(y, x)` is `tan(y/x)` for `y != 0`.
 *   - `atan2(y, x)` is `NaN` for `x = 0`.
 *   - `atan2(y, x)` is `atan(y/x)` for `x > 0`.
 *   - `atan2(y, x)` is `-atan(x/y)` for `x < 0`.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun atan2(y: Double, x: Double): Double =
nativeMath.atan2(y, x)

/**
 * Computes the hyperbolic sine of the value [x].
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sinh(x: Double) =
nativeMath.sinh(x)

/**
 * Computes the hyperbolic cosine of the value [x].
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun cosh(x: Double) =
nativeMath.cosh(x)

/**
 * Computes the hyperbolic tangent of the value [x].
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun tanh(x: Double) =
nativeMath.tanh(x)
nativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\nnativeMath.sinh(x)\n\n\n\n\n
nativeMath.ceil(x).unsafeCast<Double>() // TODO: Remove unsafe cast after removing public js.math\n\n* * Rounds the given value [x] to an integer towards negative infinity.\n* * \n\n  * @return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.\n\n  * Special cases:\n  * - `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n  * 
\n  * @see absoluteValue extension property for [Double] \n\n* * \n\n  * Rounds the given value [x] to an integer towards zero.\n* * \n\n  * @return the value [x] having its fractional part truncated.\n* * \n\n  * Special cases:\n  * - `truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n\n  * see absoluteValue extension property for [Double] \n\n* * \n\n  * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.\n* * \n\n  * @see absoluteValue extension property for [Double] \n\n* * \n\n  * Returns the absolute value of the given value [x].\n* * \n\n  * @see abs function \n\n* * \n\n  * Returns the sign of the given value [x]:\n  *   - `-1.0` if the value is negative,\n  *   - `0.0` if the value is zero,\n  *   - `1.0` if the value is positive.\n* * \n\n  * Special case:\n  *   - `sign(NaN)` is `NaN`.\n\n  * see absoluteValue extension property for [Double] \n\n* * \n\n  * Returns the smaller of two values.\n* * \n\n  * If either value is `NaN`, then the result is `NaN`.\n* * \n\n  * @see min function \n\n* * \n\n  * Returns the greater of two values.\n* * \n\n  * If either value is `NaN`, then the result is `NaN`.\n* * \n\n  * @see max function \n\n* * \n\n  * Raises this value to the power [x].\n* * \n\n  * Special cases:\n  * - `b.pow(0.0)` is `1.0`.\n  * - `b.pow(1.0)` is `b`.\n  * - `b.pow(NaN)` is `NaN`.\n  * - `NaN.pow(x)` is `NaN` for `x != 0.0`.\n* * \n\n  * Special case:\n  * - `NaN.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer.\n* * \n\n  * @see pow function \n\n* * \n\n  * Returns the ulp (unit in the last place) of this value.\n* * \n\n  * Special cases:\n  * - `NaN.ulp` is `NaN`.\n  * - `x.ulp` is `+Inf` when `x` is `+Inf` or `-Inf` or `0.0`.\n  * - `-Inf.ulp` is `+Inf` when `x` is `+Inf` or `-Inf` or `0.0`.\n* * \n\n  * Returns the ulp (unit in the last place) of this value.\n* * \n\n  * @see ulp function \n\n* * \n\n  * Returns the value nearest to this value in direction of positive infinity.\n* * \n\n  * @see nextUp function \n\n* * \n\n  * Returns the value nearest to this value in direction of negative infinity.\n* * \n\n  * @see nextDown function \n\n* * \n\n  * The implementation of this method is as close as possible to the C99 standard.\n* * \n\n  * @see DoubleMIN_VALUE in DNAC Platform Antelope 14054
value in direction from this value towards the value [to].

\* Special cases:
\* - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`
\* - `'+x.nextTowards(x)'` is `x`
\* - `'-x.nextTowards(x)'` is `-x`

\* Special cases:
\* - `x.roundToLong()` is `Long.MAX_VALUE` when `x > Long.MAX_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MIN_VALUE`
\* - `'-x.roundToLong()'` is `Long.MAX_VALUE` when `x > Long.MIN_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MAX_VALUE`

\* - `x.nextUp()` is `x`
\* - `x.nextDown()` is `-x`

\* Special cases:
\* - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`
\* - `'+x.nextTowards(x)'` is `x`
\* - `'-x.nextTowards(x)'` is `-x`

\* Special cases:
\* - `x.roundToLong()` is `Long.MAX_VALUE` when `x > Long.MAX_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MIN_VALUE`
\* - `'-x.roundToLong()'` is `Long.MAX_VALUE` when `x > Long.MIN_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MAX_VALUE`

\* - `x.nextUp()` is `x`
\* - `x.nextDown()` is `-x`

\* Special cases:
\* - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`
\* - `'+x.nextTowards(x)'` is `x`
\* - `'-x.nextTowards(x)'` is `-x`

\* Special cases:
\* - `x.roundToLong()` is `Long.MAX_VALUE` when `x > Long.MAX_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MIN_VALUE`
\* - `'-x.roundToLong()'` is `Long.MAX_VALUE` when `x > Long.MIN_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MAX_VALUE`

\* - `x.nextUp()` is `x`
\* - `x.nextDown()` is `-x`

\* Special cases:
\* - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`
\* - `'+x.nextTowards(x)'` is `x`
\* - `'-x.nextTowards(x)'` is `-x`

\* Special cases:
\* - `x.roundToLong()` is `Long.MAX_VALUE` when `x > Long.MAX_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MIN_VALUE`
\* - `'-x.roundToLong()'` is `Long.MAX_VALUE` when `x > Long.MIN_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MAX_VALUE`

\* - `x.nextUp()` is `x`
\* - `x.nextDown()` is `-x`

\* Special cases:
\* - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`
\* - `'+x.nextTowards(x)'` is `x`
\* - `'-x.nextTowards(x)'` is `-x`

\* Special cases:
\* - `x.roundToLong()` is `Long.MAX_VALUE` when `x > Long.MAX_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MIN_VALUE`
\* - `'-x.roundToLong()'` is `Long.MAX_VALUE` when `x > Long.MIN_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MAX_VALUE`

\* - `x.nextUp()` is `x`
\* - `x.nextDown()` is `-x`

\* Special cases:
\* - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`
\* - `'+x.nextTowards(x)'` is `x`
\* - `'-x.nextTowards(x)'` is `-x`

\* Special cases:
\* - `x.roundToLong()` is `Long.MAX_VALUE` when `x > Long.MAX_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MIN_VALUE`
\* - `'-x.roundToLong()'` is `Long.MAX_VALUE` when `x > Long.MIN_VALUE`
\* - `'+x.roundToLong()'` is `Long.MIN_VALUE` when `x < Long.MAX_VALUE`

\* - `x.nextUp()` is `x`
\* - `x.nextDown()` is `-x`
public actual inline fun cosh(x: Float): Float =
nativeMath.cosh(x.toDouble()).toFloat()

/**
 * Computes the inverse hyperbolic sine of the value [x].
 * Special cases:
 * The returned value is `y` such that `sinh(y) == x`
 * The sinh function is `NaN` when `x < -1` or `x > 1`
 */
public actual inline fun sinh(x: Float): Float =
nativeMath.sinh(x.toDouble()).toFloat()

/**
 * Computes the inverse hyperbolic cosine of the value [x].
 * Special cases:
 * The returned value is positive `y` such that `cosh(y) == x`
 * The cosh function is `NaN` when `x < 0`
 */
public actual inline fun acosh(x: Float): Float =
nativeMath.acosh(x.toDouble()).toFloat()

/**
 * Computes the inverse hyperbolic tangent of the value [x].
 * Special cases:
 * The returned value is `y` such that `tanh(y) == x`
 * The tanh function is `NaN` when `x < -1` or `x > 1`
 */
public actual inline fun atanh(x: Float): Float =
nativeMath.atanh(x.toDouble()).toFloat()
mathematical integer.

@SinceKotlin("1.2")
@InlineOnly
public actual inline fun ceil(x: Float): Float =
nativeMath.ceil(x.toDouble()).toFloat()

/**
 * Rounds the given value [x] to an integer towards negative
 * infinity.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun floor(x: Float): Float =
nativeMath.floor(x.toDouble()).toFloat()

/**
 * Rounds the given value [x] to an integer towards zero.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun truncate(x: Float): Float =
truncate(x.toDouble()).toFloat()

/**
 * Rounds the given value [x] towards the closest integer with
 * ties rounded towards even integer.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun round(x: Float): Float =
round(x.toDouble()).toFloat()

/**
 * Returns the absolute value of the given value [x].
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun abs(x: Float): Float =
nativeMath.abs(x.toDouble()).toFloat()

/**
 * Returns the sign of the given value [x]:
 *   - `-1.0` if the value is negative,
 *   - zero if the value is zero,
 *   - `1.0` if the value is positive
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun sign(x: Float): Float =
nativeMath.sign(x.toDouble()).toFloat()

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun min(a: Float, b: Float): Float =
nativeMath.min(a, b)

/**
 * Returns the greater of two values.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun max(a: Float, b: Float):
Float = nativeMath.max(a, b)

// extensions

/**
 * Raises this value to the power [x].
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun Float.pow(x: Float):
Float = nativeMath.pow(this.toDouble(), x.toDouble()).toFloat()

/**
 * Raises this value to the integer power [n].
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun Float.pow(n: Int): Float =
nativeMath.pow(this.toDouble(), n.toDouble()).toFloat()

/**
 * Returns the absolute value of this value.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline val Float.absoluteValue: Float get() =
nativeMath.abs(this.toDouble()).toFloat()

/**
 * Returns the sign of this value:
 *   - `-1.0` if the value is negative,
 *   - zero if the value is zero,
 *   - `1.0` if the value is positive
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline val Float.sign: Float get() =
nativeMath.sign(this.toDouble()).toFloat()

/**
 * Returns this value with the sign bit same as of the [sign]
 * value.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun Float.withSign(sign: Float): Float =
this.toDouble().withSign(sign.toDouble()).toFloat()

/**
 * Returns this value with the sign bit same as of the [sign]
 * value.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun Float.withSign(sign: Int): Float =
this.toDouble().withSign(sign.toDouble()).toFloat()

/**
 * Rounds this value to the nearest integer and converts the result to
 * `Int`. Ties are rounded towards positive infinity.
 */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun Float.roundToInt(): Int =
nativeMath.roundToInt()
@SinceKotlin("1.2")
@InlineOnly
public actual fun Float.roundToLong(): Long =

doubleToLong()

/**
 * Returns the absolute value of the given value \([n]\).
 * Special cases:\n * - 'abs(Int.MIN_VALUE)' is 'Int.MIN_VALUE' due to an overflow
 * @see absoluteValue extension property for [Int]\n */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun Int.abs(): Int =

// endregion

// region ================ Integer Math

/**
 * Returns the absolute value of the given value \([n]\).
 * Special cases:\n * - 'abs(Long.MIN_VALUE)' is 'Long.MIN_VALUE' due to an overflow
 * @see absoluteValue extension property for [Long]\n */
@SinceKotlin("1.2")
@InlineOnly
public actual inline fun Long.abs(): Long =

/**
 * Counts the number of set bits in the binary representation of this [Int] number.
 * @see Integer.bitCount() for an efficient implementation
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public actual fun Int.bitCount(): Int =

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.
 * @see Integer.numberOfTrailingZeros() for an efficient implementation
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("DEPRECATION_ERROR")
@kotlin.internal.InlineOnly
public actual inline fun Int.countLeadingZeros(): Int =

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.
 * @see Integer.numberOfTrailingZeros() for an efficient implementation
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public actual fun Int.countLeadingZeros(): Int =

Open Source Used In DNAC Platform Antelope 14058
.kotlin.js.Math.clz32(this)ln

* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Int] number.

*An@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)public actual fun
Int.countTrailingZeroBits(): Int =ln  // Hacker's Delight 5-4 algorithm for expressing countTrailingZeroBits with
countLeadingZeroBits n  Int.SIZE_BITS - (this or -this).inv().countLeadingZeroBits()ln

* Returns a number having a single bit set in the position of the most significant set bit of this [Int] number,\n* or zero, if this number is zero.

*An@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)public actual fun
Int.takeHighestOneBit(): Int =ln  if (this == 0) 0 else 1 shl(Int.SIZE_BITS - 1 - countLeadingZeroBits())ln

* Returns a number having a single bit set in the position of the least significant set bit of this [Int] number,\n* or zero, if this number is zero.

*An@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)public actual fun
Int.takeLowestOneBit(): Int =ln  // Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit\nthis and -thisln

* Rotates the binary representation of this [Int] number left by the specified [bitCount] number of
bits,\n* The most significant bits pushed out from the left side reenter the number as the least significant bits on the
right side.\n* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit
count:\n* number.rotateLeft(-n) == number.rotateRight(n)ln

* Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n* number.rotateLeft(n) == number.rotateLeft(n % 32)ln

*An@SinceKotlin("1.3")n@ExperimentalStdlibApi\npublic actual fun Int.rotateLeft(bitCount: Int): Int =
shl(bitCount) or ushr(Int.SIZE_BITS - bitCount)ln

* Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits,\n* The least significant bits pushed out from the right side reenter the number as the most significant bits on the
left side.\n* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit
count:\n* number.rotateRight(-n) == number.rotateLeft(n)ln

* Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n* number.rotateRight(n) == number.rotateRight(n % 32)ln

*An@SinceKotlin("1.3")n@ExperimentalStdlibApi\npublic actual fun Int.rotateRight(bitCount: Int): Int =ln
shl(Int.SIZE_BITS - bitCount) or ushr(bitCount)ln

* Counts the number of set bits in the binary
representation of this [Long] number.

*An@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)public actual fun
Long.countOneBits(): Int =ln  high.countOneBits() + low.countOneBits()ln

* Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.

*An@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)public actual fun
Long.countLeadingZeroBits(): Int =ln  when (val high = this.high) {\n  0 -> Int.SIZE_BITS +
  low.countLeadingZeroBits()ln
  else -> high.countLeadingZeroBits()ln
}ln

* Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.

*An@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)public actual fun
Long.countTrailingZeroBits(): Int =ln  when (val low = this.low) {\n  0 -> Int.SIZE_BITS +
high.countTrailingZeroBits()ln
  else -> low.countTrailingZeroBits()ln
}ln

* Returns a number having a single bit set in the position of the most significant set bit of this [Long] number,\n* or zero, if this number is zero.

*An@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)public actual fun
Long.takeHighestOneBit(): Long =ln  when (val high = this.high) {\n  0 -> Long(0, high.takeHighestOneBit())ln
  else -> Long(0, high.takeHighestOneBit())ln
}ln

* Returns a number having a single bit set in the position of the least significant set bit of this [Long] number,\n* or zero, if this number is zero.

*An@SinceKotlin("1.4")n@WasExperimental(ExperimentalStdlibApi::class)public actual fun
Long.takeLowestOneBit(): Long =ln  when (val low = this.low) {\n  0 -> Long(0, high.takeLowestOneBit())ln
  else -> Long(0, high.takeLowestOneBit())ln
}ln

* Rotates the binary representation of this [Long] number left by the specified [bitCount] number of bits,\n* The most significant bits pushed out from the left side reenter the number as the least significant bits on the
right side.\n* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit
count:\n* number.rotateLeft(-n) ==
number.rotateRight(n) * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally
 * number.rotateLeft(n) == number.rotateLeft(n % 64)

/*@SinceKotlin("1.3")@*/@ExperimentalStdlibApi
public actual fun Long.rotateLeft(bitCount: Int): Long {
    when {
        (bitCount and 31) != 0 ->
            val low = this.low
            val high = this.high
            val newLow = low.shl(bitCount) or high.ushr(-bitCount)
            val newHigh = high.shl(bitCount) or low.ushr(-bitCount)
            if ((bitCount and 32) == 0) Long(newLow, newHigh) else Long(newHigh, newLow)
        else -> (bitCount and 32) == 0
            this
            return
            return
    }
}

/**
 * Rotates the binary representation of this
 * [Long] number right by the specified [bitCount] number of bits.
 * The least significant bits pushed out from the
 * right side reenter the number as the most significant bits on the left side.
 * * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:

 * number.rotateRight(-n) ==
 * number.rotateLeft(n) * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally
 * number.rotateRight(n) == number.rotateRight(n % 64)

/*@SinceKotlin("1.3")@*/@ExperimentalStdlibApi
@Retention(value = AnnotationRetention.BINARY)
public actual class Promise<out T>(executor: (resolve: (T) -> Unit, reject: (Throwable) -> Unit) -> Unit) {
    public open fun <S>
        then(onFulfilled: ((T) -> S)?): Promise<S>
    public open fun <S>
        then(onFulfilled: ((T) -> S)?, onRejected: ((Throwable) -> S)?): Promise<S>
    public open fun <S>
        catch(onRejected: (Throwable) -> S?): Promise<S>
    companion object {
        public fun <S> all(promise: Array<out Promise<S>>): Promise<Array<out S>>
        public fun <S> race(promise: Array<out Promise<S>>): Promise<S>
        public fun reject(e: Throwable): Promise<Nothing>
        public fun resolve(e: S): Promise<S>
        public fun resolve(e: Promise<S>): Promise<S>
    }
}

fun Promise<Promise<T>>.then(
    onFulfilled: ((T) -> S)?
): Promise<S> {
    return this.unsafeCast<Promise<T>>().then(onFulfilled)
}

fun Promise<Promise<T>>.then(
    onFulfilled: ((T) -> S)?,
    onRejected: ((Throwable) -> S)?
): Promise<S> {
    return this.unsafeCast<Promise<T>>().then(onFulfilled, onRejected)
}

@Suppress("NOT_DOCUMENTED")
package kotlin

import kotlin.math.pow

internal actual fun defaultPlatformRandom(): Random =
    Random(js("(Math.random() * Math.pow(2, 32)) | 0").unsafeCast<Int>())

private val INV_2_26: Double = 2.0.pow(-26)
private val
    INV_2_53: Double = 2.0.pow(-53)
internal actual fun doubleFromParts(hi26: Int, low27: Int): Double =
    hi26 * INV_2_26 + low27 * INV_2_53

@Suppress("NOT_DOCUMENTED")
package kotlin.reflect

import findAssociatedObject

/**
 * The experimental marker for associated objects API.
 * * Any usage of a declaration annotated with
 * `@ExperimentalAssociatedObjects` must be accepted either by
 * annotating that usage with the [OptIn] annotation, e.g.
 * `@OptIn(ExperimentalAssociatedObjects::class)`, or by using the compiler argument `-Xopt-in=kotlin.reflect.ExperimentalAssociatedObjects`
 * * The experimental marker for associated objects API.
 * * Any usage of a declaration annotated with
 * `@ExperimentalAssociatedObjects` must be accepted either by
 * annotating that usage with the [OptIn] annotation, e.g.
 * `@OptIn(ExperimentalAssociatedObjects::class)`, or by using the compiler argument `-Xopt-in=kotlin.reflect.ExperimentalAssociatedObjects`
 */
@Retention(value = AnnotationRetention.BINARY)
public actual class RequiresOptIn {
    public fun <T, S> RequiresOptIn(level =
        RequiresOptIn.Level.ERROR,n@Retention(value = AnnotationRetention.BINARY)
    public actual fun doubleFromParts(hi26: Int, low27: Int): Double =
        hi26 * INV_2_26 + low27 * INV_2_53
    )
}
val intClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Int>>(), "Int", { it is Int })

val floatClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Float>>(), "Float", { it is Float })

val doubleClass = PrimitiveKClassImpl(js("Number").unsafeCast<JsClass<Double>>(), "Double", { it is Double })

val arrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<Array<*>>>(), "Array", { it is Array<*> })

val stringClass = PrimitiveKClassImpl(js("String").unsafeCast<JsClass<String>>(), "String", { it is String })

val throwableClass = PrimitiveKClassImpl(js("Error").unsafeCast<JsClass<Throwable>>(), "Throwable", { it is Throwable })

val booleanArrayClass = PrimitiveKClassImpl(js("BooleanArray").unsafeCast<JsClass<ByteArray>>(), "BooleanArray", { it is BooleanArray })

val charArrayClass = PrimitiveKClassImpl(js("CharArray").unsafeCast<JsClass<CharArray>>(), "CharArray", { it is CharArray })

val byteArrayClass = PrimitiveKClassImpl(js("Int8Array").unsafeCast<JsClass<ByteArray>>(), "ByteArray", { it is ByteArray })

val shortArrayClass = PrimitiveKClassImpl(js("Int16Array").unsafeCast<JsClass<ShortArray>>(), "ShortArray", { it is ShortArray })

val intArrayClass = PrimitiveKClassImpl(js("Int32Array").unsafeCast<JsClass<IntArray>>(), "IntArray", { it is IntArray })

val longArrayClass = PrimitiveKClassImpl(js("Array").unsafeCast<JsClass<LongArray>>(), "LongArray", { it is LongArray })

val floatArrayClass = PrimitiveKClassImpl(js("Float32Array").unsafeCast<JsClass<FloatArray>>(), "FloatArray", { it is FloatArray })

val doubleArrayClass = PrimitiveKClassImpl(js("Float64Array").unsafeCast<JsClass<DoubleArray>>(), "DoubleArray", { it is DoubleArray })

val functionClass = functionClass(arity: Int): KClassImpl<Any> {
    return functionClasses.get(arity) ?: run {
        val result = PrimitiveKClassImpl(js("Function").unsafeCast<JsClass<Any>>(), "Function

    } & 


}
e is LongArray -> PrimitiveClasses.longArrayClass
            e is FloatArray -> PrimitiveClasses.floatArrayClass
            e is DoubleArray -> PrimitiveClasses.doubleArrayClass
            e is KClass<*> -> KClass::class
            e is Array<*> -> PrimitiveClasses.arrayClass
            else -> {
                val constructor = js("Object").getPrototypeOf(e).constructor
                when {
                    constructor === js("Object") -> PrimitiveClasses.anyClass
                    constructor === js("Error") -> PrimitiveClasses.throwableClass
                    else -> {
                        val jsClass: JsClass<T> = constructor
                        getKClass1(jsClass)
                    }
                }
            }
        }
    }.unsafeCast<KClass<T>>()

@JsName("getKClass1")
internal fun <T : Any> getKClass1(jClass: JsClass<T>): KClass<T> {
    if (jClass === js("String")) return PrimitiveClasses.stringClass.unsafeCast<KClass<T>>()

    val metadata = jClass.asDynamic().`$metadata$`
    return if (metadata != null) {
        if (metadata.`$kClass$` == null) {
            val kClass = SimpleKClassImpl(jClass)
            metadata.`$kClass$` = kClass
            kClass
        } else {
            metadata.`$kClass$`
        }
    } else {
        SimpleKClassImpl(jClass)
    }
} /*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.js

/**
 */
@Suppress("NOT_DOCUMENTED")
public external class RegExp(pattern: String, flags: String? = definedExternally) {

    public fun test(str: String): Boolean

    public fun exec(str: String): RegExpMatch?

    public override fun toString(): String

    /**
     * The lastIndex is a read/write integer property of regular expressions that specifies the index at which to start the next match.
     */
    public var lastIndex: Int

    public val global: Boolean
    public val ignoreCase: Boolean
    public val multiline: Boolean
}

/**
 * Resets the regular expression so that subsequent [RegExp.test] and [RegExp.exec] calls will match starting with the beginning of the input string.
 */
public fun RegExp.reset() {
    lastIndex = 0
}

// TODO: Inherit from array or introduce asArray() extension
/**
 * Represents the return value of [RegExp.exec].
 */
@Suppress("NOT_DOCUMENTED")
public external interface RegExpMatch {
    public val index: Int

    public val input: String

    public val length: Int
}

/**
 * Returns the entire text matched by [RegExp.exec] if the [index] parameter is 0, or the text matched by each capturing parenthesis at the given index.
 */

/**
 * Converts the result of [RegExp.exec] to an array where the first element contains the entire matched text and each subsequent element is the text matched by each capturing parenthesis.
 */
public inline fun RegExpMatch.asArray(): Array<out String?> = unsafeCast<Array<out String>>()

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.text

/**
 * A mutable sequence of characters.
 *
 * String builder can be used to efficiently perform multiple string manipulation operations.
 * A mutable sequence of characters.
 */
public actual class StringBuilder actual constructor(content: String) : Appendable, CharSequence {

    * * * * * Constructs an empty string builder with the specified initial [capacity].


*"n * In Kotlin/JS implementation of StringBuilder the initial capacity has no effect on the further performance of operations.*"n *"n actual constructor(capacity: Int): this() {"n    }\n
/** Constructs a string builder that contains the same characters as the specified [content] char sequence. */
actual constructor(content: CharSequence): this(content.toString()) {"n
/** Constructs an empty string builder. */
actual constructor():
this("")\n
private var string: String = if (content !== undefined) content else "null"\n
actual override fun append(value: Char): StringBuilder {"n    string += value\n    return this\n\n/**\n* Reverses the contents of this string builder and returns this instance.\n*\n* Surrogate pairs included in this string builder are treated as single characters.\n* Therefore, the order of the high-low surrogates is never reversed.\n*\n* Note that the reverse operation may produce new surrogate pairs that were unpaired low-surrogates and high-surrogates before the operation.\n* For example, reversing `"\uDC00\uD800"` produces `"\uD800\uDC00"` which is a valid surrogate pair.\n*\nactual fun reverse(): StringBuilder {"n    var reversed = string.reverse()\n    var index = string.length - 1\n    while (index >= 0) {"n        val high = string[index--]\n        if (high.isHighSurrogate() && index >= 0) {"n            val low = string[index--]\n            if (low.isLowSurrogate()) {"n                reversed += low
            } else {
                reversed += low + high
            }
        }
        string = reversed
    }
    return this
\n/**\n* The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n* and then that string was appended to this string builder.\n*\nactual fun append(value: Char): StringBuilder {"n    string += value\n    return this\n\n/**\n* Appends the specified string [value] to this string builder and returns this instance.\n*\n* If [value] is `null`, then the four characters `"null"` are appended.\n*\nactual fun append(value: String): StringBuilder {"n    this.string += value\n    return this\n\n/**\n* The capacity is the maximum length this string builder can have before an allocation occurs.\n*\n* In Kotlin/JS implementation of StringBuilder the value returned from this method may not indicate the actual size of the backing storage.\n*\n* @SinceKotlin("1.3.3")\n@Experimental(ExperimentalStdlibApi::class)\nactual fun ensureCapacity(minimumCapacity: Int) {"n
Open Source Used In DNAC Platform Antelope 14065
@WasExperimental(ExperimentalStdlibApi::class) actual fun indexOf(string: String, startIndex: Int): Int = this.string.asDynamic().indexOf(string, startIndex)

@WasExperimental(ExperimentalStdlibApi::class) actual fun lastIndexOf(string: String): Int = this.string.asDynamic().lastIndexOf(string)

@WasExperimental(ExperimentalStdlibApi::class) actual fun lastIndexOf(string: String, startIndex: Int): Int {
    if (string.isEmpty() && startIndex < 0) return -1
    return this.string.asDynamic().lastIndexOf(string, startIndex)
}

@WasExperimental(ExperimentalStdlibApi::class) actual fun insert(index: Int, value: CharSequence?): StringBuilder {
    if (string.isEmpty() && startIndex < 0) return -1
    return this.string.asDynamic().lastIndexOf(string, startIndex)
}

@WasExperimental(ExperimentalStdlibApi::class) actual fun insert(index: Int, value: CharSequence): StringBuilder {
    AbstractList.checkPositionIndex(index, length)

    string = string.substring(0, index) + value.concatToString() + string.substring(index)
        return this
    }

    /**
     * Inserts characters in the specified character sequence
     * @param value the character sequence from which characters are inserted. If
     * @param index the position
     * * Returns the index within this string builder starting at the specified [startIndex].
     * @param string the string representation of the specified boolean [value] into this string builder at the specified [index].
     * * Inserts the specified character array [value] into this string builder at the specified [index].
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
     * * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
of this string builder, or `startIndex > endIndex`.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder {\n    checkReplaceRange(startIndex, endIndex, length)\n    this.string = this.string.substring(0, startIndex) + value + this.string.substring(endIndex)\n    return this\n}\nprivate fun checkReplaceRange(startIndex: Int, endIndex: Int, length: Int) {\n    if (startIndex < 0 || startIndex > length) {\n        throw IndexOutOfBoundsException("startIndex: $startIndex, length: $length")\n    }\n    if (startIndex > endIndex) {\n        throw IllegalArgumentException("startIndex($startIndex) > endIndex($endIndex)")\n    }\n}\n@WasExperimental(ExperimentalStdlibApi::class)\n@SinceKotlin("1.4")\npublic fun setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder {\n    checkReplaceRange(startIndex, endIndex, length)\n    this.string = this.string.substring(0, startIndex) + value + this.string.substring(endIndex)\n    return this\n}\nprivate fun checkReplaceRange(startIndex: Int, endIndex: Int, length: Int) {\n    if (startIndex < 0 || startIndex > length) {\n        throw IndexOutOfBoundsException("startIndex: $startIndex, length: $length")\n    }\n    if (startIndex > endIndex) {\n        throw IllegalArgumentException("startIndex($startIndex) > endIndex($endIndex)")\n    }\n}\n@WasExperimental(ExperimentalStdlibApi::class)\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun deleteAt(index: Int): StringBuilder {\n    AbstractList.checkElementIndex(index, length)\n    string = string.substring(0, index) + string.substring(index + 1)\n    return this\n}\n/**\n * Removes the character at the specified [index] from this string builder and returns this instance.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun deleteAt(index: Int): StringBuilder {\n    AbstractList.checkElementIndex(index, length)\n    string = string.substring(0, index) + string.substring(index + 1)\n    return this\n}\n/**\n * Removes characters in the specified range from this string builder and returns this instance.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun deleteRange(startIndex: Int, endIndex: Int): StringBuilder {\n    checkReplaceRange(startIndex, endIndex, length)\n    string = string.substring(0, startIndex) + string.substring(endIndex)\n    return this\n}\n/**\n * Copies characters from this string builder into the [destination] character array.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) {\n    AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n    AbstractList.checkBoundsIndexes(destinationOffset, destinationOffset + endIndex - startIndex, destination.size)\n    var dstIndex = destinationOffset\n    for (index in startIndex until endIndex) {\n        destination[dstIndex++] = string[index]\n    }\n}\n/**\n * Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {\n    @OptIn(ExperimentalStdlibApi::class)\n    string += value.concatToString(startIndex, endIndex)\n    return this\n}\n/**\n * Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance.\n */\n@SinceKotlin("1.4")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder {\n    @OptIn(ExperimentalStdlibApi::class)\n    string += value.substring(startIndex, endIndex)\n    return this\n}\n
Int, endIndex: Int): StringBuilder {
    val stringCsq = value.toString()
    AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)
    return this
}

/**
 * Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance. 
 * The inserted characters go in same order as in the [value] array, starting at [index]. 
 * @param startIndex the beginning (inclusive) of the subarray to insert. 
 * @param endIndex the end (exclusive) of the subarray to insert. 
 * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`. 
 * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {
    AbstractList.checkPositionIndex(index, this.length)
    @OptIn(ExperimentalStdlibApi::class)
    string = string.substring(0, index) + value.concatToString(startIndex, endIndex) + string.substring(index)
    return this
}

/**
 * Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance. 
 * The inserted characters go in the same order as in the [value] character sequence, starting at [index]. 
 * @param startIndex the beginning (inclusive) of the subsequence to insert. 
 * @param endIndex the end (exclusive) of the subsequence to insert. 
 * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`. 
 * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder {
    AbstractList.checkPositionIndex(index, length)
    val stringCsq = value.toString()
    AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)
    string = string.substring(0, index) + stringCsq.substring(startIndex, endIndex) + string.substring(index)
    return this
}

/**
 * Clears the content of this string builder making it empty and returns this instance. 
 *
 * @sample samples.text.Strings.clearStringBuilder
 */
@SinceKotlin("1.3")
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.clear(): StringBuilder = this.clear()

/**
 * Sets the character at the specified [index] to the specified [value]. 
 * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline operator fun StringBuilder.set(index: Int, value: Char) = this.set(index, value)

/**
 * Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance. 
 * @param startIndex the beginning (inclusive) of the range to replace. 
 * @param endIndex the end (exclusive) of the range to replace. 
 * @param value the string to replace with. 
 * @throws IndexOutOfBoundsException or [IllegalArgumentOutOfRangeException] if [index] is out of range of this string builder.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder =
    this.setRange(startIndex, endIndex, value)

/**
 * Removes the character at the specified [index] from this string builder and returns this instance. 
 * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character. 
 * @param index the index of `Char` to remove. 
 * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun StringBuilder.remove(index: Int): StringBuilder =
    this.remove(index)
@Suppress("EXTENSION_SHELDED", "NOTHING_TO_INLINE")

public actual inline fun StringBuilder.deleteAt(index: Int): StringBuilder = this.deleteAt(index)

/**
 * Removes characters in the specified range from this string builder and returns this instance.
 * @param startIndex the beginning (inclusive) of the range to remove.
 * @param endIndex the end (exclusive) of the range to remove.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.
 */

public actual inline fun StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder = this.deleteRange(startIndex, endIndex)

/**
 * Copies characters from this string builder into the [destination] character array.
 * @param destination the array to copy to.
 * @param destinationOffset the position in the array to copy to, 0 by default.
 * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.
 * @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices.
 */

public actual inline fun StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) =
    this.toCharArray(destination, destinationOffset, startIndex, endIndex)

/**
 * Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance.
 * @param value the array from which characters are appended.
 * @param startIndex the beginning (inclusive) of the subarray to append.
 * @param endIndex the end (exclusive) of the subarray to append.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices.
 */

public actual inline fun StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =
    this.appendRange(value, startIndex, endIndex)

/**
 * Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance.
 * @param value the character sequence from which a subsequence is appended.
 * @param startIndex the beginning (inclusive) of the subsequence to append.
 * @param endIndex the end (exclusive) of the subsequence to append.
 * @throws IndexOutOfBoundsException or [IllegalArgumentSequenceException] when [startIndex] or [endIndex] is out of range of the [value] character sequence.
 */

public actual inline fun StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =
    this.appendRange(value, startIndex, endIndex)

/**
 * Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance.
 * @param index the position in this string builder to insert at.
 * @param value the array from which characters are inserted.
 * @param startIndex the beginning (inclusive) of the subarray to insert.
 * @param endIndex the end (exclusive) of the subarray to insert.
 * @throws IndexOutOfBoundsException or [IllegalArgumentSequenceException] when [startIndex] or [endIndex] is out of range of the [value] array indices.
 */

public actual inline fun StringBuilder.insertRange(value: CharArray, index: Int, startIndex: Int, endIndex: Int): StringBuilder =
    this.insertRange(value, index, startIndex, endIndex)

/**
 * Inserts a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.
 * @param index the position in this string builder to insert at.
 * @param value the character sequence from which a subsequence is inserted.
 * @param startIndex the beginning (inclusive) of the subsequence to insert.
 * @param endIndex the end (exclusive) of the subsequence to insert.
 * @throws IndexOutOfBoundsException or [IllegalArgumentSequenceException] when [startIndex] or [endIndex] is out of range of the [value] character sequence.
 */

public actual inline fun StringBuilder.insertRange(value: CharSequence, index: Int, startIndex: Int, endIndex: Int): StringBuilder =
    this.insertRange(value, index, startIndex, endIndex)

/**
 * Removes characters in the specified range from this string builder and returns this instance.
 * @param startIndex the beginning (inclusive) of the range to remove.
 * @param endIndex the end (exclusive) of the range to remove.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.
 */

public actual inline fun StringBuilder.deleteAt(index: Int): StringBuilder = this.deleteAt(index)

/**
 * Removes characters in the specified range from this string builder and returns this instance.
 * @param startIndex the beginning (inclusive) of the range to remove.
 * @param endIndex the end (exclusive) of the range to remove.
 * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.
 */
StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =
this.insertRange(index, value, startIndex, endIndex)

/**
 * Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.
 * @param index the position in this string builder to insert at.
 * @param value the character sequence from which a subsequence is inserted.
 * @param startIndex the beginning (inclusive) of the subsequence to insert.
 * @param endIndex the ending (exclusive) of the subsequence to insert.
 * @return this instance.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@Suppress("EXTENSION_SHADOWED_BY_MEMBER", "NOTHING_TO_INLINE")
public actual inline fun
StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =
this.insertRange(index, value, startIndex, endIndex)

/**
 * Returns `true` if this character is a Unicode high-surrogate code unit (also known as leading-surrogate code unit).
 */
public actual fun Char.isHighSurrogate(): Boolean = this in
Char.MIN_HIGH_SURROGATE..Char.MAX_HIGH_SURROGATE

/**
 * Returns `true` if this character is a Unicode low-surrogate code unit (also known as trailing-surrogate code unit).
 */
public actual fun Char.isLowSurrogate(): Boolean = this in
Char.MIN_LOW_SURROGATE..Char.MAX_LOW_SURROGATE

/**
 * Parses the string as a signed [Byte] number and returns the result.
 * @return the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toByte(): Byte = toByteOrNull() ?: numberFormatError(this)

/**
 * Parses a string as a signed [Byte] number.
 * @param radix the radix for string to number conversion.
 * @return the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toByte(radix: Int): Byte = toByteOrNull(radix) ?: numberFormatError(this)

/**
 * Parses the string as a signed [Short] number.
 * @param radix the radix for string to number conversion.
 * @return the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
public actual fun String.toShort(): Short = toShortOrNull() ?: numberFormatError(this)
number

* @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.

*An

public actual fun String.toDouble(): Double = (+(this.asDynamic())).unsafeCast<Double>().also {
    if (it.isNaN() && !this.isNaN() || it == 0.0 && !this.isBlank())
        numberFormatError(this)
}

/**
 * Parses the string as a [Double] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a [Double] number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 * @param radix the radix to use for conversion.
 */
@kotlin.internal.InlineOnly
public actual inline fun String.toDoubleOrNull(): Double? = (+(this.asDynamic())).unsafeCast<Double>().takeIf {
    !(it.isNaN() && !this.isNaN() || it == 0.0 &&
        this.isBlank())
}

/**
 * Parses the string as a [Double] number and returns the result.
 * @param radix the radix to use for conversion.
 * @throws NumberFormatException if the string is not a valid representation of a [Double] number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
public actual fun Double.doubleToLong(): Long = doubleToLongOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a [Long] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a [Long] number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@kotlin.internal.InlineOnly
public actual inline fun Double.toDoubleLong(): Long = toDoubleOrNull().unsafeCast<Double>().let {
    if (it.isNaN() && !this.isNaN() || it == 0.0 && !this.isBlank())
        numberFormatError(this)
}

/**
 * Parses the string as a [Float] number and returns the result.
 * @param radix the radix to use for conversion.
 * @throws NumberFormatException if the string is not a valid representation of a [Float] number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@kotlin.internal.InlineOnly
public actual inline fun Float.toFloat(): Float = toDouble().unsafeCast<Float>()

/**
 * Parses the string as a [Float] number and returns the result.
 * @param radix the radix to use for conversion.
 * @throws NumberFormatException if the string is not a valid representation of a [Float] number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
public actual fun Float.toFloatOrNull(): Float? = toDoubleOrNull().unsafeCast<Float?>()

/**
 * Parses the string as a [Byte] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual inline fun Byte.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Parses the string as a [Short] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.2")
@kotlin.internal.InlineOnly
public actual inline fun Short.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Parses the string as an [Int] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a [Int] number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
public actual fun Int.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))

private fun String.isNaN(): Boolean = when (this.toLowerCase()) {
    "nan", "+nan", "-nan" -> true
    else -> false
}

/**
 * Checks whether the given [radix] is valid radix for
 * string to number conversion.
 * @param radix the radix to use for conversion.
 * @throws NumberFormatException if the string is not a valid representation of a [Number] number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@PublishedApi
internal actual fun checkRadix(radix: Int): Int {
    if (radix !in 2..36) {
        throw IllegalArgumentException("radix $radix was not in valid range 2..36")
    }
    return radix
}

internal actual fun digitOf(char: Char, radix: Int): Int = when {
    char >= '0' && char <= '9' -> char - '0'
    char >= 'A' && char <= 'Z' -> char - 'A' + 10
    char >= 'a' && char <= 'z' -> char - 'a' + 10
    else -> -1
}.let { if (it >= radix) -1 else it }

/**
 * Represents the results from a single capturing group within a [MatchResult] of [Regex].
 */
public actual data class MatchGroup(actual val value: String)

/**
 * Represents a compiled
 *RegExp
 */
public actual data class RegExp

/**
 * Used of this source code is governed by the Apache 2.0 license that can be found in the
 *license/LICENSE.txt file.
 */

package kotlin.text

import kotlin.js.RegExp

/**
 * Provides
 *enumeration values to use to set regular expression options.
 */
public actual enum class RegexOption(val value: String) {
    /** Enables case-insensitive matching. */
    IGNORE_CASE("i"),
    /** Enables multiline
 *mode. */
    MULTILINE("m")
}


/**
 * Represents the results from a single capturing group within a [MatchResult] of [RegExp].
 */
public actual data class MatchGroup(actual val value: String) *
regular expression. * Provides functions to match strings in text with a pattern, replace the found occurrences and split text around matches. * * For pattern syntax reference see [MDN RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp#Special_characters_meaning_in_regular_expressions) */

```
public actual class Regex {
    public actual constructor(pattern: String, options: Set<RegexOption>) {
        // Regex constructor logic...
    }

    public actual constructor(pattern: String, option: RegexOption) {
        // Regex constructor logic...
    }

    public actual constructor(pattern: String) {
        // Regex constructor logic...
    }

    public actual val pattern: String
    public actual val options: Set<RegexOption>
    private val nativePattern: RegExp = RegExp(pattern, options.map { it.value }.joinToString(separator = "") + "g")

    public actual infix fun matches(input: CharSequence): Boolean {
        nativePattern.reset()
        val match = nativePattern.exec(input.toString())
        return match != null && match.index == 0 && nativePattern.lastIndex == input.length
    }

    public actual fun containsMatchIn(input: CharSequence): Boolean {
        nativePattern.reset()
        return nativePattern.test(input.toString())
    }

    public actual fun find(input: CharSequence, startIndex: Int = 0): MatchResult? {
        // Regex find logic...
    }

    public actual fun findAll(input: CharSequence, startIndex: Int = 0): Sequence<MatchResult> {
        // Regex findAll logic...
    }

    public actual fun matchEntire(input: CharSequence): MatchResult? {
        // Regex matchEntire logic...
    }

    public actual fun replace(input: CharSequence, replacement: String): String {
        // Regex replace logic...
    }

    public actual inline fun replace(input: CharSequence, transform: (MatchResult) -> CharSequence): String {
        // Regex replace logic...
    }
```

[36x36]Open Source Used In DNAC Platform Antelope  14073
do {
            val foundMatch = match!!
            sb.append(input, lastStart, foundMatch.range.start)
            sb.append(transform(foundMatch))
            lastStart = foundMatch.range.endInclusive + 1
            match = foundMatch.next()}
    while (lastStart < length && match != null) {
        if (lastStart < length) {
            sb.append(input, lastStart, length)
        }
        return sb.toString()}
    }

    /**
     * Replaces the first occurrence of this regular expression in the specified [input] string with specified [replacement] expression.
     */
    public actual fun replaceFirst(input: CharSequence, replacement: String): String {
        val nonGlobalOptions = options.map { it.value }.joinToString(separator = "")
        return input.toString().nativeReplace(RegExp(pattern, nonGlobalOptions), replacement) }

    /**
     * Splits the [input] CharSequence around matches of this regular expression.
     */
    public actual fun split(input: CharSequence, limit: Int = 0): List<String> {
        require(limit >= 0) { "Limit must be non-negative, but was $limit" }
        val matches = findAll(input).let { if (limit == 0) it else it.take(limit - 1) }
        val result = mutableListOf<String>()
        var lastStart = 0
        for (match in matches) {
            result.add(input.subSequence(lastStart, match.range.start).toString())
            lastStart = match.range.endInclusive + 1
        }
        result.add(input.subSequence(lastStart, input.length).toString())
        return result
    }

    /**
     * Returns the string representation of this regular expression, namely the [pattern] of this regular expression.
     */
    public override fun toString(): String = nativePattern.toString()

    actual companion object {
        /**
         * Returns a regular expression that matches the specified [literal] string literally.
         * No characters of that string will have special meaning when searching for an occurrence of the regular expression.
         */
        public actual fun fromLiteral(literal: String): Regex = Regex(escape(literal))

        /**
         * Returns a regular expression pattern string that matches the specified [literal] string literally.
         * No characters of that string will have special meaning when searching for an occurrence of the regular expression.
         */
        public actual fun escape(literal: String): String = literal.nativeReplace(patternEscape, "\\$&")

        /**
         * Returns a literal replacement expression for the specified [literal] string.
         * No characters of that string will have special meaning when it is used as a replacement string in [Regex.replace] function.
         */
        public actual fun escapeReplacement(literal: String): String = literal.nativeReplace(replacementEscape, "$$$$")

        private val patternEscape = RegExp(\"\[-\\^$*+?.()|\[\]{}\]\", "g")
        private val replacementEscape = RegExp("\"", "g")
    }
}

/**
Creates a regular expression from the specified [pattern] string and the default options.
*/
@Deprecated("Provided for binary compatibility", level = DeprecationLevel.HIDDEN)
@JsName("Regex_sb3q2\$")
public fun Regex_0(pattern: String, option: RegexOption): Regex = Regex(pattern, setOf(option))

/** Creates a regular expression from the specified
[pattern] string and the specified single [option].
*/
@Deprecated("Provided for binary compatibility", level = DeprecationLevel.HIDDEN)
@JsName("Regex_61zpoe\$")
public fun Regex_1(pattern: String): Regex = Regex(pattern, emptySet())
val size: Int get() = match.length
override fun get(index: Int): String = match[index] ?: ""

    }
    }
    return groupValues_!!
  }

  override fun next(): MatchResult? =
  this@findNext.findNext(input, if (range.isEmpty()) range.start + 1 else range.endInclusive + 1)
}
}

import kotlin.js.RegExp

@kotlin.internal.InlineOnly
internal actual inline fun String.nativeIndexOf(ch: Char, fromIndex: Int): Int =
nativeIndexOf(ch.toString(), fromIndex)

@kotlin.internal.InlineOnly
internal actual inline fun String.nativeLastIndexOf(ch: Char, fromIndex: Int): Int =
nativeLastIndexOf(ch.toString(), fromIndex)

/**
 * Returns `true` if a substring of this string starting at the specified offset [startIndex] starts with the specified prefix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.startsWith(prefix: String, startIndex: Int, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeStartsWith(prefix, startIndex)
    else
        return regionMatches(startIndex, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if this string starts with the specified prefix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.startsWith(prefix: String, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeStartsWith(prefix, 0)
    else
        return regionMatches(0, prefix, 0, prefix.length, ignoreCase)
}

/**
 * Returns `true` if this string ends with the specified suffix.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.endsWith(suffix: String, ignoreCase: Boolean = false): Boolean {
    if (!ignoreCase)
        return nativeEndsWith(suffix)
    else
        return regionMatches(length - suffix.length, suffix, 0, suffix.length, ignoreCase)
}

/**
 * Returns a copy of this string having its first letter titlecased using the rules of the default locale.
 * or the original string if it's empty or already starts with a title case letter.
 * The title case of a character is usually the same as its upper case with several exceptions.
 * The particular list of characters with the special title case form depends on the underlying platform.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.capitalize(): String {
    return if (isNotEmpty()) substring(0, 1).toUpperCase() + substring(1) else this
}

/**
 * Returns a copy of this string having its first letter lowercased using the rules of the default locale.
 * or the original string if it's empty or already starts with a lower case letter.
 */
@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.decapitalize(): String {
    return if (isNotEmpty()) substring(0, 1).toLowerCase() + substring(1) else this
}

/**
 * Returns a string containing this char sequence repeated [n] times.
 * @throws IllegalArgumentException when n < 0.
 */
public actual fun CharSequence.repeat(n: Int): String {
    require(n >= 0) { "Count 'n' must be non-negative, but was $n." }
    return when (n) {
        0 -> ""
        1 -> this.toString()
        else -> {
            var result = ""
            if (!isEmpty()) {
                var s = this.toString()
                var count = n
                while (true) {
                    if ((count and 1) == 1) { result += s
                        count = count ushr 1}
                    count = count + 1
                    s += s
                }
            }
            return result
        }
    }
}

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(Regex.escape(oldValue), if (ignoreCase) "gi" else "g"), newValue, newValue)}
regex.escapeReplacement(newValue))

public actual fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(regex.escape(oldChar.toString())), if (ignoreCase) "g\i" else "g\i"),
newChar.toString())

public actual fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(regex.escape(oldChar.toString())), if (ignoreCase) "i" else ""),
newChar.toString())

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(regex.escape(oldChar.toString()), if (ignoreCase) "i" else ""),
newChar.toString())

@Suppress("ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS")
public actual fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =
nativeReplace(RegExp(regex.escape(oldValue), if (ignoreCase) "i" else ""),
regex.escapeReplacement(newValue))

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in
 * the LICENSE file.
 */

package kotlin.text

/** Returns the negative [size] if [throwOnMalformed] is false, throws [CharacterCodingException] otherwise. */
@OptIn(ExperimentalStdlibApi::class) private fun unprivate fun
malformed(size: Int, index: Int, throwOnMalformed: Boolean): Int {
    if (throwOnMalformed) throw CharacterCodingException("Malformed sequence starting at \$\{index - 1\}")

    return -size
}

/**
 * Returns code point corresponding to UTF-16 surrogate pair, where the first of the pair is the [high] and the second is in
 * the [string] at the [index].
 * Returns zero if the pair is malformed and [throwOnMalformed] is false.
 * @throws CharacterCodingException if the pair is malformed and [throwOnMalformed] is true.
 * @private fun codePointFromSurrogate(string: String, high: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (high !in 0xD800..0xDBFF || index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }

    val low = string[index].toInt()
    if (low !in 0xDC00..0xDFFF) {
        return malformed(0, index, throwOnMalformed)
    }
    return 0x10000 + ((high and 0x3FF) shl 10) or (low and 0x3FF)
}

/**
 * Returns code point corresponding to UTF-8 sequence of two bytes, where the first byte of the sequence is the [byte1] and the second byte is in the [bytes] array at the [index].
 * Returns zero if the sequence is malformed and [throwOnMalformed] is false.
 * @throws CharacterCodingException if the sequence of two bytes is malformed and [throwOnMalformed] is true.
 * @private fun codePointFrom2(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (byte1 and 0x1E == 0 || index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }

    val byte2 = bytes[index].toInt()
    if (byte2 and 0xC0 != 0x80) {
        return malformed(0, index, throwOnMalformed)
    }

    return (byte1 shl 6) xor byte2 xor 0xF80
}

/**
 * Returns code point corresponding to UTF-8 sequence of three bytes, where the first byte of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].
 * Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence if the sequence is malformed and [throwOnMalformed] is false.
 * @throws CharacterCodingException if the sequence of three bytes is malformed and [throwOnMalformed] is true.
 * @private fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }

    val byte2 = bytes[index].toInt()
    if (byte1 and 0xF == 0) {
        if (byte2 and 0xE0 != 0x80) {
            return malformed(0, index, throwOnMalformed)
        }

        if (byte2 and 0x80 != 0) {
            return malformed(0, index, throwOnMalformed)
        }
    }

    if (byte2 and 0xE0 == 0xA0) {
        return malformed(0, index, throwOnMalformed)
    }

    else if (byte1 and 0x80 == 0) {
        return malformed(0, index, throwOnMalformed)
    }

    else if (byte2 and 0x80 != 0) {
        return malformed(0, index, throwOnMalformed)
    }

    else if (byte2 and 0x20 != 0) {
        return malformed(0, index, throwOnMalformed)
    }

    return (byte1 shl 12) xor byte2 xor byte3 xor 0x80
}

/**
 * Returns code point corresponding to UTF-8 sequence of four bytes, where the first byte of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].
 * Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence if the sequence is malformed and [throwOnMalformed] is false.
 * @throws CharacterCodingException if the sequence of four bytes is malformed and [throwOnMalformed] is true.
 * @private fun codePointFrom4(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {
    if (index >= endIndex) {
        return malformed(0, index, throwOnMalformed)
    }

    val byte2 = bytes[index].toInt()
    if (byte1 and 0xF == 0) {
        if (byte2 and 0xE0 != 0x80) {
            return malformed(0, index, throwOnMalformed)
        }

        if (byte2 and 0x80 != 0) {
            return malformed(0, index, throwOnMalformed)
        }
    }

    if (byte2 and 0xE0 == 0xA0) {
        return malformed(0, index, throwOnMalformed)
    }

    else if (byte1 and 0x80 == 0) {
        return malformed(0, index, throwOnMalformed)
    }

    else if (byte2 and 0x80 != 0) {
        return malformed(0, index, throwOnMalformed)
    }

    else if (byte2 and 0x20 != 0) {
        return malformed(0, index, throwOnMalformed)
    }

    else if (byte2 and 0x20 != 0) {
        return malformed(0, index, throwOnMalformed)
    }

    return (byte1 shl 12) xor byte2 xor byte3 xor byte4 xor 0x80
}
malformed(0, index, throwOnMalformed)
    } else if (byte1 and 0xF == 0x0) {
        if (byte2 and 0xF0 <= 0x80) {
            // Non-shortest form
            return malformed(0, index, throwOnMalformed)
        } else if (byte1 and 0xF == 0x4) {
            if (byte2 and 0xF0 != 0x80) {
                // Out of Unicode code points domain (larger than U+10FFFF)
                return malformed(0, index, throwOnMalformed)
            } else if (byte2 and 0xC0 != 0x80) {
                // Surrogate code points in '0xD800..0xDFFF'
                return malformed(1, index, throwOnMalformed)
            } else if (byte2 and 0xC0 == 0x80) {
                val byte3 = bytes[index + 1].toInt()
                if (byte3 and 0xC0 == 0x80) {
                    // Single surrogate
                    return malformed(1, index, throwOnMalformed)
                } else if (byte3 and 0xC0 == 0x80) {
                    // Double surrogate
                    return malformed(2, index, throwOnMalformed)
                } else if (byte3 and 0xC0 == 0x80) {
                    // Three surrogate
                    return malformed(3, index, throwOnMalformed)
                }
            }
        }
    }
    return (byte1 shl 18) xor (byte2 shl 12) xor (byte3 shl 6) xor byte4 xor 0x381F80
}

/**
 * Maximum number of bytes needed to encode a single char.
 *
 * Code points in `0..0x7F` are encoded in a single byte.
 * Code points in `0x80..0x7FF` are encoded in two bytes.
 * Code points in `0x800..0xD7FF` or in `0xE000..0xFFFF` are encoded in three bytes.
 * Surrogate code points in `0xD800..0xDFFF` are not Unicode scalar values, therefore aren't encoded.
 */
private const val MAX_BYTES_PER_CHAR = 3

/**
 * The byte sequence a malformed UTF-16 char sequence is replaced by.
 */
private val REPLACEMENT_BYTE_SEQUENCE: ByteArray = byteArrayOf(0xEF.toByte(), 0xBF.toByte(), 0xBD.toByte())

/**
 * Encodes the [string] using UTF-8 and returns the resulting [ByteArray].
 *
 * @param string the string to encode.
 * @param startIndex the start offset (inclusive) of the substring to encode.
 * @param endIndex the end offset (exclusive) of the substring to encode.
 * @param throwOnMalformed whether to throw on malformed char sequence or replace by the [REPLACEMENT_BYTE_SEQUENCE].
 *
 * @throws CharacterCodingException if the char sequence is malformed and [throwOnMalformed] is true.
 */
internal fun encodeUtf8(string: String, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): ByteArray {
    require(startIndex >= 0 && endIndex <= string.length && startIndex <= endIndex)

    val bytes = ByteArray((endIndex - startIndex) * MAX_BYTES_PER_CHAR)
    var byteIndex = 0
    var charIndex = startIndex

    while (charIndex < endIndex) {
        val code = string[charIndex++].toInt()
        when {
            code < 0x80 ->
                bytes[byteIndex++] = code.toByte()
            code < 0x800 -> {
                bytes[byteIndex++] = ((code shr 6) or 0xC0).toByte()
                bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()
            }
            code < 0xD800 || code >= 0xE000 -> {
                bytes[byteIndex++] = ((code shr 12) or 0xE0).toByte()
                bytes[byteIndex++] = (((code shr 6) and 0x3F) or 0x80).toByte()
            }
            else -> { // Surrogate char value
                val codePoint = codePointFromSurrogate(string, code, charIndex, endIndex, throwOnMalformed)
                if (codePoint <= 0) {
                    bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[0]
                    bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[1]
                    bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[2]
                } else {
                    bytes[byteIndex++] = ((codePoint shr 18) or 0x0F0).toByte()
                    bytes[byteIndex++] = (((codePoint shr 12) and 0x3F) or 0x80).toByte()
                }
            }
        }
    }

    return if (bytes.size == byteIndex) bytes else bytes.copyOf(byteIndex)
}

/**
 * The character a malformed UTF-8 byte sequence is replaced by.
 */
private const val REPLACEMENT_CHAR = '\uFFFD'

/**
 * Decodes the UTF-8 [bytes] array and returns the resulting [String].
 *
 * @param bytes the byte array to decode.
 * @param startIndex the start offset (inclusive) of the array to be decoded.
 * @param endIndex the end offset (exclusive) of the array to be decoded.
 * @param throwOnMalformed whether to throw on malformed byte sequence or replace by the [REPLACEMENT_CHAR].
 *
 * @throws CharacterCodingException if the array is malformed UTF-8 byte sequence and [throwOnMalformed] is true.
 */
internal fun decodeUtf8(bytes: ByteArray, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): String {
    require(startIndex >= 0 && endIndex <= bytes.size - 1)
messageLines = if (stackStart == 0) 0 else 1 + shortInfo.count { c -> c == '\n' }
stack.lineSequence().forEachIndexed { index: Int, line: String ->
                    if (index >= messageLines)
                        target.append(indent)
                        target.append(line).append("\n")
                }
            } else {
                target.append(stack).append("\n")
            }
        } else {
            target.append(shortInfo).append("\n")
        }

        val suppressed = suppressedExceptions
        if (suppressed.isNotEmpty()) {
            val suppressedIndent = indent + "    
            for (s in suppressed) {
                s.dumpFullTrace(suppressedIndent, "Suppressed: ")
            }
        }
        return true
    }

    private fun dropCommonFrames(stack: String, stackStart: Int): String {
        var commonFrames: Int = 0
        var lastBreak: Int = 0
        var preLastBreak: Int = 0
        for (pos in 0 until minOf(topStack.length - topStackStart, stack.length - stackStart)) {
            val c = stack[stack.lastIndex - pos]
            if (c != topStack[topStack.lastIndex - pos]) break
            if (c == '\n') {
                commonFrames += 1
                preLastBreak = lastBreak
                lastBreak = pos
            }
        }
        if (commonFrames <= 1) return stack
        while (preLastBreak > 0 && stack[stack.lastIndex - (preLastBreak - 1)] == ' ')
            preLastBreak -= 1
        return stack.dropLast(preLastBreak - 1) + "... and ${commonFrames - 1} more common stack frames skipped"
    }
}
fun read(): Double = kotlin.js.Date.now() 
 override fun toString(): String = "TimeSource(Date.now())": */
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * @suppress("DEPRECATION", "DEPRECATION_ERROR")
 kotlin.time

 fun read(): Double = kotlin.js.Date.now() 
 override fun toString(): String = "TimeSource(Date.now()): */

 // This function is used to format the time as a Date object.
 fun formatToExactDecimals(value: Double, decimals: Int): String {
     val rounded = if (decimals == 0) {
         value
     } else {
         val pow = 10.0.pow(decimals)
         kotlin.js.Math.round(abs(value) * pow) / pow * sign(value)
     }
     return rounded.asDynamic().toFixed(decimals).unsafeCast<String>()
 }

 fun formatUpToDecimals(value: Double, decimals: Int): String {
     return value.asDynamic().toLocaleString("en-us", json("maximumFractionDigits" to decimals)).unsafeCast<String>()
 }

 fun formatScientific(value: Double): String {
     return value.asDynamic().toExponential(2).unsafeCast<String>()
 }

 /*
 * Copyright 2020-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

 package kotlinx.dom

 import org.w3c.dom.*
 import kotlin.contracts.*

 /**
 * Creates a new element with the specified [name].
 * The element is initialized with the specified [init] function.
 * The element is initialized with the specified [init] function.
 */
 @SinceKotlin("1.4")
 fun Document.createElement(name: String, init: Element.() -> Unit): Element {
     contract { callsInPlace(init, InvocationKind.EXACTLY_ONCE) }
     return createElement(name).apply(init)
 }

 /**
 * Appends a newly created element with the specified [name] to this element.
 * The element is initialized with the specified [init] function.
 */
 @SinceKotlin("1.4")
 fun Element.appendElement(name: String, init: Element.() -> Unit): Element {
     contract { callsInPlace(init, InvocationKind.EXACTLY_ONCE) }
     return ownerDocument!!.createElement(name, init).also {
         appendChild(it)
     }.unsafeCast<String>()
 }

 /*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

 package kotlin.text

 /**
 * Converts the string into a regular expression [Regex] with the default options.
 */
 @kotlin.internal.InlineOnly
 public inline fun String.toRegex(): Regex = Regex(this)

 /**
 * Converts the string into a regular expression [Regex] with the specified single [option].
 */
 @kotlin.internal.InlineOnly
 public inline fun String.toRegex(option: RegexOption): Regex = Regex(this, option)

 /**
 * Converts the string into a regular expression [Regex] with the specified set of [options].
 */
 @kotlin.internal.InlineOnly
 public inline fun String.toRegex(options: Set<RegexOption>): Regex = Regex(this, options)

 /*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */
Open Source Used In DNAC Platform Antelope 14081
of range of the [value] character sequence indices or when `startIndex > endIndex`.

```kotlin
*\@SinceKotlin("1.4")*\n@Experimental(ExperimentalStdlibApi::class)*\npublic expect fun StringBuilder.appendRange(value: CharSequence, start: Int, end: Int): StringBuilder
```

Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance.

```kotlin
*\n* @param index the position in this string builder to insert at.\n* @param value the array from which characters are inserted.\n* @param start the beginning (inclusive) of the subarray to insert.\n* @param end the end (exclusive) of the subarray to insert.\n* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [start] or [end] is out of range of the [value] array indices or when `start > end`.\n* @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n```

```kotlin
*\@SinceKotlin("1.4")*\n@Experimental(ExperimentalStdlibApi::class)*\npublic expect fun StringBuilder.insertRange(index: Int, value: CharArray, start: Int, end: Int): StringBuilder
```

Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.

```kotlin
*\n* @param index the position in this string builder to insert at.\n* @param value the character sequence from which a subsequence is inserted.\n* @param start the beginning (inclusive) of the subsequence to insert.\n* @param end the end (exclusive) of the subsequence to insert.\n* @throws IndexOutOfBoundsException or [IllegalArgumentException] when [start] or [end] is out of range of the [value] character sequence indices or when `start > end`.\n```

```kotlin
@Suppress("EXTENSION_SHADOWED_BY_MEMBER")
@Deprecated("Use append(value: Any?) instead", ReplaceWith("append(value = obj)"); DeprecationLevel.WARNING)
@kotlin.internal.InlineOnly
public inline fun StringBuilder.append(obj: Any?): StringBuilder = this.append(obj)
```

```kotlin
public fun StringBuilder.append(vararg value: String?): StringBuilder {
    for (item in value)
        append(item)
    return this
}
```

```kotlin
public fun StringBuilder.append(vararg value: Any?): StringBuilder {
    for (item in value)
        append(item)
    return this
}
```

```kotlin
/**
 * Appends a line feed character (`\n`) to this StringBuilder.
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun StringBuilder.appendLine(): StringBuilder = append('\n')
```

```kotlin
/** Appends [value] to this StringBuilder, followed by a line feed character (`\n`). */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun StringBuilder.appendLine(value: CharSequence?): StringBuilder = append(value).appendLine()\n```

```kotlin
/** Appends [value] to this StringBuilder, followed by a line feed character (`\n`). */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun StringBuilder.appendLine(value: String?): StringBuilder = append(value).appendLine()\n```

```kotlin
/** Appends [value] to this StringBuilder, followed by a line feed character (`\n`). */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun StringBuilder.appendLine(value: Any?): StringBuilder = append(value).appendLine()\n```
character ('\n'). */
@SinceKotlin("1.4")
public fun
StringBuilder.appendLine(value: CharArray): String =
StringBuilder = append(value).appendLine() /* This line will not be executed. */

/**
 * Appends [value] to this [StringBuilder],
 * followed by a line feed character ('\n').
 */
@SinceKotlin("1.4")
public fun
StringBuilder.appendLine(value: Char): String =
StringBuilder = append(value).appendLine() /* This line will not be executed. */

/**
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file. */

@kotlin.internal.InlineOnly
public inline fun
StringBuilder.appendLine(value: Boolean): StringBuilder =
StringBuilder = append(value).appendLine() /* This line will not be executed. */

/** Appends 
 * @SinceKotlin("1.4")
 */
@SinceKotlin("1.4")
@kotlin.internal.InlineOnly
public inline fun
StringBuilder.appendLine(): StringBuilder =
StringBuilder = append().appendLine() /* This line will not be executed. */

/**
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file. */

package kotlinx.dom

import org.w3c.dom.*

/**
 * Gets a value indicating whether this node is a TEXT_NODE or a
 * CDATA_SECTION_NODE. */
@SinceKotlin("1.4")
public val Node.isText: Boolean =
nodeType == Node.TEXT_NODE || nodeType == Node.CDATA_SECTION_NODE

/**
 * Gets a value indicating whether this node is an [Element].
 * @SinceKotlin("1.4")
 */
@SinceKotlin("1.4")
public val Node.isElement: Boolean =
nodeType == Node.ELEMENT_NODE

/**
 * Removes all the children from this node.
 * @SinceKotlin("1.4")
 */
@SinceKotlin("1.4")
fun Node.clear() {
    while (hasChildNodes()) {
        removeChild(firstChild!!)
    }
}

/**
 * Creates text node and append it to the element.
 * @SinceKotlin("1.4")
 */
@SinceKotlin("1.4")
fun Element.appendText(text: String): Element {
    appendChild(ownerDocument!!.createTextNode(text))
    return this
}

/**
 * Use of this source code is governed by the Apache 2.0 license that can be found in the
 * license/LICENSE.txt file. */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See
// github.com/kotlin/dukat for details

package org.khronos.webgl

import kotlin.js.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.workers.*
import org.w3c.xhr.*

public external interface WebGLContextAttributes {

    var alpha: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var depth: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var stencil: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var antialias: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var premultipliedAlpha: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var preserveDrawingBuffer: Boolean? /* = true */
        get() = definedExternally
        set(value) = definedExternally
    var preferLowPowerToHighPerformance: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var failIfMajorPerformanceCaveat: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally

}
WebGLContextAttributes {\n   val o = js("{{ }}")\n   o["alpha"] = alpha\n   o["depth"] = depth\n   o["stencil"] = stencil\n   o["antialias"] = antialias\n   o["premultipliedAlpha"] = premultipliedAlpha\n   o["preserveDrawingBuffer"] = preserveDrawingBuffer\n   o["preferLowPowerToHighPerformance"] = preferLowPowerToHighPerformance\n   o["failIfMajorPerformanceCaveat"] = failIfMajorPerformanceCaveat\n   return o\n}\npublic external abstract class WebGLObject\n\npublic external abstract class WebGLObject\n\n/**\n * Exposes the JavaScript [ WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin */
public external abstract class WebGLBuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [ WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin */
public external abstract class WebGLFramebuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [ WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin */
public external abstract class WebGLProgram : WebGLObject\n\n/**\n * Exposes the JavaScript [ WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin */
public external abstract class WebGLRenderbuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [ WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin */
public external abstract class WebGLShader : WebGLObject\n\n/**\n * Exposes the JavaScript [ WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin */
public external abstract class WebGLTexture : WebGLObject\n\n/**\n * Exposes the JavaScript [ WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin */
public external abstract class WebGLUniformLocation\n\n/**\n * Exposes the JavaScript [ WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin */
public external abstract class WebGLActiveInfo {\n   open val size: Int\n   open val type: Int\n   open val name: String
}\n\n/**\n * Exposes the JavaScript [ WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to Kotlin */
public external abstract class WebGLShaderPrecisionFormat {\n   open val rangeMin: Int\n   open val rangeMax: Int\n}\n
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
fun cullFace(mode: Int)
    fun deleteBuffer(buffer: WebGLBuffer?)
    fun deleteFramebuffer(framebuffer: WebGLFramebuffer?)
    fun deleteProgram(program: WebGLProgram?)
    fun deleteRenderbuffer(renderbuffer: WebGLRenderbuffer?)
    fun deleteShader(shader: WebGLShader?)
    fun deleteTexture(texture: WebGLTexture?)
    fun depthFunc(func: Int)
    fun depthMask(flag: Boolean)
    fun depthRange(zNear: Float, zFar: Float)
    fun detachShader(program: WebGLProgram?, shader: WebGLShader?)
    fun disable(cap: Int)
    fun enableVertexAttribArray(index: Int)
    fun drawArrays(mode: Int, first: Int, count: Int)
    fun drawElements(mode: Int, count: Int, type: Int, offset: Int)
    fun enable(cap: Int)
    fun enableVertexAttribArray(index: Int)
    fun finish()
    fun flush()
    fun framebufferRenderbuffer(target: Int, attachment: Int, renderbuffertarget: Int, renderbuffer: WebGLRenderbuffer?)
    fun generateMipmap(target: Int)
    fun getActiveAttrib(program: WebGLProgram?, index: Int): WebGLActiveInfo?
    fun getActiveUniform(program: WebGLProgram?, index: Int): WebGLActiveInfo?
    fun getAttachedShaders(program: WebGLProgram?): Array<WebGLShader>?
    fun getAttribLocation(program: WebGLProgram?, name: String): Int?
    fun getBufferParameter(target: Int, pname: Int): Any?
    fun getError(): Int
    fun getFramebufferAttachmentParameter(target: Int, attachment: Int, pname: Int): Any?
    fun getProgramParameter(program: WebGLProgram?, pname: Int): Any?
    fun getProgramInfoLog(program: WebGLProgram?): String?
    fun getRenderbufferParameter(target: Int, pname: Int): Any?
    fun getShaderParameter(shader: WebGLShader?, pname: Int): Any?
    fun getShaderPrecisionFormat(target: Int, precision: Int): WebGLShaderPrecisionFormat?
    fun getShaderSource(shader: WebGLShader?): String?
    fun getTexParameter(target: Int, pname: Int): Any?
    fun getUniformLocation(program: WebGLProgram?, name: String): WebGLUniformLocation?
    fun getVertexAttrib(index: Int, pname: Int): Any?
    fun getVertexAttribOffset(index: Int, pname: Int): Int
    fun hint(target: Int, mode: Int)
    fun isBuffer(buffer: WebGLBuffer?): Boolean
    fun isEnabled(cap: Int): Boolean
    fun isFramebuffer(framebuffer: WebGLFramebuffer?): Boolean
    fun isProgram(program: WebGLProgram?): Boolean
    fun isRenderbuffer(renderbuffer: WebGLRenderbuffer?): Boolean
    fun isShader(shader: WebGLShader?): Boolean
    fun isTexture(texture: WebGLTexture?): Boolean
    fun lineWidth(width: Float)
    fun linkProgram(program: WebGLProgram?)
    fun pixelStorei(pname: Int, param: Int)
    fun polygonOffset(factor: Float, units: Float)
    fun readPixels(x: Int, y: Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?)
    fun renderbufferStorage(target: Int, format: Int, width: Int, height: Int)
    fun sampleCoverage(value: Float, invert: Boolean)
    fun scissor(x: Int, y: Int, width: Int, height: Int)
    fun shaderSource(shader: WebGLShader?, source: String)
    fun stencilFunc(func: Int, ref: Int, mask: Int)
    fun stencilFuncSeparate(face: Int, func: Int, ref: Int, mask: Int)
    fun stencilMask(mask: Int)
    fun stencilMaskSeparate(face: Int, mask: Int)
    fun stencilOp(fail: Int, zfail: Int, zpass: Int)
    fun texImage2D(target: Int, level: Int, internalformat: Int, format: Int, type: Int, pixels: ArrayBufferView?)
    fun texImage2D(target: Int, level: Int, internalformat: Int, format: Int, type: Int, source: TexImageSource?)
    fun texParameterf(target: Int, pname: Int, param: Float)
    fun texParameteri(target: Int, pname: Int, param: Int)
    fun uniform1f(location: WebGLUniformLocation?, x: Float)
    fun uniform1fv(location: WebGLUniformLocation?, v: FloatArray)
    fun uniform1iv(location: WebGLUniformLocation?, v: IntArray)
    fun uniform1l(location: WebGLUniformLocation?, v: IntArray)
    fun uniform2f(location: WebGLUniformLocation?, x: Float, y: Float)
    fun uniform2fv(location: WebGLUniformLocation?, v: FloatArray)
    fun uniform2iv(location: WebGLUniformLocation?, v: IntArray)
    fun uniform2l(location: WebGLUniformLocation?, v: IntArray)
    fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float)
    fun uniform3fv(location: WebGLUniformLocation?, v: FloatArray)
    fun uniform3iv(location: WebGLUniformLocation?, v: IntArray)
    fun uniform3l(location: WebGLUniformLocation?, v: IntArray)
    fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float, w: Float)
    fun uniform4fv(location: WebGLUniformLocation?, v: FloatArray)
    fun uniform4iv(location: WebGLUniformLocation?, v: IntArray)
    fun uniform4l(location: WebGLUniformLocation?, v: IntArray)
fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)
fun uniform3fv(location: WebGLUniformLocation?, v: Array<Float>)
fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int)
fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform3iv(location: WebGLUniformLocation?, v: Array<Int>)
fun uniform4f(location: WebGLUniformLocation?, v: Float32Array)
fun uniform4fv(location: WebGLUniformLocation?, v: Array<Float>)
fun uniform4i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int)
fun uniform4iv(location: WebGLUniformLocation?, v: Int32Array)
fun uniform4iv(location: WebGLUniformLocation?, v: Array<Int>)
fun uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)
fun uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)
fun uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)
fun uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)

fun useProgram(program: WebGLProgram?)
fun validateProgram(program: WebGLProgram?)
fun vertexAttrib1f(index: Int, x: Float)
fun vertexAttrib1fv(index: Int, values: dynamic)
fun vertexAttrib2f(index: Int, x: Float, y: Float)
fun vertexAttrib2fv(index: Int, values: dynamic)
fun vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)
fun vertexAttrib3fv(index: Int, values: dynamic)
fun vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)
fun vertexAttrib4fv(index: Int, values: dynamic)

fun viewport(x: Int, y: Int, width: Int, height: Int)

companion object {
    val DEPTH_BUFFER_BIT: Int
    val STENCIL_BUFFER_BIT: Int
    val COLOR_BUFFER_BIT: Int
    val POINTS: Int
    val LINES: Int
    val LINE_LOOP: Int
    val LINE_STRIP: Int
    val TRIANGLES: Int
    val TRIANGLE_STRIP: Int
    val TRIANGLE_FAN: Int
    val ZERO: Int
    val ONE: Int
    val SRC_COLOR: Int
    val ONE_MINUS_SRC_COLOR: Int
    val SRC_ALPHA: Int
    val ONE_MINUS_SRC_ALPHA: Int
    val DST_COLOR: Int
    val ONE_MINUS_DST_COLOR: Int
    val DST_ALPHA: Int
    val ONE_MINUS_DST_ALPHA: Int
    val BLEND_COLOR: Int
    val CONSTANT_COLOR: Int
    val ONE_MINUS_CONSTANT_COLOR: Int
    val CONSTANT_ALPHA: Int
    val ONE_MINUS_CONSTANT_ALPHA: Int
    val ARRAY_BUFFER: Int
    val ELEMENT_ARRAY_BUFFER: Int
    val ARRAY_BUFFER_BINDING: Int
    val ELEMENT_ARRAY_BUFFER_BINDING: Int
    val STREAM_DRAW: Int
    val STATIC_DRAW: Int
    val DYNAMIC_DRAW: Int
    val BUFFER_SIZE: Int
    val BUFFER_USAGE: Int
    val CURRENT_VERTEX_ATTRIB: Int
    val FRONT: Int
    val BACK: Int
    val FRONT_AND_BACK: Int
    val DEPTH_FACE: Int
    val BLEND: Int
    val DITHER: Int
    val STENCIL_TEST: Int
    val DEPTH_TEST: Int
    val SCISSOR_TEST: Int
    val POLYGON_OFFSET_FILL: Int
    val SAMPLE_ALPHA_TO_COVERAGE: Int
    val SAMPLE_COVERAGE: Int
    val NO_ERROR: Int
    val INVALID_ENUM: Int
    val INVALID_VALUE: Int
    val INVALID_OPERATION: Int
    val OUT_OF_MEMORY: Int
    val CW: Int
    val CCW: Int
    val LINE_WIDTH: Int
    val ALIASED_POINT_SIZE_RANGE: Int
    val ALIASED_LINE_WIDTH_RANGE: Int
    val CULL_FACE_MODE: Int
    val DEPTH_RANGE: Int
    val DEPTH_WRITEMASK: Int
    val DEPTH_CLEAR_VALUE: Int
    val DEPTH_FUNC: Int
    val STENCIL_CLEAR_VALUE: Int
    val STENCIL_FUNC: Int
    val STENCIL_FAIL: Int
    val STENCIL_PASS_DEPTH_FAIL: Int
    val STENCIL_PASS_DEPTH_PASS: Int
    val STENCIL_REF:
FLOAT_VEC3: Int
float val FLOAT_VEC4: Int
        val INT_VEC2: Int
        val INT_VEC3: Int
        val INT_VEC4: Int
        val BOOL: Int
        val BOOL_VEC2: Int
        val BOOL_VEC3: Int
        val BOOL_VEC4: Int
        val SAMPLER_2D: Int
        val VERTEX_ATTRIB_ARRAY_ENABLED: Int
        val VERTEX_ATTRIB_ARRAY_SIZE: Int
        val VERTEX_ATTRIB_ARRAY_STRIDE: Int
        val VERTEX_ATTRIB_ARRAY_TYPE: Int
        val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int
        val VERTEX_ATTRIB_ARRAY_POINTER: Int
        val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING: Int
        val IMPLEMENTATION_COLOR_READ_TYPE: Int
        val IMPLEMENTATION_COLOR_READ_FORMAT: Int
        val LOW_FLOAT: Int
        val MEDIUM_FLOAT: Int
        val HIGH_FLOAT: Int
        val LOW_INT: Int
        val MEDIUM_INT: Int
        val HIGH_INT: Int
        val LOW_FLOAT: Int
        val MEDIUM_FLOAT: Int
        val HIGH_FLOAT: Int
        val LOW_INT: Int
        val MEDIUM_INT: Int
        val HIGH_INT: Int
        val FRAMEBUFFER: Int
        val RENDERBUFFER: Int
        val RGBA4: Int
        val RGB5_A1: Int
        val RGB565: Int
        val DEPTH_COMPONENT16: Int
        val STENCIL_INDEX: Int
        val STENCIL_INDEX8: Int
        val DEPTH_STENCIL: Int
        val RENDERBUFFER_WIDTH: Int
        val RENDERBUFFER_HEIGHT: Int
        val RENDERBUFFER_INTERNAL_FORMAT: Int
        val RENDERBUFFER_RED_SIZE: Int
        val RENDERBUFFER_GREEN_SIZE: Int
        val RENDERBUFFER_BLUE_SIZE: Int
        val RENDERBUFFER_ALPHA_SIZE: Int
        val RENDERBUFFER_DEPTH_SIZE: Int
        val RENDERBUFFER_STENCIL_SIZE: Int
        val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int
        val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int
        val FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int
        val FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int
        val COLOR_ATTACHMENT0: Int
        val DEPTH_ATTACHMENT: Int
        val STENCIL_ATTACHMENT: Int
        val DEPTH_STENCIL_ATTACHMENT: Int
        val NONE: Int
        val FRAMEBUFFER_COMPLETE: Int
        val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int
        val FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT: Int
        val FRAMEBUFFER_UNSUPPORTED: Int
        val FRAMEBUFFER_BINDING: Int
        val RENDERBUFFER_BINDING: Int
        val MAX_RENDERBUFFER_SIZE: Int
        val INVALID_FRAMEBUFFER_OPERATION: Int
        val UNPACK_FLIP_Y_WEBGL: Int
        val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int
        val CONTEXT_LOST_WEBGL: Int
        val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int
        val BROWSER_DEFAULT_WEBGL: Int
        }
}

/**
 * Exposes the JavaScript
[WebGLRenderingContext](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderingContext) to Kotlin
*/
public external abstract class WebGLRenderingContext : WebGLRenderingContextBase, RenderingContext {
companion object {
        val DEPTH_BUFFER_BIT: Int
        val STENCIL_BUFFER_BIT: Int
        val COLOR_BUFFER_BIT: Int
        val POINTS: Int
        val LINES: Int
        val LINE_LOOP: Int
        val LINE_STRIP: Int
        val TRIANGLES: Int
        val TRIANGLE_STRIP: Int
        val TRIANGLE_FAN: Int
        val SRC_COLOR: Int
        val SRC_ALPHA: Int
        val DST_COLOR: Int
        val DST_ALPHA: Int
        val SRC_ALPHA_SATURATE: Int
        val FUNC_ADD: Int
        val BLEND_EQUATION: Int
        val BLEND_EQUATION_RGB: Int
        val BLEND_EQUATION_ALPHA: Int
        val FUNC_SUBTRACT: Int
        val BLEND_DST_RGB: Int
        val BLEND_DST_RGB: Int
        val BLEND_DST_ALPHA: Int
        val BLEND_SRC_ALPHA: Int
        val CONSTANT_COLOR: Int
        val ONE_MINUS_CONSTANT_COLOR: Int
        val CONSTANT_ALPHA: Int
        val ONE_MINUS_CONSTANT_ALPHA: Int
        val BLEND_COLOR: Int
        val ARRAY_BUFFER: Int
        val ELEMENT_ARRAY_BUFFER: Int
        val STREAM_DRAW: Int
        val STATIC_DRAW: Int
        val DYNAMIC_DRAW: Int
        val BUFFER_SIZE: Int
        val
TEXTURE_CUBE_MAP_POSITIVE_Z: Int
        val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int
        val MAX_CUBE_MAP_TEXTURE_SIZE: Int
        val TEXTURE0: Int
        val TEXTURE1: Int
        val TEXTURE2: Int
        val TEXTURE3: Int
        val TEXTURE4: Int
        val TEXTURE5: Int
        val TEXTURE6: Int
        val TEXTURE7: Int
        val TEXTURE8: Int
        val TEXTURE9: Int
        val TEXTURE10: Int
        val TEXTURE11: Int
        val TEXTURE12: Int
        val TEXTURE13: Int
        val TEXTURE14: Int
        val TEXTURE15: Int
        val TEXTURE16: Int
        val TEXTURE17: Int
        val TEXTURE18: Int
        val TEXTURE19: Int
        val TEXTURE20: Int
        val TEXTURE21: Int
        val TEXTURE22: Int
        val TEXTURE23: Int
        val TEXTURE24: Int
        val TEXTURE25: Int
        val TEXTURE26: Int
        val TEXTURE27: Int
        val TEXTURE28: Int
        val TEXTURE29: Int
        val TEXTURE30: Int
        val TEXTURE31: Int
        val ACTIVE_TEXTURE: Int
        val REPEAT: Int
        val CLAMP_TO_EDGE: Int
        val MIRRORED_REPEAT: Int
        val FLOAT_VEC2: Int
        val FLOAT_VEC3: Int
        val FLOAT_VEC4: Int
        val INT_VEC2: Int
        val INT_VEC3: Int
        val INT_VEC4: Int
        val BOOL: Int
        val BOOL_VEC2: Int
        val BOOL_VEC3: Int
        val BOOL_VEC4: Int
        val FLOAT_MAT2: Int
        val FLOAT_MAT3: Int
        val FLOAT_MAT4: Int
        val SAMPLER_2D: Int
        val SAMPLER_CUBE: Int
        val VERTEX_ATTRIB_ARRAY_ENABLED: Int
        val VERTEX_ATTRIB_ARRAY_SIZE: Int
        val VERTEX_ATTRIB_ARRAY_STRIDE: Int
        val VERTEX_ATTRIB_ARRAY_TYPE: Int
        val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int
        val VERTEX_ATTRIB_ARRAY_POINTER: Int
        val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING: Int
        val IMPLEMENTATION_COLOR_READ_TYPE: Int
        val IMPLEMENTATION_COLOR_READ_FORMAT: Int
        val LOW_FLOAT: Int
        val MEDIUM_FLOAT: Int
        val HIGH_FLOAT: Int
        val LOW_INT: Int
        val MEDIUM_INT: Int
        val HIGH_INT: Int
        val FRAMEBUFFER: Int
        val RENDERBUFFER: Int
        val RGBA4: Int
        val RGB5_A1: Int
        val RGB565: Int
        val DEPTH_COMPONENT16: Int
        val STENCIL_INDEX: Int
        val STENCIL_INDEX8: Int
        val DEPTH_STENCIL: Int
        val RENDERBUFFER_WIDTH: Int
        val RENDERBUFFER_HEIGHT: Int
        val RENDERBUFFER_INTERNAL_FORMAT: Int
        val RENDERBUFFER_RED_SIZE: Int
        val RENDERBUFFER_GREEN_SIZE: Int
        val RENDERBUFFER_BLUE_SIZE: Int
        val RENDERBUFFER_ALPHA_SIZE: Int
        val RENDERBUFFER_DEPTH_SIZE: Int
        val RENDERBUFFER_STENCIL_SIZE: Int
        val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int
        val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int
        val FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int
        val FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int
        val COLOR_ATTACHMENT0: Int
        val DEPTH_ATTACHMENT: Int
        val STENCIL_ATTACHMENT: Int
        val DEPTH_STENCIL_ATTACHMENT: Int
        val NONE: Int
        val FRAMEBUFFER_COMPLETE: Int
        val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int
        val FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT: Int
        val FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int
        val FRAMEBUFFER_UNSUPPORTED: Int
        val FRAMEBUFFER_BINDING: Int
        val RENDERBUFFER_BINDING: Int
        val INVALID_FRAMEBUFFER_OPERATION: Int
        val UNPACK_FLIP_Y_WEBGL: Int
        val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int
        val CONTEXT_LOST_WEBGL: Int
        val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int
        val BROWSER_DEFAULT_WEBGL: Int
        val CONTEXT_CROSS_ORIGIN_plane

        /**
         * Exposes the JavaScript
         */
        public external open class WebGLContextEvent(type: String, eventInit: WebGLContextEventInit =
            definedExternally) : Event {
            val statusMessage: String
            override fun toString(): String = type
            companion object {
                val NONE: Short
                val CAPTURING_PHASE: Short
                val AT_TARGET: Short
                val BUBBLING_PHASE: Short
            }

            @kotlin.internal.InlineOnly
            public inline fun

            Open Source Used In DNAC Platform Antelope 14092
    @n val o = js("({})")@n
    o["statusMessage"] = statusMessage
    o["bubbles"] = bubbles
    o["Cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 */
public external open

class ArrayBuffer(length: Int) : BufferData-source {
    open val byteLength: Int
    fun slice(begin: Int, end: Int = definedExternally): ArrayBuffer

    companion object {
        fun isView(value: Any?): Boolean
    }
}

/**
 * Exposes the JavaScript
 */
public external interface ArrayBufferView : BufferData-source {
    val buffer: ArrayBuffer
    val byteOffset: Int
    val byteLength: Int
    fun set(array: ArrayBuffer, offset: Int = definedExternally)
    fun set(array: Array<Byte>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int)

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Int8Array.get(index: Int): Byte = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Int8Array.set(index: Int, value: Byte) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
 */
public external open

class Uint8Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint8Array)
    constructor(array: Array<Byte>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int
    override val byteLength: Int
    fun set(array: Uint8Array, offset: Int = definedExternally)
    fun set(array: Array<Byte>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint8Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint8Array.get(index: Int): Byte = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint8Array.set(index: Int, value: Byte) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
 */

Open Source Used In DNAC Platform Antelope 14094

[Uint16Array](https://developer.mozilla.org/en/docs/Web/API/Uint16Array) to Kotlin

```kotlin
public external open class Uint16Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint16Array)
    constructor(array: Array<Short>), offset: Int = definedExternally, length: Int = definedExternally
    override var open val buffer: ArrayBuffer

    override var override val byteLength: Int
    override fun set(array: Uint16Array, offset: Int = definedExternally)

    fun set(array: Array<Short>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint16Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint16Array.get(index: Int): Short = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint16Array.set(index: Int, value: Short) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
[Int32Array](https://developer.mozilla.org/en/docs/Web/API/Int32Array) to Kotlin

```kotlin
public external open class Int32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Int32Array)
    constructor(array: Array<Int>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int

    override val byteLength: Int
    fun set(array: Int32Array, offset: Int = definedExternally)
    fun set(array: Array<Int>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Int32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Int32Array.get(index: Int): Int = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Int32Array.set(index: Int, value: Int) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
[Uint32Array](https://developer.mozilla.org/en/docs/Web/API/Uint32Array) to Kotlin

```kotlin
public external open class Uint32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Uint32Array)
    constructor(array: Array<Int>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int

    override val byteLength: Int
    fun set(array: Uint32Array, offset: Int = definedExternally)
    fun set(array: Array<Int>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Uint32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Uint32Array.get(index: Int): Int = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Uint32Array.set(index: Int, value: Int) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
[Float32Array](https://developer.mozilla.org/en/docs/Web/API/Float32Array) to Kotlin

```kotlin
public external open class Float32Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Float32Array)
    constructor(array: Array<Float>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer
    override val byteOffset: Int

    override val byteLength: Int
    fun set(array: Float32Array, offset: Int = definedExternally)
    fun set(array: Array<Float>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Float32Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Float32Array.get(index: Int): Float = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Float32Array.set(index: Int, value: Float) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
[Float64Array](https://developer.mozilla.org/en/docs/Web/API/Float64Array) to Kotlin

```kotlin
public external open class Float64Array : ArrayBufferView {
    constructor(length: Int)
    constructor(array: Float64Array)
    constructor(array: Array<Double>)
    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)
    open val length: Int
    override val buffer: ArrayBuffer

    override val byteOffset: Int
    override val byteLength: Int
    fun set(array: Float64Array, offset: Int = definedExternally)
    fun set(array: Array<Double>, offset: Int = definedExternally)
    fun subarray(start: Int, end: Int): Float64Array

    companion object {
        val BYTES_PER_ELEMENT: Int
    }
}

@kotlin.internal.InlineOnly
public inline operator fun Float64Array.get(index: Int): Double = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Float64Array.set(index: Int, value: Double) { asDynamic()[index] = value }

/**
 * Exposes the JavaScript
[DataView](https://developer.mozilla.org/en/docs/Web/API/DataView) to Kotlin

```kotlin
public external open class DataView(buffer: ArrayBuffer, byteOffset: Int = definedExternally, byteLength: Int = definedExternally)

```
org.w3c.workers.*

import org.w3c.xhr.*

public external abstract class MediaList : ItemArrayLike<String> {
    open var mediaText: String
    fun appendMedium(medium: String)
    fun deleteMedium(medium: String)

    override fun item(index: Int): String?
}

@kotlin.internal.InlineOnly
public inline operator fun MediaList.get(index: Int): String? = asDynamic()[index]

/**
 * Exposes the JavaScript [StyleSheet](https://developer.mozilla.org/en/docs/Web/API/StyleSheet) to Kotlin
 */
public external abstract class StyleSheet {
    open val type: String
    open val href: String?
    open val ownerNode: 
    open val parentStyleSheet: StyleSheet
    open val disabled: Boolean

    open var media: MediaList
        open var disabled: Boolean
}

/**
 * Exposes the JavaScript [CSSStyleSheet](https://developer.mozilla.org/en/docs/Web/API/CSSStyleSheet) to Kotlin
 */
public external abstract class CSSStyleSheet : StyleSheet {
    open val ownerRule: StyleSheet?
    open val cssRules:
        open val media: MediaList
        open val disabled: Boolean
}

/**
 * Exposes the JavaScript [StyleSheetList](https://developer.mozilla.org/en/docs/Web/API/StyleSheetList) to Kotlin
 */
public external abstract class StyleSheetList : ItemArrayLike<StyleSheet> {
    override fun item(index: Int): StyleSheet?
}

@kotlin.internal.InlineOnly
public inline operator fun StyleSheetList.get(index: Int): StyleSheet? = asDynamic()[index]

/**
 * Exposes the JavaScript [LinkStyle](https://developer.mozilla.org/en/docs/Web/API/LinkStyle) to Kotlin
 */
public external interface LinkStyle {
    val sheet: StyleSheet
        get() = definedExternally
}

/**
 * Exposes the JavaScript [CSSRuleList](https://developer.mozilla.org/en/docs/Web/API/CSSRuleList) to Kotlin
 */
public external abstract class CSSRuleList : ItemArrayLike<CSSRule> {
    override fun item(index: Int): CSSRule?
}

@kotlin.internal.InlineOnly
public inline operator fun CSSRuleList.get(index: Int): CSSRule? = asDynamic()[index]

/**
 * Exposes the JavaScript [CSSRule](https://developer.mozilla.org/en/docs/Web/API/CSSRule) to Kotlin
 */
public external abstract class CSSRule {
    open val type: Short
    open var cssText: String
    open val parentRule: CSSRule
    open val parentStyleSheet: CSSStyleSheet

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSStyleRule](https://developer.mozilla.org/en/docs/Web/API/CSSStyleRule) to Kotlin
 */
public external abstract class CSSStyleRule : CSSRule {
    open var selectorText: String
    open val style: CSSStyleDeclaration

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
        val CHARSET_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSImportRule](https://developer.mozilla.org/en/docs/Web/API/CSSImportRule) to Kotlin
 */
public external abstract class CSSImportRule : CSSRule {
    open val href: String
    open val media: MediaList
    open val styleSheet: CSSStyleSheet

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSGroupingRule](https://developer.mozilla.org/en/docs/Web/API/CSSGroupingRule) to Kotlin
 */
public external abstract class CSSGroupingRule : CSSRule {
    open val cssRules: CSSRuleList
    fun insertRule(rule: String, index: Int): Int
    fun deleteRule(index: Int)

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/**
 * Exposes the JavaScript [CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin
 */
public external abstract class CSSMediaRule : CSSGroupingRule {
    open val media: MediaList

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}
/*
 * Exposes the JavaScript 
 [CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin
 */
public external abstract class CSSPageRule : CSSGroupingRule {
    open var selectorText: String
    open val style: CSSStyleDeclaration

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/*
 * Exposes the JavaScript 
 [CSSMarginRule](https://developer.mozilla.org/en/docs/Web/API/CSSMarginRule) to Kotlin
 */
public external abstract class CSSMarginRule : CSSRule {
    open val name: String
    open val style: CSSStyleDeclaration

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/*
 * Exposes the JavaScript 
 [CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin
 */
public external abstract class CSSNamespaceRule : CSSRule {
    open val namespaceURI: String
    open val prefix: String

    companion object {
        val STYLE_RULE: Short
        val CHARSET_RULE: Short
        val IMPORT_RULE: Short
        val MEDIA_RULE: Short
        val FONT_FACE_RULE: Short
        val PAGE_RULE: Short
        val MARGIN_RULE: Short
        val NAMESPACE_RULE: Short
    }
}

/*
 * Exposes the JavaScript 
 [CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin
 */
public external abstract class CSSStyleDeclaration : ItemArrayLike<String> {
    open var cssText: String
    open val parentRule: CSSRule?
    open var cssFloat: String
    open var alignItems: String
    open var alignSelf: String
    open var animation: String
    open var animationDelay: String
    open var animationDirection: String
    open var animationDuration: String
    open var animationFillMode: String
    open var animationIterationCount: String
    open var animationName: String
    open var animationPlayState: String
    open var animationTimingFunction: String
    open var background: String
    open var backgroundAttachment: String
    open var backgroundClip: String
    open var backgroundColor: String
    open var backgroundImage: String
    open var backgroundOrigin: String
    open var backgroundPosition: String
    open var backgroundRepeat: String
    open var backgroundSize: String
    open var border: String
    open var borderBottom: String
    open var borderBottomColor: String
    open var borderBottomLeftRadius: String
    open var borderBottomRightRadius: String
    open var borderBottomStyle: String
    open var borderBottomWidth: String
    open var borderCollapse: String
    open var borderColor: String
    open var borderImage: String
    open var borderImageOutset: String
    open var borderImageRepeat: String
    open var borderImageSlice: String
    open var borderImageSource: String
    open var borderImageWidth: String
    open var borderLeft: String
    open var borderLeftColor: String
    open var borderLeftStyle: String
    open var borderTop: String
    open var borderTopColor: String
    open var borderTopLeftRadius: String
    open var borderTopRightRadius: String
    open var borderTopStyle: String
    open var borderTopWidth: String
    open var borderSpacing: String
    open var borderBottom: String
    open var borderBottomColor: String
    open var borderBottomStyle: String
    open var borderBottomWidth: String
    open var borderStyle: String
    open var borderTop: String
    open var borderTopColor: String
    open var borderTopLeftRadius: String
    open var borderTopRightRadius: String
    open var borderTopStyle: String
    open var borderTopWidth: String
    open var borderLeftWidth: String
    open var borderTopLeftWidth: String
    open var borderTopRightWidth: String
    open var borderLeftColor: String
    open var borderTopColor: String
    open var borderLeftStyle: String
    open var borderTopStyle: String
    open var columnCount: String
    open var columnFill: String
    open var columnGap: String
    open var columnRule: String
    open var columnRuleColor: String
    open var columnRuleStyle: String
    open var columnRuleWidth: String
    open var columnSpan: String
    open var columnWidth: String
    open var columns: String
    open var content: String
    open var counterIncrement: String
    open var counterReset: String
    open var cursor: String
    open var direction: String
    open var display: String
    open var emptyCells: String
    open var filter: String
    open var flex: String
    open var flexBasis: String
    open var flexDirection: String
    open var flexGrow: String
    open var flexShrink: String
    open var flexWrap: String
    open var font: String
    open var fontFamily: String

String\nopen var fontFeatureSettings: String\nopen var fontKerning: String\nopen var fontLanguageOverride: String\nopen var fontSize: String\nopen var fontSizeAdjust: String\nopen var fontStretch: String\nopen var fontStyle: String\nopen var fontSynthesis: String\nopen var fontVariant: String\nopen var fontVariantAlternates: String\nopen var fontVariantCaps: String\nopen var fontVariantEastAsian: String\nopen var fontVariantLigatures: String\nopen var fontVariantNumeric: String\nopen var fontVariantPosition: String\nopen var fontWeight: String\nopen var hangingPunctuation: String\nopen var height: String\nopen var hyphens: String\nopen var imageOrientation: String\nopen var imageRendering: String\nopen var imageResolution: String\nopen var imeMode: String\nopen var justifyContent: String\nopen var letterSpacing: String\nopen var lineBreak: String\nopen var lineHeight: String\nopen var listStyle: String\nopen var listStyleImage: String\nopen var listStylePosition: String\nopen var listStyleType: String\nopen var margin: String\nopen var marginBottom: String\nopen var marginLeft: String\nopen var marginRight: String\nopen var marginTop: String\nopen var mark: String\nopen var markAfter: String\nopen var marks: String\nopen var marqueeDirection: String\nopen var marqueePlayCount: String\nopen var marqueeSpeed: String\nopen var marqueeStyle: String\nopen var mask: String\nopen var maskType: String\nopen var maxHeight: String\nopen var maxWidth: String\nopen var minHeight: String\nopen var minMaxHeight: String\nopen var navDown: String\nopen var navIndex: String\nopen var navLeft: String\nopen var navRight: String\nopen var navUp: String\nopen var objectFit: String\nopen var objectPosition: String\nopen var opacity: String\nopen var orphans: String\nopen var outline: String\nopen var outlineColor: String\nopen var outlineOffset: String\nopen var outlineStyle: String\nopen var overFlowWrap: String\nopen var overFlowX: String\nopen var overFlowY: String\nopen var padding: String\nopen var paddingBottom: String\nopen var paddingLeft: String\nopen var paddingRight: String\nopen var paddingTop: String\nopen var pageBreakAfter: String\nopen var pageBreakBefore: String\nopen var pageBreakInside: String\nopen var perspective: String\nopen var perspectiveOrigin: String\nopen var phonemes: String\nopen var position: String\nopen var quotes: String\nopen var resize: String\nopen var rest: String\nopen var restAfter: String\nopen var restBefore: String\nopen var right: String\nopen var tabSize: String\nopen var tableLayout: String\nopen var textAlign: String\nopen var textAlignLast: String\nopen var textCombineUpright: String\nopen var textDecoration: String\nopen var textDecorationColor: String\nopen var textDecorationLine: String\nopen var textDecorationStyle: String\nopen var textIndent: String\nopen var textJustify: String\nopen var textOrientation: String\nopen var textOverflow: String\nopen var textShadow: String\nopen var textTransform: String\nopen var textDecorationLine: String\nopen var textDecorationStyle: String\nopen var textUnderlinePosition: String\nopen var top: String\nopen var transform: String\nopen var transformOrigin: String\nopen var transition: String\nopen var transitionDelay: String\nopen var transitionDuration: String\nopen var transitionProperty: String\nopen var transitionTimingFunction: String\nopen var unicodeBidi: String\nopen var verticalAlign: String\nopen var visibility: String\nopen var voiceBalance: String\nopen var voiceDuration: String\nopen var voicePitch: String\nopen var voicePitchRange: String\nopen var voiceRate: String\nopen var voiceStress: String\nopen var voiceVolume: String\nopen var whiteSpace: String\nopen var widows: String\nopen var width: String\nopen var wordBreak: String\nopen var wordSpacing: String\nopen var wordWrap: String\nopen var writingMode: String\nopen var zIndex: String\nopen var _dashed_attribute: String\nfun getPropertyValue(property: String): String\nfun getPropertyValue(property: String): String\nfun setProperty(property: String, value: String, priority: String = definedExternally): String\nfun setProperty(property: String, value: String): String
fun setPropertyPriority(property: String, priority: String): String\nfun removeProperty(property: String): String
override fun item(index: Int): String?

@kotlin.internal.InlineOnly
public inline operator fun CSSStyleDeclaration.get(index: Int): String?

public external interface ElementCSSInlineStyle {
    val style: CSSStyleDeclaration
}

/**
 * Exposes the JavaScript
[
CSS](https://developer.mozilla.org/en/docs/Web/API/CSS)
to Kotlin
*/
public external abstract class CSS {

Open Source Used in DNAC Platform Antelope 14098
companion object {
    fun escape(ident: String): String
}

/**
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package org.w3c.dom.encryptedmedia

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

public external interface MediaKeySystemConfiguration {
    var label: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var initDataTypes: Array<String>? /* = arrayOf() */
        get() = definedExternally
        set(value) = definedExternally
    var audioCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() */
        get() = definedExternally
        set(value) = definedExternally
    var videoCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() */
        get() = definedExternally
        set(value) = definedExternally
    var distinctiveIdentifier: MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL */
        get() = definedExternally
        set(value) = definedExternally
    var persistentState: MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL */
        get() = definedExternally
        set(value) = definedExternally
    var sessionTypes: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["label"] = label
    o["initDataTypes"] = initDataTypes
    o["audioCapabilities"] = audioCapabilities
    o["videoCapabilities"] = videoCapabilities
    o["distinctiveIdentifier"] = distinctiveIdentifier
    o["persistentState"] = persistentState
    o["sessionTypes"] = sessionTypes
    return o
}

public external interface MediaKeySystemMediaCapability {
    var contentType: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var robustness: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun MediaKeySystemMediaCapability(contentType: String? = "", robustness: String? = "") : MediaKeySystemMediaCapability {
    val o = js("({})")
    o["contentType"] = contentType
    o["robustness"] = robustness
    return o
}

public external abstract class MediaKeySystemAccess {
    open val keySystem: String
    fun getConfiguration(): MediaKeySystemConfiguration
    fun createMediaKeys(): Promise<MediaKeys>
}

public external abstract class MediaKeys {
    fun createSession(sessionType: MediaKeySessionType = definedExternally): MediaKeySession
}

public external abstract class MediaKeySession : EventTarget {
    open val sessionId: String
    open val expiration: Double
    open val closed: Promise<Unit>
    open val keyStatuses: MediaKeyStatusMap
    open var onkeystatuschange: (Event) -> dynamic
    open var onmessage: ((MessageEvent) -> dynamic)
    fun generateRequest(initDataType: String, initData: dynamic): Promise<
        MediaKeys>
        fun getConfiguration(): MediaKeySystemConfiguration
        fun createMediaKeys(): MediaKeySession
        fun setServerCertificate(serverCertificate: dynamic): Promise<
            Boolean>
        fun createSession(sessionType: MediaKeySessionType = definedExternally): MediaKeySession
        fun update(response: dynamic): Promise<
            Boolean>
        fun remove(): Promise<
            Boolean>
        fun close(): Promise<
            Boolean>
        fun size: Int
        fun has(keyId: dynamic): Boolean
        fun get(keyId: dynamic): Any?
        fun open val openSize:
        fun open(val message: MediaKeyMessageEvent)): ReturnType{
            var message: MediaKeyMessageEvent
        open val open message: ArrayBuffer
NONE: Short
val CAPTURING_PHASE: Short
val AT_TARGET: Short
val BUBBLING_PHASE: Short

val messageType: MediaKeyMessageType?
val get() = definedExternally
val set(value) = definedExternally

val message: ArrayBuffer?
val get() = definedExternally
val set(value) = definedExternally

val composed: Boolean?
val get() = definedExternally
val set(value) = definedExternally

val bubbles: Boolean?
val get() = definedExternally
val set(value) = definedExternally

val cancelable: Boolean?
val get() = definedExternally
val set(value) = definedExternally

@kotlin.internal.InlineOnly
    var o = js("({})")
    o["messageType"] = messageType
    o["message"] = message
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

fun MediaKeyMessageEventInit(): Event {
    open val initDataType: String?
    var initData: ArrayBuffer?
    return null

    var message: ArrayBuffer?
    var eventInitDict: MediaKeyMessageEventInit = definedExternally

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external open class MediaEncryptedEvent(type: String, eventInitDict: MediaEncryptedEventInit = definedExternally) : Event {
        open val initDataType: String?
        var initDataType: String? /* = "" */
        val initData: ArrayBuffer?
        var initData: ArrayBuffer? /* = null */

        companion object {
            val NONE: Short
            val CAPTURING_PHASE: Short
            val AT_TARGET: Short
            val BUBBLING_PHASE: Short
        }

        public external interface MediaEncryptedEventInit : EventInit {
            var initDataType: String? /* = "" */
            var initData: ArrayBuffer? /* = null */
        }

        @kotlin.internal.InlineOnly
            val o = js("({})")
            o["initDataType"] = initDataType
            o["initData"] = initData
            o["bubbles"] = bubbles
            o["cancelable"] = cancelable
            o["composed"] = composed
            return o
        }
    }

    public external interface MediaKeysRequirement {
        companion object
    }


    companion object
    @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
    public external interface MediaKeySessionType {
        companion object
    }

    public inline val MediaKeySessionType.Companion.TEMPORARY: MediaKeySessionType get() = "temporary".asDynamic().unsafeCast<MediaKeySessionType>()

    public inline val MediaKeySessionType.Companion.PERSISTENT_LICENSE: MediaKeySessionType get() = "persistent-license".asDynamic().unsafeCast<MediaKeySessionType>()

    companion object
    @Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
    public external interface MediaKeyStatus {
        companion object
    }

    public inline val MediaKeyStatus.Companion.USABLE: MediaKeyStatus get() = "usable".asDynamic().unsafeCast<MediaKeyStatus>()

    public inline val MediaKeyStatus.Companion.EXPIRED: MediaKeyStatus get() = "expired".asDynamic().unsafeCast<MediaKeyStatus>()

    public inline val MediaKeyStatus.Companion.RELEASED: MediaKeyStatus get() = "released".asDynamic().unsafeCast<MediaKeyStatus>()

    public inline val MediaKeyStatus.Companion.OUTPUT_RESTRICTED: MediaKeyStatus get() = "output-restricted".asDynamic().unsafeCast<MediaKeyStatus>()

    public inline val MediaKeyStatus.Companion.OUTPUT_DOWNSCALED: MediaKeyStatus get() = "output-downscaled".asDynamic().unsafeCast<MediaKeyStatus>()

    public inline val MediaKeyStatus.Companion.STATUS_PENDING: MediaKeyStatus get() = "status-pending".asDynamic().unsafeCast<MediaKeyStatus>()

    public inline val MediaKeyStatus.Companion.INTERNAL_ERROR: MediaKeyStatus get() = "internal-error".asDynamic().unsafeCast<MediaKeyStatus>()
```kotlin
o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierFnLock"] = modifierFnLock
    o["modifierHyper"] = modifierHyper
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["modifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 * [WheelEvent](https://developer.mozilla.org/en/docs/Web/API/WheelEvent) to Kotlin
 */
public external open class WheelEvent(type: String, eventInitDict: WheelEventInit = definedExternally) : MouseEvent {
    open val deltaX: Double
    open val deltaY: Double
    open val deltaZ: Double

    companion object {
        val DOM_DELTA_PIXEL: Int
        val DOM_DELTA_LINE: Int
        val DOM_DELTA_PAGE: Int
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface WheelEventInit : MouseEventInit {
    var deltaX: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var deltaY: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var deltaZ: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("("")")
    o["deltaX"] = deltaX
    o["deltaY"] = deltaY
    o["deltaZ"] = deltaZ
    o["screenX"] = screenX
    o["screenY"] = screenY
    o["clientX"] = clientX
    o["clientY"] = clientY
    o["button"] = button
    o["buttons"] = buttons
    o["relatedTarget"] = relatedTarget
    o["region"] = region
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierFnLock"] = modifierFnLock
    o["modifierHyper"] = modifierHyper
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["modifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 * [InputEvent](https://developer.mozilla.org/en/docs/Web/API/InputEvent) to Kotlin
 */
public external open class InputEvent(type: String, eventInitDict: InputEventInit = definedExternally) : UIEvent {
    open val data: String
    open val isComposing: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface InputEventInit : UIEventInit {
    var data: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var isComposing: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("("")")
    o["data"] = data
    o["isComposing"] = isComposing
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript
 * [KeyboardEvent](https://developer.mozilla.org/en/docs/Web/API/KeyboardEvent) to Kotlin
 */
public external open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent {
    open val data: String
    open val isComposing: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface KeyboardEventInit : KeyboardEventInit {
    var data: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var isComposing: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("("")")
    o["data"] = data
    o["isComposing"] = isComposing
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}
```

Open Source Used in DNAC Platform Antelope 14103
open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent {

    // Properties
    open val key: String
    open val code: String
    open val location: Int
    open val ctrlKey: Boolean
    open val shiftKey: Boolean
    open val altKey: Boolean
    open val metaKey: Boolean
    open val repeat: Boolean
    open val isComposing: Boolean
    open val charCode: Int
    open val keyCode: Int
    open val which: Int

    fun getModifierState(keyArg: String): Boolean

    companion object {
        val DOM_KEY_LOCATION_STANDARD: Int
        val DOM_KEY_LOCATION_LEFT: Int
        val DOM_KEY_LOCATION_RIGHT: Int
        val DOM_KEY_LOCATION_NUMPAD: Int
    }
}

public external interface KeyboardEventInit : EventModifierInit {

    var key: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var code: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var location: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var repeat: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var isComposing: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
isComposing: Boolean? = false, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false,
metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn:
Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock:
Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol:
    val o = js("({})")
    o["key"] = key
    o["code"] = code
    o["location"] = location
    o["repeat"] = repeat
    o["isComposing"] = isComposing
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["modifierAltGraph"] = modifierAltGraph
    o["modifierCapsLock"] = modifierCapsLock
    o["modifierFn"] = modifierFn
    o["modifierFnLock"] = modifierFnLock
    o["modifierHyper"] = modifierHyper
    o["modifierNumLock"] = modifierNumLock
    o["modifierScrollLock"] = modifierScrollLock
    o["modifierSuper"] = modifierSuper
    o["modifierSymbol"] = modifierSymbol
    o["modifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
* Exposes the JavaScript
[CompositionEvent](https://developer.mozilla.org/en/docs/Web/API/CompositionEvent) to Kotlin *
*/
public external open class CompositionEvent(type: String, eventInitDict: CompositionEventInit = definedExternally) :
UIEvent {

    // Properties
    open val data: String

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface CompositionEventInit : UIEventInit {

    var data: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun CompositionEventInit(data: String? = "", view: Window? = null, detail: Int? = 0, bubbles:
    val o = js("({})")
    o["data"] = data
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
* Exposes the JavaScript
[Event](https://developer.mozilla.org/en/docs/Web/API/Event) to Kotlin *
*/
public external open class Event(type: String, eventInitDict: EventInit = definedExternally) {

    // Properties
    open val type: String
    open val target: EventTarget?
    open val currentTarget: EventTarget?
    open val eventPhase: Short
    open val bubbles: Boolean
    open val defaultPrevented: Boolean
    open val composed: Boolean
    open val isTrusted: Boolean
    open val timeStamp: Number
    fun composedPath():
    Array<EventTarget>
    fun stopPropagation():
    fun stopImmediatePropagation():
    fun preventDefault():
    fun initEvent(type: String, bubbles: Boolean, cancelable: Boolean? = null, companion object {

    Short
    val CAPTURING_PHASE: Short
    val AT_TARGET: Short
    val BUBBLING_PHASE: Short
    }
Short

|n|\n|n/* Exposes the JavaScript

[EventTarget](https://developer.mozilla.org/en/docs/Web/API/EventTarget) to Kotlin

```kotlin
fun addEventListener(type: String, callback: EventListener?, options: dynamic =
definedExternally):
fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally):
fun removeEventListener(type: String, callback: EventListener?, options: dynamic =
definedExternally):
fun removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally):
fun dispatchEvent(event: Event): Boolean
```

[EventTarget](https://developer.mozilla.org/en/docs/Web/API/EventTarget) to Kotlin

```kotlin
class EventTarget {
    fun addEventListener(type: String, callback: EventListener?, options: dynamic =
definedExternally):
    fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally):
    fun removeEventListener(type: String, callback: EventListener?, options: dynamic =
definedExternally):
    fun removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic =
definedExternally):
}
```

### Kotlin Programming Language contributors:

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

* NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!

---

**Public inline abstract class HTMLAllCollection {**

```kotlin
open val length: Int
fun item(nameOrIndex: String = definedExternally): UnionElementOrHTMLCollection?
fun namedItem(name: String): UnionElementOrHTMLCollection?
```

---

**Public inline operator fun HTMLAllCollection.get(index: Int): Element? =**

```kotlin
asDynamic()[index]
```

---

**Public inline operator fun HTMLAllCollection.get(name: String): UnionElementOrHTMLCollection? =**

```kotlin
asDynamic()[name]
```

---

**Public abstract class HTMLFormControlsCollection : HTMLCollection**

---

**Public abstract class RadioNodeList : NodeList, UnionElementOrRadioNodeList {**

```kotlin
open var value: String
```

---

**Public abstract class HTMLOptionsCollection : HTMLCollection {**

```kotlin
override var length: Int
open var selectedIndex: Int
fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally): UnionElementOrHTMLCollection
fun remove(index: Int)
```

---

**Public inline operator fun HTMLOptionsCollection.set(index: Int, option: HTMLOptionElement?):**

```kotlin
asDynamic()[index] = option
```

---

**Public abstract class HTMLElement : Element, GlobalEventHandlers, DocumentAndElementEventHandlers,**

```kotlin
open var title: String
open var lang: String
open var translate: Boolean
open var dir: String
open var dataset: DOMStringMap
open var hidden: Boolean
open var tabIndex: Int
open var accessKey: String
open var accessKeyLabel: String
open var draggable: Boolean
open val dropzone: DOMTokenList
open var contextMenu: HTMLMenuElement
open var spellcheck: Boolean
open var innerText: String
open var offsetParent: Element
open val offsetTop: Int
open val offsetLeft: Int
open val offsetWidth: Int
open val offsetHeight: Int
open fun click(): Unit
open fun focus(): Unit
open fun blur(): Unit
open fun forceSpellCheck(): Unit
```

---

**Public inline abstract class HTMLFormControlsCollection: HTMLCollection**

```kotlin
fun add(element: UnionHTMLFormElementOrHTMLCollection, before: dynamic = definedExternally): UnionElementOrHTMLCollection
fun remove(index: Int)
```

---

**Public abstract class HTMLFormElement: HTMLElement**

```kotlin
open var type: String
open var id: String
open var name: String
open var value: String
open var checked: Boolean
open var disabled: Boolean
open var noValidate: Boolean
open var autofocus: Boolean
open var required: Boolean
open var readonly: Boolean
open var spellcheck: Boolean
open var innerText: String
open var offsetParent: Element
open val offsetTop: Int
open val offsetLeft: Int
open val offsetWidth: Int
open val offsetHeight: Int
open fun click(): Unit
open fun focus(): Unit
open fun blur(): Unit
open fun forceSpellCheck(): Unit
```
/**
 * Exposes the JavaScript [HTMLUnknownElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUnknownElement) to Kotlin
 */
public external abstract class HTMLUnknownElement : HTMLElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [DOMStringMap](https://developer.mozilla.org/en/docs/Web/API/DOMStringMap) to Kotlin
 */
public external abstract class DOMStringMap

@kotlin.internal.InlineOnly
public inline operator fun DOMStringMap.get(name: String): String? = asDynamic()[name]

@kotlin.internal.InlineOnly
public inline operator fun DOMStringMap.set(name: String, value: String) { asDynamic()[name] = value }

/**
 * Exposes the JavaScript [HTMLHtmlElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHtmlElement) to Kotlin
 */
public external abstract class HTMLHtmlElement : HTMLElement {
    open var version: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLHeadElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadElement) to Kotlin
 */
public external abstract class HTMLHeadElement : HTMLElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTitleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTitleElement) to Kotlin
 */
public external abstract class HTMLTitleElement : HTMLElement {
    open var text: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
Open Source Used In DNAC Platform Antelope 14108
Kotlin

/* public external interface HTMLHyperlinkElementUtils */
val href: String
val origin: String
val protocol: String
val username: String
val password: String
val host: String
val hostname: String
val port: String
val pathname: String
val search: String
val hash: String

val open var cite: String
val dateTime: String

val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

val open var src: String
val srcset: String
val sizes: String
val media: String
val type: String
val alt: String
val src: String
val naturalWidth: Int
val naturalHeight: Int
val complete: Boolean
val lowsrc: String
val align: String
val hspace: Int
val vspace: Int
val longDesc: String
val referrerPolicy: String
val name: String
val crossOrigin: String?
val useMap: String
val isMap: Boolean
val border: String
val x: Int
val y: Int

val open var onerror: String
Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short

/**
 * Exposes the JavaScript
[HTMLIFrameElement](https://developer.mozilla.org/en/docs/Web/API/HTMLIFrameElement) to Kotlin
*/
public external abstract class HTMLIFrameElement : HTMLElement {
    open var src: String
    open var srcdoc: String
    open var name: String
    open val sandbox: DOMTokenList
    open var allowFullscreen: Boolean
    open var allowUserMedia: Boolean
    open var open: String
    open var height: String
    open var referrerPolicy: String
    open val contentDocument: Document?
    open val contentWindow: Window?
    open var align: String
    open var open: String
    open var frameBorder: String
    open var longDesc: String
    open var marginHeight: String
    open var marginWidth: String
    fun getSVGDocument(): Document?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript
[HTMLEmbedElement](https://developer.mozilla.org/en/docs/Web/API/HTMLEmbedElement) to Kotlin
*/
public external abstract class HTMLEmbedElement : HTMLElement {
    open var src: String
    open var type: String
    open var width: String
    open var height: String
    open var align: String
    open var name: String
    fun getSVGDocument(): Document?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript
[HTMLObjectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLObjectElement) to Kotlin
*/
public external abstract class HTMLObjectElement : HTMLElement {
    open var data: String
    open var type: String
    open var typeMustMatch: Boolean
    open var name: String
    open var useMap: String
    open val form: HTMLFormElement?
    open var width: String
    open var height: String
    open val contentDocument: Document?
    open val contentWindow: Window?
    open var willValidate: Boolean
    open var validity: ValidityState
    open var validationMessage: String
    open var align: String
    open var archive: String
    open var code: String
    open var declare: Boolean
    open var hspace: Int
    open var standby: String
    open var vspage: String
    open var codeBase: String
    open var codeType: String
    open var border: String
    fun getSVGDocument(): Document?
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
**Exposes the JavaScript [HTMLParamElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParamElement) to Kotlin

```kotlin
public external abstract class HTMLParamElement : HTMLElement {
    open var name: String
    open var value: String
    open var type: String
    open var valueType: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

**Exposes the JavaScript [HTMLVideoElement](https://developer.mozilla.org/en/docs/Web/API/HTMLVideoElement) to Kotlin

```kotlin
public external abstract class HTMLVideoElement : HTMLMediaElement, CanvasImageSource, TexImageSource {
    open var width: Int
    open var height: Int
    open val videoWidth: Int
    open val videoHeight: Int
    open var poster: String
    open var playsInline: Boolean

    companion object {
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
    }
}
```

**Exposes the JavaScript [HTMLAudioElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAudioElement) to Kotlin

```kotlin
public external abstract class HTMLAudioElement : HTMLMediaElement {

    companion object {
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
    }
}
```
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLTrackElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTrackElement) to Kotlin
 */
public external abstract class HTMLTrackElement : HTMLElement {
    open var kind: String
    open var src: String
    open var srclang: String
    open var label: String
    open var default: Boolean
    open val readyState: Short

    companion object {
        val NONE: Short
        val LOADING: Short
        val LOADED: Short
        val ERROR: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLMediaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement) to Kotlin
 */
public external abstract class HTMLMediaElement : HTMLElement {
    open var src: String
    open var srcObject: MediaProvider?
    open val currentSrc: String
    open var crossOrigin: String?
    open val networkState: Short
    open var preload: String
    open var buffered: TimeRanges
    open val readyState: Short
    open val seeking: Boolean
    open var currentTime: Double
    open val duration: Double
    open var paused: Boolean
    open var defaultPlaybackRate: Double
    open var playbackRate: Double
    open var autoplay: Boolean
    open var loop: Boolean
    open var controls: Boolean
    open var volume: Double
    open var muted: Boolean
    open var defaultMuted: Boolean
    open var audioTracks: AudioTrackList
    open var videoTracks: VideoTrackList
    open var textTracks: TextTrackList
    open var mediaKeys: MediaKeys?
    open var onencrypted: ((Event) -> dynamic)?
    open var onwaitingforkey: ((Event) -> dynamic)?

    fun load()
    fun canPlayType(type: String): CanPlayTypeResult
    fun fastSeek(time: Double)
    fun getStartDate(): dynamic
    fun play(): Promise<Unit>
    fun pause()
    fun addTextTrack(kind: TextTrackKind, label: String = definedExternally, language: String = definedExternally): TextTrack
    fun setMediaKeys(mediaKeys: MediaKeys?): Promise<Unit>

    companion object {
        val NETWORK_EMPTY: Short
        val NETWORK_IDLE: Short
        val NETWORK_LOADING: Short
        val NETWORK_NO_SOURCE: Short
        val HAVE_NOTHING: Short
        val HAVE_METADATA: Short
        val HAVE_CURRENT_DATA: Short
        val HAVE_FUTURE_DATA: Short
        val HAVE_ENOUGH_DATA: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [MediaError](https://developer.mozilla.org/en/docs/Web/API/MediaError) to Kotlin
 */
public external abstract class MediaError {
    open var code: Short

    companion object {
        val MEDIA_ERR_ABORTED: Short
        val MEDIA_ERR_NETWORK: Short
        val MEDIA_ERR_DECODE: Short
        val MEDIA_ERR_SRC_NOT_SUPPORTED: Short
    }
}

/**
 * Exposes the JavaScript [AudioTrackList](https://developer.mozilla.org/en/docs/Web/API/AudioTrackList) to Kotlin
 */
public external abstract class AudioTrackList : EventTarget {
    open val length: Int
    open var onchange: ((Event) -> dynamic)?

    companion object {
        val MEDIA_ERR_ABORTED: Short
        val MEDIA_ERR_NETWORK: Short
        val MEDIA_ERR_Decode: Short
        val MEDIA_ERR_SRC_NOT_SUPPORTED: Short
    }
}
Open Source Used In DNAC Platform Antelope 14116

```kotlin
fun getTrackById(id: String): AudioTrack?

@kotlin.internal.InlineOnly
public inline operator fun AudioTrackList.get(index: Int): AudioTrack? = asDynamic()[index]

/**
 * Exposes the JavaScript [AudioTrack](https://developer.mozilla.org/en/docs/Web/API/AudioTrack) to Kotlin */

public external abstract class AudioTrack : UnionAudioTrackOrTextTrackOrVideoTrack {
    open val id: String
    open val kind: String
    open val label: String
    open val language: String
    open var enabled: Boolean
    open val sourceBuffer: SourceBuffer?
}

/**
 * Exposes the JavaScript [VideoTrackList](https://developer.mozilla.org/en/docs/Web/API/VideoTrackList) to Kotlin */

public external abstract class VideoTrackList : EventTarget {
    open val length: Int
    open var onchange: ((Event) -> dynamic)?
    open var onaddtrack: ((TrackEvent) -> dynamic)?
    open var onremovetrack: ((TrackEvent) -> dynamic)?
    fun getTrackById(id: String): VideoTrack?
}

@kotlin.internal.InlineOnly
public inline operator fun VideoTrackList.get(index: Int): VideoTrack? = asDynamic()[index]

/**
 * Exposes the JavaScript [VideoTrack](https://developer.mozilla.org/en/docs/Web/API/VideoTrack) to Kotlin */

public external abstract class VideoTrack : UnionAudioTrackOrTextTrackOrVideoTrack {
    open val id: String
    open val kind: String
    open val label: String
    open val language: String
    open var selected: Boolean
    open val sourceBuffer: SourceBuffer?
}

public external abstract class TextTrackList : EventTarget {
    open val length: Int
    open var onchange: ((Event) -> dynamic)?
    open var onaddtrack: ((TrackEvent) -> dynamic)?
    open var onremovetrack: ((TrackEvent) -> dynamic)?
    fun getTrackById(id: String): TextTrack?
}

@kotlin.internal.InlineOnly
public inline operator fun TextTrackList.get(index: Int): TextTrack? = asDynamic()[index]

/**
 * Exposes the JavaScript [TextTrack](https://developer.mozilla.org/en/docs/Web/API/TextTrack) to Kotlin */

public external abstract class TextTrack : EventTarget, UnionAudioTrackOrTextTrackOrVideoTrack {
    open val kind: TextTrackKind
    open val label: String
    open val language: String
    open val id: String
    open val inBandMetadataTrackDispatchType: String
    open var mode: TextTrackMode
    open val cues: TextTrackCueList?
    open val activeCues: TextTrackCueList?
    open var oncuechange: ((Event) -> dynamic)?
    open val sourceBuffer: SourceBuffer?
    fun addCue(cue: TextTrackCue)
    fun removeCue(cue: TextTrackCue)
}

public external abstract class TextTrackCueList {
    open val length: Int
    fun getCueById(id: String): TextTrackCue?
}

@kotlin.internal.InlineOnly
public inline operator fun TextTrackCueList.get(index: Int): TextTrackCue? = asDynamic()[index]

/**
 * Exposes the JavaScript [TextTrackCue](https://developer.mozilla.org/en/docs/Web/API/TextTrackCue) to Kotlin */

public external abstract class TextTrackCue : EventTarget {
    open val track: TextTrack?
    open var id: String
    open var startTime: Double
    open var endTime: Double
    open var pauseOnExit: Boolean
    open var onenter: ((Event) -> dynamic)?
    open var onexit: ((Event) -> dynamic)?
}

/**
 * Exposes the JavaScript [TimeRanges](https://developer.mozilla.org/en/docs/Web/API/TimeRanges) to Kotlin */

public external abstract class TimeRanges {
    open val length: Int
    fun start(index: Int): Double
    fun end(index: Int): Double
}

/**
 * Exposes the JavaScript [TrackEvent](https://developer.mozilla.org/en/docs/Web/API/TrackEvent) to Kotlin */

public external open class TrackEvent(type: String, eventInitDict: TrackEventInit = definedExternally) : Event {
    open val track: UnionAudioTrackOrTextTrackOrVideoTrack?

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface TrackEventInit : EventInit {
    var track: UnionAudioTrackOrTextTrackOrVideoTrack? = null
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun TrackEventInit(track: UnionAudioTrackOrTextTrackOrVideoTrack? = null, bubbles: Boolean = false, cancelable: Boolean = false, composed: Boolean? = false) { TrackEventInit("

@kotlin.internal.InlineOnly
public inline fun TextTrackCueList.get(index: Int): TextTrackCue? = asDynamic()
public external abstract class HTMLMapElement : HTMLElement {
    open var name: String
    open var areas: HTMLCollection

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAreaElement) to Kotlin *
 */
public external abstract class HTMLAreaElement : HTMLElement, HTMLHyperlinkElementUtils {
    open var alt: String
    open var coords: String
    open var shape: String
    open var target: String
    open var download: String
    open var ping: String
    open var rel: String
    open var referrerPolicy: String
    open var noHref: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTableElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableElement) to Kotlin *
 */
public external abstract class HTMLTableElement : HTMLElement {
    open var caption: HTMLTableCaptionElement?
    open var tHead: HTMLTableSectionElement?
    open var tFoot: HTMLTableSectionElement?
    open var tBodies: HTMLCollection
    open var tFoot: HTMLTableSectionElement?
    open var tBodies: HTMLCollection
    open var rows: HTMLCollection
    open var tBodies: HTMLCollection
    open var align: String
    open var border: String
    open var frame: String
    open var rules: String
    open var summary: String
    open var width: String
    open var bgColor: String
    open var cellPadding: String
    open var cellSpacing: String
    fun createCaption(): HTMLTableCaptionElement
    fun deleteCaption(): HTMLTableCaptionElement
    fun createTHead(): HTMLTableSectionElement
    fun deleteTHead(): HTMLTableSectionElement
    fun createTBody(): HTMLTableSectionElement
    fun deleteTBody(): HTMLTableSectionElement
    fun insertRow(index: Int = definedExternally): HTMLTableRowElement
    fun deleteRow(index: Int)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTableCaptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCaptionElement) to Kotlin *
 */
public external abstract class HTMLTableCaptionElement : HTMLElement {
    open var align: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
/* Exposes the JavaScript [HTMLTableColElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableColElement) to Kotlin

```kotlin
public external abstract class HTMLTableColElement : HTMLElement {
    open var span: Int
    open var align: String
    open var ch: String
    open var chOff: String
    open var vAlign: String
    open var width: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTableSectionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableSectionElement) to Kotlin
 */
public external abstract class HTMLTableSectionElement : HTMLElement {
    open var rows: HTMLCollection
    open var align: String
    open var ch: String
    open var chOff: String
    open var vAlign: String
    fun insertRow(index: Int = definedExternally): HTMLElement
    fun deleteRow(index: Int)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTableRowElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableRowElement) to Kotlin
 */
public external abstract class HTMLTableRowElement : HTMLElement {
    open var rowIndex: Int
    open var sectionRowIndex: Int
    open var cells: HTMLCollection
    open var align: String
    open var ch: String
    open var chOff: String
    open var vAlign: String
    open var bgColor: String
    fun insertCell(index: Int = definedExternally): HTMLElement
    fun deleteCell(index: Int)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLTableCellElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCellElement) to Kotlin
 */
public external abstract class HTMLTableCellElement : HTMLElement {
    open var colSpan: Int
    open var
String
open var formMethod: String
open var formNoValidate: Boolean
open var formTarget: String
open var height: Int
open var indeterminate: Boolean
open var inputMode: String
open val list: HTMLInputElement?
open var max: String
open var maxLength: Int
open var min: String
open var minLength: Int
open var multiple: Boolean
open var name: String
open var pattern: String
open var placeholder: String
open var readOnly: Boolean
open var required: Boolean
open var size: Int
open var src: String
open var step: String
open var type: String
open var defaultValue: String
open var value: String
open var valueAsDate: dynamic
open var valueAsNumber: Double
open var width: Int
open val willValidate: Boolean
open val validity: ValidityState
open val validationMessage: String
open val labels: NodeList
open var selectionStart: Int?
open var selectionEnd: Int?
open var selectionDirection: String?
open var align: String
open var useMap: String
fun stepUp(n: Int = definedExternally)
fun stepDown(n: Int = definedExternally)
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)
fun select()
fun setRangeText(replacement: String)
fun setRangeText(replacement: String, start: Int, end: Int, selectionMode: SelectionMode = definedExternally)
fun setSelectionRange(start: Int, end: Int, direction: String = definedExternally)

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript HTMLButtonElement to Kotlin
 */
public external abstract class HTMLButtonElement : HTMLElement {
open var autofocus: Boolean
open var disabled: Boolean
open val form: HTMLFormElement?
open var formAction: String
open var formEnctype: String
open var formMethod: String
open var formNoValidate: Boolean
open var formTarget: String
open var name: String
open var type: String
open var value: String
open var menu: HTMLMenuElement?
open val willValidate: Boolean
open val validity: ValidityState
open val validationMessage: String
open val labels: NodeList
fun checkValidity(): Boolean
fun reportValidity(): Boolean
fun setCustomValidity(error: String)

companion object {
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITYREFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSINGINSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITIONContains: Short
val DOCUMENT_POSITIONCONTAINEDBY: Short
val DOCUMENTPOSITIONIMPLEMENTATIONSPECIFIC: Short
}

/**
 * Exposes the JavaScript HTMLSelectElement to Kotlin
 */
public external abstract class HTMLSelectElement : HTMLElement, ItemArrayLike<Element> {
open var autocomplete: String
open var autofocus: Boolean
open var disabled: Boolean
open val form: HTMLFormElement?
open var multiple: Boolean
open var name: String
open var required: Boolean
open var size: Int
open var type: String
open val options: HTMLOptionsCollection
override var length: Int
open val selectedOptions: HTMLCollection
open var selectedIndex: Int
open var value: String
open val willValidate: Boolean
open val validity: ValidityState
open val validationMessage: String
open val
OPEN SOURCE USED IN DNAC PLATFORM ANTELOPE 14122

DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLTextAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTextAreaElement) to Kotlin
 */
public external abstract class HTMLTextAreaElement : HTMLElement {
    open var autocomplete: String
    open var autofocus: Boolean
    open var cols: Int
    open var dirName: String
    open var disabled: Boolean
    open val form: HTMLFormElement?
    open var inputMode: String
    open var maxLength: Int
    open var minLength: Int
    open var name: String
    open var placeholder: String
    open var readOnly: Boolean
    open var required: Boolean
    open var rows: Int
    open var wrap: String
    open val type: String
    open var defaultValue: String
    open var value: String
    open val textLength: Int
    open val willValidate: Boolean

    open val validity: ValidityState
    open val validationMessage: String
    open val labels: NodeList
    open var selectionStart: Int?
    open var selectionEnd: Int?
    open var selectionDirection: String?
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLKeygenElement](https://developer.mozilla.org/en/docs/Web/API/HTMLKeygenElement) to Kotlin
 */
public external abstract class HTMLKeygenElement : HTMLElement {
    open var autofocus: Boolean
    open var challenge: String
    open var disabled: Boolean
    open val form: HTMLFormElement?
    open var keytype: String
    open var name: String
    open val type: String
    open val willValidate: Boolean
    open val validity: ValidityState
    open val validationMessage: String
    open val labels: NodeList
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLOutputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOutputElement) to Kotlin
 */
public external abstract class HTMLOutputElement : HTMLElement {
    open val htmlFor: DOMTokenList
    open val form: HTMLFormElement?
    open var name: String
    open val type: String
    open var defaultValue: String
    open var value: String
    open val willValidate: Boolean
    open val validity: ValidityState
    open val validationMessage: String
    open val labels: NodeList
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {

val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLProgressElement](https://developer.mozilla.org/en/docs/Web/API/HTMLProgressElement) to Kotlin

/**
 * Exposes the JavaScript [HTMLMeterElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMeterElement) to Kotlin
 */
public external abstract class HTMLMeterElement : HTMLElement {
    open var value: Double
    open var min: Double
    open var max: Double
    open var low: Double
    open var high: Double
    open var optimum: Double
    open val labels: NodeList

    companion object {

val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLFieldSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFieldSetElement) to Kotlin

/**
 * Exposes the JavaScript [HTMLFieldSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFieldSetElement) to Kotlin
 */
public external abstract class HTMLFieldSetElement : HTMLElement {
    open var disabled: Boolean
    open val form: HTMLFormElement?
    open var name: String
    open val type: String
    open val elements: HTMLCollection
    open val willValidate: Boolean
    open val validity: ValidityState
    open val validationMessage: String
    fun checkValidity(): Boolean
    fun reportValidity(): Boolean
    fun setCustomValidity(error: String)

    companion object {

val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

* Exposes the JavaScript [HTMLFormElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFormElement) to Kotlin

// End of class definitions
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/Open Source Used In DNAC Platform Antelope 14124

/**
 * Exposes the JavaScript [HTMLLegendElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLegendElement) to Kotlin
 */
public external abstract class HTMLLegendElement : HTMLElement {
    open val form: HTMLFormElement?
    open var align: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }

    /**
     * Exposes the JavaScript [ValidityState](https://developer.mozilla.org/en/docs/Web/API/ValidityState) to Kotlin
     */
    public external abstract class ValidityState {
        open val valueMissing: Boolean
        open val typeMismatch: Boolean
        open val patternMismatch: Boolean
        open val tooLong: Boolean
        open val tooShort: Boolean
        open val rangeUnderflow: Boolean
        open val rangeOverflow: Boolean
        open val stepMismatch: Boolean
        open val badInput: Boolean
        open val customError: Boolean
        open val valid: Boolean
    }

    /**
     * Exposes the JavaScript [HTMLDetailsElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDetailsElement) to Kotlin
     */
    public external abstract class HTMLDetailsElement : HTMLElement {
        open var open: Boolean

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
        }
    }

    public external abstract class HTMLMenuElement : HTMLElement {
        open var type: String
        open var label: String
        open var compact: Boolean

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
        }
    }

    public external abstract class HTMLMenuItemElement : HTMLElement {
        open var type: String
        open var label: String
        open var icon: String
        open var disabled: Boolean
        open var checked: Boolean
        open var radiogroup: String
        open var default: Boolean

        companion object {
            val ELEMENT_NODE: Short
            val ATTRIBUTE_NODE: Short
            val TEXT_NODE: Short
            val CDATA_SECTION_NODE: Short
            val ENTITY_REFERENCE_NODE: Short
            val ENTITY_NODE: Short
            val PROCESSING_INSTRUCTION_NODE: Short
            val COMMENT_NODE: Short
            val DOCUMENT_NODE: Short
            val DOCUMENT_TYPE_NODE: Short
            val DOCUMENT_FRAGMENT_NODE: Short
            val DOCUMENT_POSITION_DISCONNECTED: Short
            val DOCUMENT_POSITION_PRECEDING: Short
            val DOCUMENT_POSITION_FOLLOWING: Short
            val DOCUMENT_POSITION_CONTAINS: Short
            val DOCUMENT_POSITION_CONTAINED_BY: Short
        }
    }
}
DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short

val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short

val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [HTMLSlotElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSlotElement) to Kotlin
 */
public external abstract class HTMLSlotElement : HTMLElement {
    open var name: String
    fun assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>

    companion object {

        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface AssignedNodesOptions {
    var flatten: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun AssignedNodesOptions(flatten: Boolean? = false): AssignedNodesOptions {
    val o = js("({})")
    o["flatten"] = flatten
    return o
}

/**
 * Exposes the JavaScript [HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement) to Kotlin
 */
public external abstract class HTMLCanvasElement : HTMLElement, CanvasImageSource, TexImageSource {
    open var width: Int
    open var height: Int
    fun getContext(contextId: String, vararg arguments: Any?): RenderingContext?
    fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally): String
    fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? = definedExternally)

    companion object {

        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface CanvasRenderingContext2DSettings {
    var alpha: Boolean? /* = true */
    get() = definedExternally
    set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {
    val o = js("({})")
    o["alpha"] = alpha
    return o
}

/**
 * Exposes the JavaScript [CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D) to Kotlin
 */
public external abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform, CanvasCompositing, CanvasImageSmoothing, CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters, CanvasRect, CanvasDrawPath, CanvasUserInterface, CanvasText, CanvasDrawImage, CanvasHitRegion, CanvasImageData, CanvasPathDrawingStyles, CanvasTextDrawingStyles, CanvasPath, RenderingContext {

    open val canvas: HTMLCanvasElement
}

public external interface CanvasState {
    fun save()
    fun restore()
}

public external interface CanvasTransform {
    fun scale(x: Double, y: Double)
    fun rotate(angle: Double)
    fun translate(x: Double, y: Double)
    fun transform(a: Double, b: Double, c: Double, d: Double, e: Double, f: Double)
    fun getTransform(): DOMMatrix
}

Open Source Used In DNAC Platform Antelope 14126
JavaScript [CanvasPattern](https://developer.mozilla.org/en/docs/Web/API/CanvasPattern) to Kotlin

```kotlin
fun setTransform(transform: dynamic = definedExternally): Unit
```

Exposed by the JavaScript [TextMetrics](https://developer.mozilla.org/en/docs/Web/API/TextMetrics) to Kotlin

```kotlin
external abstract class TextMetrics {
    open val width: Double
    open val actualBoundingBoxLeft: Double
    open val fontBoundingBoxAscent: Double
    open val fontBoundingBoxDescent: Double
    open val actualBoundingBoxAscent: Double
    open val actualBoundingBoxDescent: Double
    open val emHeightAscent: Double
    open val emHeightDescent: Double
    open val hangingBaseline: Double
    open val alphabeticBaseline: Double
    open val ideographicBaseline: Double
}
```

Exposed by the JavaScript [ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to Kotlin

```kotlin
public external open class ImageData : ImageBitmapSource, TexImageSource {
    constructor(sw: Int, sh: Int)
    constructor(data: Uint8ClampedArray, sw: Int, sh: Int = definedExternally)
    open val width: Int
    open val height: Int
    open val data: Uint8ClampedArray
}
```

Exposed by the JavaScript [CanvasPath](https://developer.mozilla.org/en/docs/Web/API/CanvasPath) to Kotlin

```kotlin
fun closePath()
fun moveTo(x: Double, y: Double)
fun lineTo(x: Double, y: Double)
fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)
fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)
fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)
fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)
fun rect(x: Double, y: Double, width: Double, height: Double)
fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean = definedExternally) {
```
whenDefined(name: String): Promise<Unit>

public external interface ElementDefinitionOptions {
    var extends: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ElementDefinitionOptions(extends: String? = undefined): ElementDefinitionOptions {
    val o = js("{}")
    o["extends"] = extends
    return o
}

public external interface ElementContentEditable {
    var isContentEditable: Boolean
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline operator fun ElementDefinitionOptions.get(index: Int): ElementDefinitionOptions

/**
 * Exposes the JavaScript
 [DataTransfer](https://developer.mozilla.org/en/docs/Web/API/DataTransfer) to Kotlin
 */
public external class DataTransfer {
    open var dropEffect: String?
    open var effectAllowed: String
    open val items: DataTransferItemList
    open val types: Array<out String>
    open val files: FileList
    fun setDragImage(image: Element, x: Int, y: Int): Unit
    fun getData(format: String): String
    fun setData(format: String, data: String): Unit
    fun clearData(format: String = definedExternally): Unit
}

/**
 * Exposes the JavaScript
 [DataTransferItemList](https://developer.mozilla.org/en/docs/Web/API/DataTransferItemList) to Kotlin
 */
public external abstract class DataTransferItemList {
    open val length: Int
    fun add(data: String, type: String): DataTransferItem?
    fun add(file: File): DataTransferItem?
    fun remove(index: Int)
    fun clear()
}

@kotlin.internal.InlineOnly
public inline operator fun DataTransferItemList.get(index: Int): DataTransferItem? = asDynamic()[index]

/**
 * Exposes the JavaScript
 [DataTransferItem](https://developer.mozilla.org/en/docs/Web/API/DataTransferItem) to Kotlin
 */
public external abstract class DataTransferItem {
    open var kind: String
    open var type: String
    fun getAsString(_callback: ((String) -> Unit)?)
    fun getAsFile(): File?
}

/**
 * Exposes the JavaScript
 [DragEvent](https://developer.mozilla.org/en/docs/Web/API/DragEvent) to Kotlin
 */
public external open class DragEvent(type: String, eventInitDict: DragEventInit = definedExternally) : MouseEvent {
    open val dataTransfer: DataTransfer?
}

public external interface DragEventInit : MouseEventInit {
    var dataTransfer: DataTransfer? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline operator fun DragEventInit.get(): DragEventInit

/**
 * Exposes the JavaScript
 [Window](https://developer.mozilla.org/en/docs/Web/API/Window) to Kotlin
 */
public external abstract class Window : EventTarget, GlobalEventHandlers, WindowEventHandlers, WindowOrWorkerGlobalScope, WindowSessionStorage, WindowLocalStorage, GlobalPerformance, UnionMessagePortOrWindowProxy {
    open var name: String
    open val location: Location
    open val history: History
    open val customElements: CustomElementRegistry
    open val locationbar: BarProp
    open val menubar: BarProp

BarProp
open val personalbar: BarProp
open val scrollbars: BarProp
open val toolbar: BarProp
open var status: String
open var closed: Boolean
open val frames: Window
open val length: Int
open val top: Window
open var opener: Any?
open val parent: Window
open val frameElement: Element
open val navigator: Navigator
open val applicationCache: ApplicationCache
open external: Extern
open val screen: Screen
open val innerWidth: Int
open val innerHeight: Int
open val scrollX: Double
open val scrollY: Double
open val scrollRestoration: ScrollRestoration
open val state: Any?
fund go(delta: Int = definedExternally)
fund back()
fund forward()
fund pushState(data: Any?, title: String, url: String? = definedExternally)
fund replaceState(data: Any?, title: String, url: String? = definedExternally)
fund assign(url: String)
fund replace(url: String)
fund reload()
Boolean? = false, composed: Boolean? = false): HashChangeEventInit {
    o = js("({})")
    o["oldURL"] = oldURL
    o["newURL"] = newURL
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript PageTransitionEvent to Kotlin
 */
public open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) : Event {
    val persisted: Boolean
    val bubbles: Boolean
    val cancelable: Boolean
    val composed: Boolean

    get() = definedExternally
    set(value) = definedExternally

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short

            return o
        }
    }
}

public open interface PageTransitionEventInit : EventInit {
    var persisted: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var bubbles: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var cancelable: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var composed: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript BeforeUnloadEvent to Kotlin
 */
public open class BeforeUnloadEvent : Event {
    val returnValue: String

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public abstract class ApplicationCache : EventTarget {
    val status: Short
    var onchecking: ((Event) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    var onnoupdate: ((Event) -> dynamic)?
    var ondownloading: ((Event) -> dynamic)?
    var onprogress: ((ProgressEvent) -> dynamic)?
    var onupdateready: ((Event) -> dynamic)?
    var oncached: ((Event) -> dynamic)?
    var onobsolete: ((Event) -> dynamic)?

    fun update()
    fun abort()
    fun swapCache()

    companion object {
        val UNCACHED: Short
        val IDLE: Short
        val CHECKING: Short
        val DOWNLOADING: Short
        val UPDATEREADY: Short
        val OBSOLETE: Short
    }
}

/**
 * Exposes the JavaScript NavigatorOnLine to Kotlin
 */
public interface NavigatorOnLine {
    val onLine: Boolean
}

/**
 * Exposes the JavaScript ErrorEvent to Kotlin
 */
public open class ErrorEvent(type: String, eventInitDict: ErrorEventInit = definedExternally) : Event {
    val message: String
    val filename: String
    val lineno: Int
    val colno: Int
    val error: Any

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public interface ErrorEventInit : EventInit {
    var message: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var filename: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var lineno: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var colno: Int? /* = 0 */
        get() = definedExternally
        set(value) = definedExternally
    var error: Any? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["message"] = message
    o["filename"] = filename
    o["lineno"] = lineno
    o["colno"] = colno
    o["error"] = error
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript PromiseRejectionEvent to Kotlin
 */
public open class PromiseRejectionEvent(type: String, eventInitDict: PromiseRejectionEventInit) : Event {
    val promise: Promise<Any>
    val reason: Any

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public interface PromiseRejectionEventInit : EventInit {
    var promise: Promise<Any>?
        get() = definedExternally
        set(value) = definedExternally
    var reason: Any
        get() = definedExternally
        set(value) = definedExternally
}
definedExternally
set(value) = definedExternally
var onfocus: ((FocusEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var oninput: ((InputEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onkeydown: ((KeyboardEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onkeypress: ((KeyboardEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onkeyup: ((KeyboardEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onload: ((Event) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onloadeddata: ((Event) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onloadedmetadata: ((Event) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onmessage: ((MessageEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onmousemove: ((MouseEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onmouseout: ((MouseEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onmouseover: ((MouseEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onmouseup: ((MouseEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally
var onun commodo return on\n}];
*/
* Exposes the JavaScript
[GlobalEventHandlers](https://developer.mozilla.org/en/docs/Web/API/GlobalEventHandlers) to Kotlin
*/

```kotlin
public external interface GlobalEventHandlers {
    var onabort: ((Event) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onblur: ((FocusEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncancel: ((Event) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncancelpromise: Promise<any?>?
        get() = undefined
        set(value) = undefined
    var onerror: ((PromiseRejectionEventInit) -> dynamic)?
        get() = undefined
        set(value) = undefined
    var onfocus: ((FocusEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oninput: ((InputEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onkeydown: ((KeyboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onkeypress: ((KeyboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onkeyup: ((KeyboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onmessage: ((MessageEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onmessageerror: ((MessageEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onmousemove: ((MouseEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onmouseout: ((MouseEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onmouseover: ((MouseEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onmouseup: ((MouseEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onun commodo return on\n}];
*/
* Exposes the JavaScript
[GlobalEventHandlers](https://developer.mozilla.org/en/docs/Web/API/GlobalEventHandlers) to Kotlin
*/

```
definedExternally
var onmousemove: (MouseEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onwheel: (WheelEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onplay: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onplaying: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onprogress: (ProgressEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onratechange: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onreset: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onresize: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onscroll: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onseeked: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onseeking: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onselect: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onshow: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onsuspend: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var ontimeupdate: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var ontoggle: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onvolumechange: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onwaiting: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var ongotpointercapture: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onlostpointercapture: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpointerdown: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpointermove: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpointerup: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpointercancel: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpointerover: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpointerout: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpointerenter: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpointerleave: (PointerEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onafterprint: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onbeforeprint: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onbeforeunload: (BeforeUnloadEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onhashchange: (HashChangeEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onlanguagechange: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onmessage: (MessageEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onoffline: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var ononline: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpopstate: (PopStateEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onbeforeunload: (BeforeUnloadEvent) -> String?
get() = definedExternally
set(value) = definedExternally
var onhashchange: (HashChangeEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onlanguagechange: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpopupstate: (PopStateEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onmessage: (MessageEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onoffline: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var ononline: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onpopstate: (PopStateEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onmessage: (MessageEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onhashchange: (HashChangeEvent) -> dynamic?
get() = definedExternally
set(value) = definedExternally
var onlanguagechange: (Event) -> dynamic?
get() = definedExternally
set(value) = definedExternally

var onunhandledrejection:
((PromiseRejectionEvent) -> dynamic)?
get() = definedExternally
set(value) = definedExternally

var onunload: ((Event) -> dynamic)?
get() = definedExternally
set(value) = definedExternally

public external interface DocumentAndElementEventHandlers {
    var oncopy: ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var oncut: ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    var onpaste: ((ClipboardEvent) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
}

/**
 * Exposes the JavaScript
 */
public external interface WindowOrWorkerGlobalScope {
    val origin: String
    val caches: CacheStorage
    fun btoa(data: String): String
    fun atob(data: String): String
    fun setTimeout(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int
    fun clearTimeout(handle: Int = definedExternally)
    fun setInterval(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int
    fun createImageBitmap(image: ImageBitmapSource, options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>
    fun fetch(input: dynamic, init: RequestInit = definedExternally): Promise<Response>
}

/**
 * Exposes the JavaScript
 */
public external abstract class Navigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorContentUtils, NavigatorCookies, NavigatorPlugins, NavigatorConcurrentHardware {
    open val clipboard: Clipboard
    open val mediaDevices: MediaDevices
    open val maxTouchPoints: Int
    open val serviceWorker: ServiceWorkerContainer
    fun requestMediaKeySystemAccess(keySystem: String, supportedConfigurations: Array<MediaKeySystemConfiguration>): Promise<MediaKeySystemAccess>
    fun getUserMedia(constraints: MediaStreamConstraints, successCallback: (MediaStream) -> Unit, errorCallback: (dynamic) -> Unit)
    fun vibrate(pattern: dynamic): Boolean
}

/**
 * Exposes the JavaScript
 */
public external interface NavigatorID {
    val appCodeName: String
    val appName: String
    val appVersion: String
    val platform: String
    val vendor: String
    val vendorSub: String
    val userAgent: String
    val vendor: String
    val vendorSub: String
    fun taintEnabled(): Boolean
}

/**
 * Exposes the JavaScript
 */
public external interface NavigatorLanguage {
    val language: String
    val languages: Array<out String>
}

public external interface NavigatorContentUtils {
    fun registerProtocolHandler(scheme: String, url: String, title: String)
    fun registerContentHandler(mimeType: String, url: String, title: String)
    fun isProtocolHandlerRegistered(scheme: String, url: String): String
    fun isContentHandlerRegistered(mimeType: String, url: String): String
    fun unregisterProtocolHandler(scheme: String, url: String)
    fun unregisterContentHandler(mimeType: String, url: String)
}

public external interface NavigatorCookies {
    val cookieEnabled: Boolean
}

/**
 * Exposes the JavaScript
 */
public external abstract class PluginArray : ItemArrayLike<Plugin> {
    fun refresh(reload: Boolean = definedExternally)
    override fun item(index: Int): Plugin?
    fun namedItem(name: String): Plugin?
}

@kotlin.internal.InlineOnly
public inline operator fun PluginArray.get(index: Int): Plugin? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun PluginArray.get(name: String): Plugin? = asDynamic()[name]

/**
 * Exposes the JavaScript
 */
public external class PluginArray {
    fun refresh(reload: Boolean = definedExternally)
    override fun item(index: Int): Plugin?
    fun namedItem(name: String): Plugin?
    fun get(name: String): Plugin?
}

// Other JavaScript exposes...
abstract class MimeTypeArray : ItemArrayLike<MimeType> {
    override fun item(index: Int): MimeType?
    fun namedItem(name: String): MimeType?
}

@kotlin.internal.InlineOnly
public inline operator fun MimeTypeArray.get(index: Int): MimeType? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun MimeTypeArray.get(name: String): MimeType? = asDynamic()[name]

/**
 * Exposes the JavaScript [Plugin](https://developer.mozilla.org/en/docs/Web/API/Plugin) to Kotlin
 */
public external abstract class Plugin : ItemArrayLike<MimeType> {
    open val name: String
    open val description: String
    open val filename: String
    override fun item(index: Int): MimeType?
    fun namedItem(name: String): MimeType?
}

@kotlin.internal.InlineOnly
public inline operator fun Plugin.get(index: Int): MimeType? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun Plugin.get(name: String): MimeType? = asDynamic()[name]

/**
 * Exposes the JavaScript [MimeType](https://developer.mozilla.org/en/docs/Web/API/MimeType) to Kotlin
 */
public external abstract class MimeType {
    open val type: String
    open val description: String
    open val suffixes: String
    open val enabledPlugin: Plugin
}

/**
 * Exposes the JavaScript [ImageBitmap](https://developer.mozilla.org/en/docs/Web/API/ImageBitmap) to Kotlin
 */
public external abstract class ImageBitmap : CanvasImageSource, TexImageSource {
    open val width: Int
    open val height: Int
    fun close()
}

public external interface ImageBitmapOptions {
    var imageOrientation: ImageOrientation? /* = ImageOrientation.NONE */
            get() = definedExternally
            set(value) = definedExternally
    var premultiplyAlpha: PremultiplyAlpha? /* = PremultiplyAlpha.DEFAUL */
            get() = definedExternally
            set(value) = definedExternally
    var colorSpaceConversion: ColorSpaceConversion? /* = ColorSpaceConversion.DEFAULT */
            get() = definedExternally
            set(value) = definedExternally
    var resizeWidth: Int?
            get() = definedExternally
            set(value) = definedExternally
    var resizeHeight: Int?
            get() = definedExternally
            set(value) = definedExternally
    var resizeQuality: ResizeQuality? /* = ResizeQuality.LOW */
            get() = definedExternally
            set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["imageOrientation"] = imageOrientation
    o["premultiplyAlpha"] = premultiplyAlpha
    o["colorSpaceConversion"] = colorSpaceConversion
    o["resizeWidth"] = resizeWidth
    o["resizeHeight"] = resizeHeight
    o["resizeQuality"] = resizeQuality
    return o
}

/**
 * Exposes the JavaScript [MessageEvent](https://developer.mozilla.org/en/docs/Web/API/MessageEvent) to Kotlin
 */
public external open class MessageEvent(type: String, eventInitDict: MessageEventInit = definedExternally) : Event {
    open val data: Any?
    open val origin: String
    open val lastEventId: String
    open val ports: Array<MessagePort>
    fun initMessageEvent(type: String, bubbles: Boolean, cancelable: Boolean, data: Any?, origin: String, lastEventId: String, ports: Array<MessagePort>)

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface MessageEventInit : EventInit {
        var data: Any? /* = null */
                get() = definedExternally
                set(value) = definedExternally
        var origin: String? /* = "" */
                get() = definedExternally
                set(value) = definedExternally
        var lastEventId: String? /* = "" */
                get() = definedExternally
                set(value) = definedExternally
        var source: UnionMessagePortOrWindowProxy? /* = null */
                get() = definedExternally
                set(value) = definedExternally
        var ports: Array<MessagePort>? /* = arrayOf() */
                get() = definedExternally
                set(value) = definedExternally
}

@kotlin.internal.InlineOnly
```javascript
function createEventObject(
    o = 
    {
        lastEventId: lastEventId,
        source: source,
        ports: ports,
        bubbles: bubbles,
        cancelable: cancelable,
        composed: composed
    }
) {
    return o
}

/**
 * Exposes the JavaScript [EventSource](https://developer.mozilla.org/en/docs/Web/API/EventSource) to Kotlin
 */
public external open class EventSource(url: String, eventSourceInitDict: EventSourceInit = definedExternally) : EventTarget {
    open val url: String
    open val withCredentials: Boolean
    open val readyState: Short
    var onopen: ((Event) -> dynamic)?
    var onmessage: ((MessageEvent) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    fun close()

    companion object {
        val CONNECTING: Short
        val OPEN: Short
        val CLOSED: Short
    }
}

/**
 * Exposes the JavaScript [WebSocket](https://developer.mozilla.org/en/docs/Web/API/WebSocket) to Kotlin
 */
public external open class WebSocket(url: String, protocols: dynamic = definedExternally) : EventTarget {
    open val url: String
    open val readyState: Short
    open val bufferedAmount: Number
    var onopen: ((Event) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    var onclose: ((Event) -> dynamic)?
    open val extensions: String
    open val protocol: String
    var onmessage: ((MessageEvent) -> dynamic)?
    var binaryType: BinaryType
    fun close(code: Short = definedExternally, reason: String = definedExternally)
    fun send(data: String)
    fun send(data: Blob)
    fun send(data: ArrayBuffer)
    fun send(data: ArrayBufferView)

    companion object {
        val CONNECTING: Short
        val OPEN: Short
        val CLOSING: Short
        val CLOSED: Short
    }
}

/**
 * Exposes the JavaScript [BroadcastChannel](https://developer.mozilla.org/en/docs/Web/API/BroadcastChannel) to Kotlin
 */
public external open class BroadcastChannel(name: String) : EventTarget {
    open val name: String
    var onmessage: ((MessageEvent) -> dynamic)?
    fun postMessage(message: Any?)
    fun close()
}

/**
 * Exposes the JavaScript [WorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WorkerGlobalScope) to Kotlin
 */
public external abstract class WorkerGlobalScope : EventTarget, WindowOrWorkerGlobalScope, GlobalPerformance {
    open val self: WorkerGlobalScope
    open val location: WorkerLocation
    open val navigator: WorkerNavigator
    open var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?
    open var onlanguagechange: ((Event) -> dynamic)?
    open var onoffline: ((Event) -> dynamic)?
    open var onrejectionhandled: ((PromiseRejectionEvent) -> dynamic)?
    fun importScripts(vararg urls: String)
}

```
JavaScript

[DedicatedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/DedicatedWorkerGlobalScope) to Kotlin

```kotlin
*\public external abstract class DedicatedWorkerGlobalScope : WorkerGlobalScope {\n  open var onmessage: ((MessageEvent) -> dynamic)?\n  fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n  fun close()\n} /* Exposes the JavaScript

[SharedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/SharedWorkerGlobalScope) to Kotlin

```kotlin
*\public external abstract class SharedWorkerGlobalScope : WorkerGlobalScope {\n  open var name: String\n  open val applicationCache: ApplicationCache\n  open var onconnect: ((Event) -> dynamic)?\n  override var onerror: ((Event) -> dynamic)?\n  fun close()\n} /* Exposes the JavaScript

[AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to Kotlin

```kotlin
*\public external open class Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {\n  var onmessage: ((MessageEvent) -> dynamic)?\n  override var onerror: ((Event) -> dynamic)?\n  fun terminate()\n  fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n} /* Exposes the JavaScript

[Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin

```kotlin
*\public external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {\n  open val port: MessagePort\n} /* Exposes the JavaScript

[NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to Kotlin

```kotlin
*\public external abstract class NavigatorConcurrentHardware {\n  val hardwareConcurrency: Number\n} /* Exposes the JavaScript

[WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin

```kotlin
*\public external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorConcurrentHardware {\n  open val serviceWorker: ServiceWorkerContainer\n} /* Exposes the JavaScript

[Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to Kotlin

```kotlin
*\public external abstract class Storage {\n  open val length: Int\n  fun key(index: Int): String?\n  fun removeItem(key: String)\n  fun clear()\n  fun getItem(key: String): String?\n  fun setItem(key: String, value: String)\n} /* Exposes the JavaScript

[WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin

```kotlin
*\public external abstract class WindowSessionStorage {\n  val sessionStorage: Storage\n} /* Exposes the JavaScript

[WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage) to Kotlin

```kotlin
*\public external abstract class WindowLocalStorage {\n  val localStorage: Storage\n} /* Exposes the JavaScript

[StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin

```kotlin
*\public external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) : Event {\n```
open val key: String?
open val oldValue: String?
open val newValue: String?
open val url: String?

open val storageArea: Storage?

companion object {
    val NONE: Short
    val CAPTURING_PHASE: Short
    val AT_TARGET: Short
    val BUBBLING_PHASE: Short
}

public external interface StorageEventInit : EventInit {
    var key: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var oldValue: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var newValue: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var url: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
    var storageArea: Storage? /* = null */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun StorageEventInit(key: String? = null,
    val o = js("({})")
    o["key"] = key
    o["oldValue"] = oldValue
    o["newValue"] = newValue
    o["url"] = url
    o["storageArea"] = storageArea
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    return o
}

public external abstract class HTMLAppletElement : HTMLElement {
    open var align: String
    open var alt: String
    open var archive: String
    open var code: String
    open var codeBase: String
    open var height: String
    open var hspace: Int
    open var name: String
    open var object: String
    open var vspace: Int
    open var width: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLMarqueeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMarqueeElement) to Kotlin
 */
public external abstract class HTMLMarqueeElement : HTMLElement {
    open var behavior: String
    open var bgColor: String
    open var direction: String
    open var height: String
    open var hspace: Int
    open var loop: Int
    open var scrollAmount: Int
    open var scrollDelay: Int
    open var trueSpeed: Boolean
    open var vspace: Int
    open var width: String
    open var onbounce: ((Event) -> dynamic)?
    open var onfinish: ((Event) -> dynamic)?
    open var onstart: ((Event) -> dynamic)?
    fun start()
    fun stop()

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLFrameSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFrameSetElement) to Kotlin
 */
public external abstract class HTMLFrameSetElement : HTMLElement, WindowEventHandlers {
    open var cols: String
    open var rows: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

Open Source Used In DNAC Platform Antelope 14138
DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

open external abstract class HTMLFrameElement : HTMLElement {
    open var name: String
    open var scrolling: String
    open var src: String
    open var frameBorder: String
    open var longDesc: String
    open var noResize: Boolean
    open var contentDocument: Document?
    open var contentWindow: Window?
    open var marginHeight: String
    open var marginWidth: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external abstract class HTMLDirectoryElement : HTMLElement {
    open var compact: Boolean

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [HTMLFontElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFontElement) to Kotlin
 */
public external abstract class HTMLFontElement : HTMLElement {
    open var color: String
    open var face: String
    open var size: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface {External
    fun AddSearchProvider()
    fun IsSearchProviderInstalled()
}

public external interface EventInit {
    var bubbles: Boolean? /* = false */
    var cancelable: Boolean? /* = false */
    var composed: Boolean? /* = false */

    @kotlin.internal.InlineOnly
        val o = js("({})")
        o["bubbles"] = bubbles
        o["cancelable"] = cancelable
        o["composed"] = composed
        return o
    }
}

/**
 * Exposes the JavaScript [CustomEvent](https://developer.mozilla.org/en/docs/Web/API/CustomEvent) to Kotlin
 */
public
external open class CustomEvent(type: String, eventInitDict: CustomEventInit = definedExternally): Event {
  open val detail: Any?
  fun initCustomEvent(type: String, bubbles: Boolean, cancelable: Boolean, detail: Any?)
  companion object {
    val NONE: Short
    val CAPTURING_PHASE: Short
    val AT_TARGET: Short
    val BUBBLING_PHASE: Short
  }

  public external interface CustomEventInit : EventInit {
    var detail: Any? /* = null */
    set(value) = definedExternally
  }

  @kotlin.internal.InlineOnly
    val o = js("({})")
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
  }

  public interface AddEventListenerOptions : EventListenerOptions {
    var passive: Boolean? /* = false */
    set(value) = definedExternally
  }

  @kotlin.internal.InlineOnly
    val o = js("({})")
    o["passive"] = passive
    o["once"] = once
    o["capture"] = capture
    return o
  }

  public external interface NonElementParentNode {
    fun getElementsByClassName(className: String): NodeList
    fun getElementsByClassNameAndId(className: String, id: String): Element?
    fun getElementsByClassNameAndId(id: String, className: String): Element?
  }

  public interface ParentNode {
    val children: HTMLCollection
    val firstElementChild: Element?
    val lastElementChild: Element?
    val childElementCount: Int
    fun prepend(vararg nodes: Any)
    fun append(vararg nodes: Any)
    fun querySelector(selectors: String): Element?
    fun querySelectorAll(selectors: String): NodeList
  }

  public interface NonDocumentTypeChildNode {
    val previousElementSibling: Element?
    val nextElementSibling: Element?
  }

  public interface ChildNode {
    fun before(vararg nodes: Any)
    fun after(vararg nodes: Any)
    fun replaceWith(vararg nodes: Any)
    fun remove(): Unit
  }

  public interface Slotable {
    val assignedSlot: HTMLSlotElement?
  }

  public external abstract class NodeList : ItemArrayLike<Node> {
    override fun item(index: Int): Node?
  }

  public interface HTMLCollection : ItemArrayLike<Element>, UnionElementOrHTMLCollection {
    override fun item(index: Int): Element?
  }

  public class MutationObserver(callback: (Array<MutationRecord>, MutationObserver) -> Unit)

  public class MutationObserver { // Optionally, you can implement this class
    fun observe(target: Node, options: MutationObserverInit = definedExternally)
    fun disconnect()
takeRecords(): Array<MutationRecord> * Exposes the JavaScript [MutationObserverInit](https://developer.mozilla.org/en/docs/Web/API/MutationObserverInit) to Kotlin

```kotlin
public external interface MutationObserverInit {
    var childList: Boolean? /* = false */
        get() =
        set(value) = definedExternally
    var attributes: Boolean?
        get() =
        set(value) = definedExternally
    var characterData: Boolean?
        get() =
        set(value) = definedExternally
    var subtree: Boolean? /* = false */
        get() =
        set(value) = definedExternally
    var attributeOldValue: Boolean?
        get() =
        set(value) = definedExternally
    var characterDataOldValue: Boolean?
        get() =
        set(value) = definedExternally
    var attributeFilter: Array<String>?
        get() =
        set(value) = definedExternally
}
```

```kotlin
@kotlin.internal.InlineOnly
    val o = js("({})")
    o["childList"] = childList
    o["attributes"] = attributes
    o["characterData"] = characterData
    o["subtree"] = subtree
    o["attributeOldValue"] = attributeOldValue
    o["characterDataOldValue"] = characterDataOldValue
    o["attributeFilter"] = attributeFilter
    return o
}
```

```kotlin
public external abstract class MutationRecord {
    open val type: String
    open val target: Node
    open val addedNodes: NodeList
    open val removedNodes: NodeList
    open val previousSibling: Node?
    open val nextSibling: Node?
    open val attributeName: String?
    open val attributeNamespace: String?
    open val oldValue: String?
}
```

```kotlin
public external abstract class Node : EventTarget {
    open val nodeType: Short
    open val nodeName: String
    open val baseURI: String
    open val isConnected: Boolean
    open val ownerDocument: Document?
    open val parentNode: Node?
    open val parentElement: Element?
    open val childNodes: NodeList
    open val firstChild: Node?
    open val lastChild: Node?
    open val previousSibling: Node?
    open val nextSibling: Node?
    open var nodeValue: String?
    open var textContent: String?
    fun getRootNode(options: GetRootNodeOptions = definedExternally): Node
    fun normalize(): Node
    fun cloneNode(deep: Boolean = definedExternally): Node
    fun isEqualNode(otherNode: Node?): Boolean
    fun isSameNode(otherNode: Node?): Boolean
    fun compareDocumentPosition(other: Node): Short
    fun contains(other: Node?): Boolean
    fun lookupPrefix(namespace: String?): String?
    fun lookupNamespaceURI(prefix: String?): String?
    fun isDefaultNamespace(namespace: String?): Boolean
    fun insertBefore(node: Node, child: Node?): Node
    fun appendChild(node: Node): Node
    fun replaceChild(node: Node, child: Node): Node
    fun removeChild(child: Node): Node

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

```kotlin
public external interface GetRootNodeOptions {
    var composed: Boolean? /* = false */
        get() =
        set(value) = definedExternally
}
```

```kotlin
@kotlin.internal.InlineOnly
public inline fun GetRootNodeOptions(composed: Boolean? = false): GetRootNodeOptions {
    val o = js("({})")
    o["composed"] = composed
    return o
}
```

```kotlin
/**
 * Exposes the JavaScript [Document](https://developer.mozilla.org/en/docs/Web/API/Document) to Kotlin
 */
    open val implementation:

```
DOMImplementation
    open val URL: String
    open val documentURI: String
    open val origin: String
    open val compatMode: String
    open val characterSet: String
    open val inputEncoding: String
    open val contentType: String
    open val doctype: DocumentType
    open val documentElement: Element
    open val location: Location
    var domain: String
    open val referrer: String
    var cookie: String
    open val lastModified: String
    open val readyState: DocumentReadyState
    var title: String
    var dir: String
    var body: HTMLElement
    open val head: HTMLHeadElement
    open val images: HTMLCollection
    open val embeds: HTMLCollection
    open val links: HTMLCollection
    open val forms: HTMLCollection
    open val scripts: HTMLCollection
    open val currentScript: HTMLOrSVGScriptElement
    open val defaultView: Window
    open val activeElement: Element
    var designMode: String
    var onreadystatechange: ((Event) -> dynamic)
    var fgColor: String
    var linkColor: String
    var vlinkColor: String
    var alinkColor: String
    var bgColor: String
    open val anchors: HTMLCollection
    open val applets: HTMLCollection
    open val all: HTMLAllCollection
    open val scrollingElement: Element
    open val styleSheets: StyleSheetList
    open val rootElement: SVGSVGElem,...
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

*/

var Document.get(name: String): dynamic =
    asDynamic()[name]

/**
 * Exposes the JavaScript [XMLDocument](https://developer.mozilla.org/en/docs/Web/API/XMLDocument) to Kotlin
 */
public external open class XMLDocument : Document {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface ElementCreationOptions {
    var `is`: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ElementCreationOptions(`is`: String? = undefined): ElementCreationOptions {
    val o = js("({})")
    o["is"] = `is`
    return o
}

/**
 * Exposes the JavaScript [DocumentType](https://developer.mozilla.org/en/docs/Web/API/DocumentType) to Kotlin
 */
public external abstract class DocumentType : Node, ChildNode {
    open val name: String
    open val publicId: String
    open val systemId: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [DocumentFragment](https://developer.mozilla.org/en/docs/Web/API/DocumentFragment) to Kotlin
 */
public external open class DocumentFragment : Node, NonElementParentNode, ParentNode {
    override val children: HTMLCollection
    override val firstElementChild: Element?
    override val lastElementChild: Element?
    override val childElementCount: Int
    override fun getElementById(elementId: String): Element?
    override fun querySelector(selectors: String): Element?
    override fun querySelectorAll(selectors: String): NodeList

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */); DOMQuad
override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */); DOMQuad
override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */); DOMQuad
override fun before(vararg nodes: dynamic)
override fun after(vararg nodes: dynamic)
override fun replaceWith(vararg nodes: dynamic)
override fun remove()

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript [CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin
 */
public external open class CDATASection : Text {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin
 */
public external abstract class ProcessingInstruction : CharacterData, LinkStyle,
UnionElementOrProcessingInstruction {
    open val target: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [Comment](https://developer.mozilla.org/en/docs/Web/API/Comment) to Kotlin
 */
public external open class Comment(data: String = definedExternally) : CharacterData {
    override val previousElementSibling: Element?
    override val nextElementSibling: Element?
    override fun before(vararg nodes: dynamic)
    override fun after(vararg nodes: dynamic)
    override fun replaceWith(vararg nodes: dynamic)
    override fun remove()

companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}
**Exposes the JavaScript [Range](https://developer.mozilla.org/en/docs/Web/API/Range) to Kotlin */

```kotlin
public external open class Range {
    open val startContainer: Node
    open val startOffset: Int
    open val endContainer: Node
    open val endOffset: Int
    fun setStartBefore(node: Node)
    fun setStartAfter(node: Node)
    fun setEndBefore(node: Node)
    fun setEndAfter(node: Node)
    fun collapse(toStart: Boolean = definedExternally)
    fun selectNode(node: Node)
    fun selectNodeContents(node: Node)
    fun compareBoundaryPoints(how: Short, sourceRange: Range): Short
    fun deleteContents()
    fun extractContents(): DocumentFragment
    fun cloneContents(): DocumentFragment
    fun insertNode(node: Node)
    fun surroundContents(newParent: Node)
    fun cloneRange(): Range
    fun detach()
    fun isPointInRange(node: Node, offset: Int): Boolean
    fun comparePoint(node: Node, offset: Int): Short
    fun intersectsNode(node: Node): Boolean
    fun getClientRects(): Array<DOMRect>
    fun getBoundingClientRect(): DOMRect
    fun createContextualFragment(fragment: String): DocumentFragment
}
```

**Exposes the JavaScript [NodeIterator](https://developer.mozilla.org/en/docs/Web/API/NodeIterator) to Kotlin */

```kotlin
public external abstract class NodeIterator {
    open val root: Node
    open val referenceNode: Node
    open val pointerBeforeReferenceNode: Boolean
    open val whatToShow: Int
    open val filter: NodeFilter?
    fun nextNode(): Node?
    fun previousNode(): Node?
    fun detach()
}
```

**Exposes the JavaScript [TreeWalker](https://developer.mozilla.org/en/docs/Web/API/TreeWalker) to Kotlin */

```kotlin
public external abstract class TreeWalker {
    open val root: Node
    open val whatToShow: Int
    open val filter: NodeFilter?
    open var currentNode: Node
    fun parentNode(): Node?
    fun firstChild(): Node?
    fun lastChild(): Node?
    fun previousSibling(): Node?
    fun nextSibling(): Node?
    fun previousNode(): Node?
    fun nextNode(): Node?
}
```

**Exposes the JavaScript [NodeFilter](https://developer.mozilla.org/en/docs/Web/API/NodeFilter) to Kotlin */

```kotlin
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NodeFilter {
    fun acceptNode(node: Node): Short
}
```

**Exposes the JavaScript [DOMTokenList](https://developer.mozilla.org/en/docs/Web/API/DOMTokenList) to Kotlin */

```kotlin
public external abstract class DOMTokenList : ItemArrayLike<String> {
    open var value: String
    fun contains(token: String): Boolean
    fun add(vararg tokens: String)
    fun remove(vararg tokens: String)
    fun toggle(token: String, force: Boolean = definedExternally): Boolean
    fun replace(token: String, newToken: String)
    fun supports(token: String): Boolean
    override fun item(index: Int): String?
}
```

**Exposes the JavaScript [DOMPointReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMPointReadOnly) to Kotlin */

```kotlin
public external open class DOMPointReadOnly(x: Double, y: Double, z: Double, w: Double) {
    open val x: Double
    open val y: Double
    open val z: Double
    open val w: Double
    fun matrixTransform(matrix: DOMMatrix): DOMPoint
}
```

**Exposes the JavaScript [DOMPoint](https://developer.mozilla.org/en/docs/Web/API/DOMPoint) to Kotlin */

```kotlin
public external open class DOMPoint : DOMPointReadOnly {
    constructor(init: DOMPointInit)
}
```

---

**Open Source Used In DNAC Platform Antelope 14148**
definedExternally, y: Double = definedExternally, z: Double = definedExternally, w: Double = definedExternally}

override var x: Double
    override var y: Double
    override var z: Double
    override var w: Double
}

/**
 * Exposes the JavaScript
 [DOMPointInit](https://developer.mozilla.org/en/docs/Web/API/DOMPointInit) to Kotlin
 */
public external interface DOMPointInit {
    var x: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var y: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var z: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var w: Double? /* = 1.0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun DOMPointInit(x: Double? = 0.0, y: Double? = 0.0, z: Double? = 0.0, w: Double? = 1.0): DOMPointInit {
    val o = js("({})")
    o["x"] = x
    o["y"] = y
    o["z"] = z
    o["w"] = w
    return o
}

/**
 * Exposes the JavaScript
 [DOMRect](https://developer.mozilla.org/en/docs/Web/API/DOMRect) to Kotlin
 */
public external open class DOMRect(x: Double = definedExternally, y: Double = definedExternally, width: Double = definedExternally, height: Double = definedExternally) : DOMRectReadOnly {
    override var x: Double
    override var y: Double
    override var width: Double
    override var height: Double
}

/**
 * Exposes the JavaScript
 [DOMRectReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMRectReadOnly) to Kotlin
 */
public external open class DOMRectReadOnly(x: Double, y: Double, width: Double, height: Double) {
    open val x: Double
    open val y: Double
    open val width: Double
    open val height: Double
    open val top: Double
    open val right: Double
    open val bottom: Double
    open val left: Double
}

public external interface DOMRectInit {
    var x: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var y: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var width: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
    var height: Double? /* = 0.0 */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun DOMRectInit(x: Double? = 0.0, y: Double? = 0.0, width: Double? = 0.0, height: Double? = 0.0): DOMRectInit {
    val o = js("({})")
    o["x"] = x
    o["y"] = y
    o["width"] = width
    o["height"] = height
    return o
}

public external interface DOMRectList : ItemArrayLike<DOMRect> {
    override fun item(index: Int): DOMRect?
}

@kotlin.internal.InlineOnly
public inline operator fun DOMRectList.get(index: Int): DOMRect? = asDynamic()[index]

/**
 * Exposes the JavaScript
 [DOMQuad](https://developer.mozilla.org/en/docs/Web/API/DOMQuad) to Kotlin
 */
public external open class DOMQuad {
    constructor(p1: DOMPointInit = definedExternally, p2: DOMPointInit = definedExternally, p3: DOMPointInit = definedExternally, p4: DOMPointInit = definedExternally)
    constructor(rect: DOMRectInit)
    open val p1: DOMPoint
    open val p2: DOMPoint
    open val p3: DOMPoint
    open val p4: DOMPoint
    open val bounds: DOMRectReadOnly
}

/**
 * Exposes the JavaScript
 [DOMMatrixReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMMatrixReadOnly) to Kotlin
 */
public external open class DOMMatrixReadOnly(numberSequence: Array<Double>) {
    open val a: Double
    open val b: Double
    open val c: Double
    open val d: Double
    open val e: Double
    open val f: Double
    open val m11: Double
    open val m12: Double
    open val m13: Double
    open val m14: Double
    open val m21: Double
    open val m22: Double
    open val m23: Double
    open val m24: Double
    open val m31: Double
    open val m32: Double
    open val m33: Double
    open val m34: Double
    open val m41: Double
    open val m42: Double
    open val m43: Double
    open val m44: Double
    open val is2D: Boolean
    open val isIdentity: Boolean
    fun translate(tx: Double, ty: Double, tz: Double)
    fun scale(scale: Double, originX: Double, originY: Double)
    fun scaleNonUniform(scaleX: Double, scaleY: Double, scaleZ: Double)
    fun rotate(angle: Double, originX: Double, originY: Double)
    fun rotateFromVector(x: Double, y: Double, z: Double)
}
Double, y: Double): DOMMatrix
    fun rotateAxisAngle(x: Double, y: Double, z: Double, angle: Double): DOMMatrix
    fun skewX(x: Double): DOMMatrix
    fun skewY(y: Double): DOMMatrix
    fun multiply(other: DOMMatrix): DOMMatrix
    fun flipX(): DOMMatrix
    fun flipY(): DOMMatrix
    fun transformPoint(point: DOMPointInit = definedExternally): DOMPoint
    fun toFloat32Array(): Float32Array
    fun toFloat64Array(): Float64Array
    fun flipX(): DOMMatrix
    fun flipY(): DOMMatrix
    fun inverse(): DOMMatrix
    fun transformPoint(point: DOMPointInit = definedExternally): DOMPoint
    fun toFloat32Array(): Float32Array
    fun toFloat64Array(): Float64Array
}

/**
 * Exposes the JavaScript [DOMMatrix](https://developer.mozilla.org/en/docs/Web/API/DOMMatrix) to Kotlin
 */
public external open class DOMMatrix() : DOMMatrixReadOnly {
    constructor(transformList: String)
    constructor(other: DOMMatrixReadOnly)
    constructor(array32: Float32Array)
    constructor(array64: Float64Array)
    constructor(numberSequence: Array<Double>)

    override var a: Double
    override var b: Double
    override var c: Double
    override var d: Double
    override var e: Double
    override var f: Double
    override var m11: Double
    override var m12: Double
    override var m13: Double
    override var m14: Double
    override var m21: Double
    override var m22: Double
    override var m23: Double
    override var m24: Double
    override var m31: Double
    override var m32: Double
    override var m33: Double
    override var m34: Double
    override var m41: Double
    override var m42: Double
    override var m43: Double
    override var m44: Double
    fun multiplySelf(other: DOMMatrix): DOMMatrix
    fun preMultiplySelf(other: DOMMatrix): DOMMatrix
    fun translateSelf(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix
    fun scaleSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
    fun scale3dSelf(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix
    fun scaleNonUniformSelf(scaleX: Double, scaleY: Double, scaleZ: Double = definedExternally): DOMMatrix
    fun rotateSelf(angle: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix
    fun rotateFromVectorSelf(x: Double, y: Double): DOMMatrix
    fun rotateAxisAngleSelf(x: Double, y: Double, z: Double, angle: Double): DOMMatrix
    fun skewXSelf(sx: Double): DOMMatrix
    fun skewYSelf(sy: Double): DOMMatrix
    fun invertSelf(): DOMMatrix
    fun setMatrixValue(transformList: String): DOMMatrix
}

public external interface ScrollOptions {
    var behavior: ScrollBehavior? /* = ScrollBehavior.AUTO */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ScrollOptions(behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollOptions {
    val o = js("({})")
    o["behavior"] = behavior
    return o
}

/**
 * Exposes the JavaScript [ScrollToOptions](https://developer.mozilla.org/en/docs/Web/API/ScrollToOptions) to Kotlin
 */
public external interface ScrollToOptions : ScrollOptions {
    var left: Double?
        get() = definedExternally
        set(value) = definedExternally
    var top: Double?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["left"] = left
    o["top"] = top
    o["behavior"] = behavior
    return o
}

/**
 * Exposes the JavaScript [MediaQueryList](https://developer.mozilla.org/en/docs/Web/API/MediaQueryList) to Kotlin
 */
public external abstract class MediaQueryList : EventTarget {
    open val media: String
    open val matches: Boolean
    open var onchange: ((Event) -> dynamic)?

    fun addListener(listener: EventListener?)
    fun addListener(listener: ((Event) -> Unit)?)
    fun removeListener(listener: EventListener?)
    fun removeListener(listener: ((Event) -> Unit)?)
}

/**
 * Exposes the JavaScript [MediaQueryListEvent](https://developer.mozilla.org/en/docs/Web/API/MediaQueryListEvent) to Kotlin
 */
public external open class MediaQueryListEvent(type: String, eventInitDict: MediaQueryListEventInit = definedExternally) : Event {
    open val media: String
    open val matches: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }

    public external interface MediaQueryListEventInit : EventInit {
        var media: String? /* = "" */
            get() = definedExternally
            set(value) = definedExternally
        var matches: Boolean? /* = false */
            get() = definedExternally
            set(value) = definedExternally
    }

    @kotlin.internal.InlineOnly
    val o = js("({})")
    o["media"] = media
    o["matches"] = matches
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [Screen](https://developer.mozilla.org/en/docs/Web/API/Screen) to Kotlin
 */
public external abstract class Screen {
    open val availWidth: Int
    open val availHeight: Int
    open val open

/**
 * Exposes the JavaScript [CaretPosition](https://developer.mozilla.org/en/docs/Web/API/CaretPosition) to Kotlin
 */
public external abstract class CaretPosition {
    open val offsetNode: Node
    open val offset: Int
}

public external interface ScrollIntoViewOptions : ScrollOptions {
    var block: ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */
    fun getClientRect(): DOMRect?
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["block"] = block
    o["inline"] = inline
    o["behavior"] = behavior
    o["return o"] = return o
    return o
}

public external interface ConvertCoordinateOptions {
    var fromBox: CSSBoxType? /* = CSSBoxType.BORDER */
    var toBox: CSSBoxType? /* = CSSBoxType.BORDER */
}

@kotlin.internal.InlineOnly
public inline fun ConvertCoordinateOptions(fromBox: CSSBoxType? = CSSBoxType.BORDER, toBox: CSSBoxType? = CSSBoxType.BORDER): ConvertCoordinateOptions {
    val o = js("({})")
    o["fromBox"] = fromBox
    o["toBox"] = toBox
    return o
}

/**
 * Exposes the JavaScript [GeometryUtils](https://developer.mozilla.org/en/docs/Web/API/GeometryUtils) to Kotlin
 */
public external interface GeometryUtils {
    fun getBoxQuads(options: BoxQuadOptions = definedExternally): Array<DOMQuad>
    fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMQuad
    fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMQuad
    fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMPoint
}

/**
 * Exposes the JavaScript [Touch](https://developer.mozilla.org/en/docs/Web/API/Touch) to Kotlin
 */
public external open class Touch {
    open val identifier: Int
    open val target: EventTarget
    open val screenX: Int
    open val screenY: Int
    open val clientX: Int
    open val clientY: Int
    open val pageX: Int
    open val pageY: Int
    open val region: String?
}

public external open class TouchList : ItemArrayLike<Touch> {
    override fun item(index: Int): Touch?
}

@kotlin.internal.InlineOnly
public inline operator fun TouchList.get(index: Int): Touch? = asDynamic()[index]

public external open class TouchEvent : UIEvent {
    open val touches: TouchList
    open val targetTouches: TouchList
    open val changedTouches: TouchList
    open val altKey: Boolean
    open val metaKey: Boolean
    open val shiftKey: Boolean
    open val ctrlKey: Boolean
}

/**
 * Exposes the JavaScript [Image](https://developer.mozilla.org/en/docs/Web/API/Image) to Kotlin
 */
public external open class Image(width: Int = definedExternally, height: Int = definedExternally) : HTMLImageElement {
    override var onabort: ((Event) -> dynamic)?
    override var onblur: ((FocusEvent) -> dynamic)?
    override var canplay: ((Event) -> dynamic)?
    override var oncanplay: ((Event) -> dynamic)?
    override var oncanplaythrough: (Event) -> dynamic)?
}
override val isContentEditable: Boolean
override val style: CSSStyleDeclaration
override val children: HTMLCollection
override val firstElementChild: Element?
override val lastElementChild: Element?
override val childElementCount: Int
override val previousElementSibling: Element?
override val nextElementSibling: Element?
override val assignedSlot: HTMLSlotElement
override fun prepend(vararg nodes: dynamic)
override fun append(vararg nodes: dynamic)
override fun querySelector(selectors: String): Element?
override fun querySelectorAll(selectors: String): NodeList
override fun before(vararg nodes: dynamic)
override fun after(vararg nodes: dynamic)
override fun replaceWith(vararg nodes: dynamic)
override fun remove()
override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>
override fun convertRectFromNode(rect: DOMRectInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad
override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint

companion object {
    val NETWORK_EMPTY: Short
    val NETWORK_IDLE: Short
    val NETWORK_LOADING: Short
    val NETWORK_NO_SOURCE: Short
    val HAVE_NOTHING: Short
    val HAVE_METADATA: Short
    val HAVE_CURRENT_DATA: Short
    val HAVE_FUTURE_DATA: Short
    val HAVE_ENOUGH_DATA: Short
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
}

/**
 * Exposes the JavaScript
 */
public external open class Option(text: String = definedExternally, value: String = definedExternally, defaultSelected: Boolean = definedExternally, selected: Boolean = definedExternally) : HTMLOptionElement {
    override var onabort: ((Event) -> dynamic)?
    override var onblur: ((FocusEvent) -> dynamic)?
    override var oncancel: ((Event) -> dynamic)?
    override var oncanplay: ((Event) -> dynamic)?
    override var oncanplaythrough: ((Event) -> dynamic)?
    override var onchange: ((Event) -> dynamic)?
    override var onclick: ((MouseEvent) -> dynamic)?
    override var onclose: ((Event) -> dynamic)?
    override var oncontextmenu: ((MouseEvent) -> dynamic)?
    override var oncuechange: ((Event) -> dynamic)?
    override var ondblclick: ((MouseEvent) -> dynamic)?
    override var ondrag: ((DragEvent) -> dynamic)?
    override var ondragend: ((DragEvent) -> dynamic)?
    override var ondragenter: ((DragEvent) -> dynamic)?
    override var ondragleave: ((DragEvent) -> dynamic)?
    override var ondragover: ((DragEvent) -> dynamic)?
    override var ondragstart: ((DragEvent) -> dynamic)?
    override var ondurationchange: ((Event) -> dynamic)?
    override var onemptied: ((Event) -> dynamic)?
    override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?
    override var onfocus: ((FocusEvent) -> dynamic)?
    override var onkeydown: ((KeyboardEvent) -> dynamic)?
    override var onkeypress: ((KeyboardEvent) -> dynamic)?
    override var onkeyup: ((KeyboardEvent) -> dynamic)?
    override var onloading: ((Event) -> dynamic)?
    override var onloadeddata: ((Event) -> dynamic)?
    override var onloadedmetadata: ((Event) -> dynamic)?
    override var onmessage: ((MessageEvent) -> dynamic)?
    override var onmousedown: ((MouseEvent) -> dynamic)?
    override var onmousemove: ((MouseEvent) -> dynamic)?
    override var onmouseout: ((MouseEvent) -> dynamic)?
    override var onmouseover: ((MouseEvent) -> dynamic)?
    override var onmouseup: ((MouseEvent) -> dynamic)?
    override var onmessage: ((MessageEvent) -> dynamic)?
    override var onmousemove: ((MouseEvent) -> dynamic)?
    override var onmouseup: ((MouseEvent) -> dynamic)
override fun onmousedown((MouseEvent) -> dynamic)?
        override fun onmouseup((MouseEvent) -> dynamic)?
        override fun onmousemove((MouseEvent) -> dynamic)?
        override fun onmousedown((MouseEvent) -> dynamic)?
        override fun onmouseup((MouseEvent) -> dynamic)?
        override fun onmouseout((MouseEvent) -> dynamic)?
        override fun onmouseover((MouseEvent) -> dynamic)?
        override fun onmousedown((MouseEvent) -> dynamic)?
        override fun onmouseup((MouseEvent) -> dynamic)?
        override fun onmousemove((MouseEvent) -> dynamic)?
        override fun onmouseout((MouseEvent) -> dynamic)?
        override fun onmouseover((MouseEvent) -> dynamic)?
        override fun onmousemove((MouseEvent) -> dynamic)?
        override fun onmouseout((MouseEvent) -> dynamic)?
        override fun onmouseover((MouseEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dynamic)?
        override fun onkeydown((KeyboardEvent) -> dynamic)?
        override fun onkeypress((KeyboardEvent) -> dynamic)?
        override fun onkeyup((KeyboardEvent) -> dyna
interface DocumentReadyState {
    companion object
}


/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanPlayTypeResult {
    companion object
}

public inline val CanPlayTypeResult.Companion.EMPTY: CanPlayTypeResult get() = "".asDynamic().unsafeCast<CanPlayTypeResult>()

public inline val CanPlayTypeResult.Companion.MAYBE: CanPlayTypeResult get() = "maybe".asDynamic().unsafeCast<CanPlayTypeResult>()

public inline val CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() = "probably".asDynamic().unsafeCast<CanPlayTypeResult>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface TextTrackMode {
    companion object
}

public inline val TextTrackMode.Companion.DISABLED: TextTrackMode get() = "disabled".asDynamic().unsafeCast<TextTrackMode>()

public inline val TextTrackMode.Companion.HIDDEN: TextTrackMode get() = "hidden".asDynamic().unsafeCast<TextTrackMode>()

public inline val TextTrackMode.Companion.SHOWING: TextTrackMode get() = "showing".asDynamic().unsafeCast<TextTrackMode>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface TextTrackKind {
    companion object
}

public inline val TextTrackKind.Companion.SUBTITLES: TextTrackKind get() = "subtitles".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.CAPTIONS: TextTrackKind get() = "captions".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.DESCRIPTIONS: TextTrackKind get() = "descriptions".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.CHAPTERS: TextTrackKind get() = "chapters".asDynamic().unsafeCast<TextTrackKind>()

public inline val TextTrackKind.Companion.METADATA: TextTrackKind get() = "metadata".asDynamic().unsafeCast<TextTrackKind>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface SelectionMode {
    companion object
}

public inline val SelectionMode.Companion.SELECT: SelectionMode get() = "select".asDynamic().unsafeCast<SelectionMode>()

public inline val SelectionMode.Companion.START: SelectionMode get() = "start".asDynamic().unsafeCast<SelectionMode>()

public inline val SelectionMode.Companion.END: SelectionMode get() = "end".asDynamic().unsafeCast<SelectionMode>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasFillRule {
    companion object
}

public inline val CanvasFillRule.Companion.NONZERO: CanvasFillRule get() = "nonzero".asDynamic().unsafeCast<CanvasFillRule>()

public inline val CanvasFillRule.Companion.EVENODD: CanvasFillRule get() = "evenodd".asDynamic().unsafeCast<CanvasFillRule>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ImageSmoothingQuality {
    companion object
}


/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasFillRule {
    companion object
}

public inline val CanvasFillRule.Companion.NONZERO: CanvasFillRule get() = "nonzero".asDynamic().unsafeCast<CanvasFillRule>()

public inline val CanvasFillRule.Companion.EVENODD: CanvasFillRule get() = "evenodd".asDynamic().unsafeCast<CanvasFillRule>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ImageSmoothingQuality {
    companion object
}


"medium".asDynamic().unsafeCast<ImageSmoothingQuality>()


/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasLineCap {
    companion object
}

public inline val CanvasLineCap.Companion.BUTT: CanvasLineCap get() = "butt".asDynamic().unsafeCast<CanvasLineCap>()

public inline val CanvasLineCap.Companion.ROUND: CanvasLineCap get() = "round".asDynamic().unsafeCast<CanvasLineCap>()

public inline val CanvasLineCap.Companion.SQUARE: CanvasLineCap get() = "square".asDynamic().unsafeCast<CanvasLineCap>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasLineJoin {
    companion object
}

public inline val CanvasLineJoin.Companion.BUTT: CanvasLineJoin get() = "butt".asDynamic().unsafeCast<CanvasLineJoin>()

public inline val CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() = "round".asDynamic().unsafeCast<CanvasLineJoin>()

public inline val CanvasLineJoin.Companion.BEVEL: CanvasLineJoin get() = "bevel".asDynamic().unsafeCast<CanvasLineJoin>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasTextAlign {
    companion object
}

public inline val CanvasTextAlign.Companion.START: CanvasTextAlign get() = "start".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.END: CanvasTextAlign get() = "end".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.LEFT: CanvasTextAlign get() = "left".asDynamic().unsafeCast<CanvasTextAlign>()

public inline val CanvasTextAlign.Companion.RIGHT: CanvasTextAlign get() = "right".asDynamic().unsafeCast<CanvasTextAlign>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CanvasDirection {
    companion object
}

public inline val CanvasDirection.Companion.LTR: CanvasDirection get() = "ltr".asDynamic().unsafeCast<CanvasDirection>()

public inline val CanvasDirection.Companion.RTL: CanvasDirection get() = "rtl".asDynamic().unsafeCast<CanvasDirection>()

public inline val CanvasDirection.Companion.INHERIT: CanvasDirection get() = "inherit".asDynamic().unsafeCast<CanvasDirection>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface ScrollRestoration {
    companion object
}

public inline val ScrollRestoration.Companion.AUTO: ScrollRestoration get() = "auto".asDynamic().unsafeCast<ScrollRestoration>()


/* please, don't implement this interface! */
"end".asDynamic().unsafeCast<ScrollLogicalPosition>()

public inline val ScrollLogicalPosition.Companion.NEAREST: ScrollLogicalPosition get() = "nearest".asDynamic().unsafeCast<ScrollLogicalPosition>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface CSSBoxType {

companion object
}

public inline val CSSBoxType.Companion.MARGIN: CSSBoxType get() = "margin".asDynamic().unsafeCast<CSSBoxType>()

public inline val CSSBoxType.Companion.BORDER: CSSBoxType get() = "border".asDynamic().unsafeCast<CSSBoxType>()

public inline val CSSBoxType.Companion.PADDING: CSSBoxType get() = "padding".asDynamic().unsafeCast<CSSBoxType>()

public inline val CSSBoxType.Companion.CONTENT: CSSBoxType get() = "content".asDynamic().unsafeCast<CSSBoxType>()

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED,
DO NOT EDIT! See github.com/kotlin/dukat for details

package org.w3c.fetch

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript [Headers](https://developer.mozilla.org/en/docs/Web/API/Headers) to Kotlin
 */
public external open class Headers(init: dynamic = definedExternally) {
    fun append(name: String, value: String)
    fun delete(name: String)
    fun get(name: String): String?
    fun has(name: String): Boolean
    fun set(name: String, value: String)
}

/**
 * Exposes the JavaScript [Body](https://developer.mozilla.org/en/docs/Web/API/Body) to Kotlin
 */
public external interface Body {
    val bodyUsed: Boolean
    fun arrayBuffer(): Promise<ArrayBuffer>
    fun blob(): Promise<Blob>
    fun formData(): Promise<FormData>
    fun json(): Promise<Any?>
    fun text(): Promise<String>
}

/**
 * Exposes the JavaScript [Request](https://developer.mozilla.org/en/docs/Web/API/Request) to Kotlin
 */
public external open class Request(input: dynamic, init: RequestInit = definedExternally) : Body {
    open val method: String
    open val url: String
    open val headers: Headers
    open val type: RequestType
    open val destination: RequestDestination
    open val referrer: String
    open val referrerPolicy: dynamic
    open val mode: RequestMode
    open val credentials: RequestCredentials
    open val cache: RequestCache
    open val redirect: RequestRedirect
    open val integrity: String
    open val keepalive: Boolean
    override val bodyUsed: Boolean
    fun clone(): Request
    override fun arrayBuffer(): Promise<ArrayBuffer>
    override fun blob(): Promise<Blob>
    override fun formData(): Promise<FormData>
    override fun json(): Promise<Any?>
    override fun text(): Promise<String>
}

public external interface RequestInit {
    var method: String?
        get() = definedExternally
        set(value) = definedExternally
    var headers: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var body: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var referrer: String?
        get() = definedExternally
        set(value) = definedExternally
    var referrerPolicy: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var mode: RequestMode
        get() = definedExternally
        set(value) = definedExternally
    var credentials: RequestCredentials
        get() = definedExternally
        set(value) = definedExternally
    var cache: RequestCache
        get() = definedExternally
        set(value) = definedExternally
    var redirect: RequestRedirect
        get() = definedExternally
        set(value) = definedExternally
    var integrity: String
        get() = definedExternally
        set(value) = definedExternally
    var keepalive: Boolean
        get() = definedExternally
        set(value) = definedExternally
    var window: Any?
        get() = definedExternally
        set(value) = definedExternally
    var @kotlin.internal.InlineOnly
        get() = definedExternally
        set(value) = definedExternally
}

Open Source Used In DNAC Platform Antelope 14159
    val o = js("({})")
    o["method"] = method
    o["headers"] = headers
    o["body"] = body
    o["referrer"] = referrer
    o["referrerPolicy"] = referrerPolicy
    o["mode"] = mode
    o["credentials"] = credentials
    o["cache"] = cache
    o["redirect"]
    o["integrity"] = integrity
    o["keepalive"] = keepalive
    o["window"] = window
    return o
}

/**
 * Exposes the JavaScript [Response](https://developer.mozilla.org/en/docs/Web/API/Response) to Kotlin
 */
public external open class Response(body: dynamic = definedExternally, init: ResponseInit = definedExternally) : Body {  
    open val type: ResponseType
    open val url: String
    open val redirected: Boolean
    open val status: Short
    open val ok: Boolean
    open val statusText: String
    open val headers: Headers
    open val body: dynamic
    open val trailer: Promise<Headers>

    override val bodyUsed: Boolean

    fun clone(): Response
    override fun arrayBuffer(): Promise<ArrayBuffer>
    override fun blob(): Promise<Blob>
    override fun formData(): Promise<FormData>
    override fun json(): Promise<Any?>

    override fun text(): Promise<String>

    companion object {
        fun error(): Response
        fun redirect(url: String, status: Short = definedExternally): Response
    }
}

public external interface ResponseInit {
    var status: Short? /* = 200 */
        get() = definedExternally
        set(value) = definedExternally
    var statusText: String? /* = "OK" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["status"] = status
    o["statusText"] = statusText
    o["headers"] = headers
    return o
}

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestType {
    companion object
}

public inline val RequestType.Companion.EMPTY: RequestType get() = "".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.AUDIO: RequestType get() = "audio".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.FONT: RequestType get() = "font".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.IMAGE: RequestType get() = "image".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.SCRIPT: RequestType get() = "script".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.STYLE: RequestType get() = "style".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.TRACK: RequestType get() = "track".asDynamic().unsafeCast<RequestType>()

public inline val RequestType.Companion.VIDEO: RequestType get() = "video".asDynamic().unsafeCast<RequestType>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestDestination {
    companion object
}

public inline val RequestDestination.Companion.EMPTY: RequestDestination get() = "".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.DOCUMENT: RequestDestination get() = "document".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.TRACK: RequestDestination get() = "track".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.VIDEO: RequestDestination get() = "video".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.MANIFEST: RequestDestination get() = "manifest".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.MEDIA: RequestDestination get() =
"media".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.OBJECT: RequestDestination get() =
"object".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.REPORT: RequestDestination get() =
"report".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.SCRIPT: RequestDestination get() =
"script".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.SERVICEWORKER: RequestDestination get() =
"serviceworker".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.SHAREDWORKER: RequestDestination get() =
"sharedworker".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.STYLE: RequestDestination get() =
"style".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.WORKER: RequestDestination get() =
"worker".asDynamic().unsafeCast<RequestDestination>()

public inline val RequestDestination.Companion.XSLT: RequestDestination get() =
"xslt".asDynamic().unsafeCast<RequestDestination>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestMode {

    companion object
}

public inline val RequestMode.Companion.NAVIGATE: RequestMode get() =
"navigate".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.SAME_ORIGIN: RequestMode get() = "same-origin".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.NO_CORS: RequestMode get() = "no-cors".asDynamic().unsafeCast<RequestMode>()

public inline val RequestMode.Companion.CORS: RequestMode get() = "cors".asDynamic().unsafeCast<RequestMode>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestCredentials {

    companion object
}

public inline val RequestCredentials.Companion.OMIT: RequestCredentials get() = "omit".asDynamic().unsafeCast<RequestCredentials>()

public inline val RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = "same-origin".asDynamic().unsafeCast<RequestCredentials>()

public inline val RequestCredentials.Companion.INCLUDE: RequestCredentials get() = "include".asDynamic().unsafeCast<RequestCredentials>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestCache {

    companion object
}

public inline val RequestCache.Companion.DEFAULT: RequestCache get() = "default".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.NO_STORE: RequestCache get() = "no-store".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.RELOAD: RequestCache get() = "reload".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.NO_CACHE: RequestCache get() = "no-cache".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.FORCE_CACHE: RequestCache get() = "force-cache".asDynamic().unsafeCast<RequestCache>()

public inline val RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = "only-if-cached".asDynamic().unsafeCast<RequestCache>()

/* please, don't implement this interface! */
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface RequestRedirect {

    companion object
}

public inline val RequestRedirect.Companion.FOLLOW: RequestRedirect get() = "follow".asDynamic().unsafeCast<RequestRedirect>()

public inline val RequestRedirect.Companion.ERROR: RequestRedirect get() = "error".asDynamic().unsafeCast<RequestRedirect>()

public inline val RequestRedirect.Companion.MANUAL: RequestRedirect get() =
"manual".asDynamic().unsafeCast<RequestRedirect>()

public inline val RequestRedirect.Companion.POST: RequestRedirect get() =
"post".asDynamic().unsafeCast<RequestRedirect>()
MediaTrackSupportedConstraints::

```kotlin

    val o = js("({})")
    o["width"] = width
    o["height"] = height
    o["aspectRatio"] = aspectRatio
    o["frameRate"] = frameRate
    o["facingMode"] = facingMode
    o["resizeMode"] = resizeMode
    o["volume"] = volume
    o["sampleRate"] = sampleRate
    o["sampleSize"] = sampleSize
    o["echoCancellation"] = echoCancellation
    o["autoGainControl"] = autoGainControl
    o["noiseSuppression"] = noiseSuppression
    o["latency"] = latency
    o["channelCount"] = channelCount
    o["deviceId"] = deviceId
    o["groupId"] = groupId
    return o
```

MediaTrackCapabilities::

```kotlin
public external interface MediaTrackCapabilities {
    var width: ULongRange?
    var height: ULongRange?
    var aspectRatio: DoubleRange?
    var frameRate: DoubleRange?
    var facingMode: Array<String>?
    var resizeMode: Array<String>?
    var volume: DoubleRange?
    var sampleRate: ULongRange?
    var sampleSize: ULongRange?
    var echoCancellation: Array<Boolean>?
    var autoGainControl: Array<Boolean>?
    var noiseSuppression: Array<Boolean>?
    var latency: DoubleRange?
    var channelCount: ULongRange?
    var deviceId: String?
    var groupId: String?
}
```

```
/**
 * Exposes the JavaScript [MediaTrackConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackConstraints) to Kotlin
 */
```

Open Source Used In DNAC Platform Antelope 14163

---

Open Source Used In DNAC Platform Antelope 14163
Array<MediaTrackConstraintSet>?
get() = definedExternally
set(value) = definedExternally
@kotlin.internal.InlineOnly
public inline fun MediaTrackConstraints(advanced:
Array<MediaTrackConstraintSet>?
val o = js("{}")
on o["advanced"] = advanced
on o["width"] = width
on o["height"] = height
on o["aspectRatio"] = aspectRatio
on o["frameRate"] = frameRate
on o["facingMode"] = facingMode
on o["resizeMode"] = resizeMode
on o["volume"] = volume
on o["sampleRate"] = sampleRate
on o["sampleSize"] = sampleSize
on o["echoCancellation"] = echoCancellation
on o["autoGainControl"] = autoGainControl
on o["noiseSuppression"] = noiseSuppression
on o["latency"] = latency
on o["channelCount"] = channelCount
MediaTrackConstraintSet
val o = js("{}")
on o["value"] = value
on o["width"] = width
on o["height"] = height
on o["aspectRatio"] = aspectRatio
on o["frameRate"] = frameRate
on o["facingMode"] = facingMode
on o["resizeMode"] = resizeMode
on o["volume"] = volume
on o["sampleRate"] = sampleRate
on o["sampleSize"] = sampleSize
on o["echoCancellation"] = echoCancellation
on o["autoGainControl"] = autoGainControl
on o["noiseSuppression"] = noiseSuppression
on o["latency"] = latency
on o["channelCount"] = channelCount
on o["deviceId"] = deviceId
MediaTrackSettings
val o = js("{}")
on o["value"] = value
on o["width"] = width
on o["height"] = height
on o["aspectRatio"] = aspectRatio
on o["frameRate"] = frameRate
on o["facingMode"] = facingMode
on o["resizeMode"] = resizeMode
on o["volume"] = volume
on o["sampleRate"] = sampleRate
on o["sampleSize"] = sampleSize
on o["echoCancellation"] = echoCancellation
on o["autoGainControl"] = autoGainControl
on o["noiseSuppression"] = noiseSuppression
on o["latency"] = latency
on o["channelCount"] = channelCount
on o["deviceId"] = deviceId
on o["groupId"] = groupId

Kotlin

 */
public external interface MediaStreamConstraints {
    var video: dynamic /* = false */
    get() = definedExternally
    set(value) = definedExternally
    var audio: dynamic /* = false */
    get() = definedExternally
    set(value) = definedExternally
}

Kotlin

 /*
 */
public external interface MediaStreamConstraints {
    var video: dynamic /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var audio: dynamic /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
public inline fun MediaStreamConstraints(video: dynamic = false, audio: dynamic = false): MediaStreamConstraints {
    val o = js("{}")
    o["video"] = video
    o["audio"] = audio
    return o
}

Kotlin

 public external interface ConstrainablePattern {
    var onoverconstrained: ((Event) -> dynamic)?
        get() = definedExternally
        set(value) = definedExternally
    fun getCapabilities(): Capabilities
    fun getConstraints(): Constraints
    fun getSettings(): Settings
    fun applyConstraints(constraints: Constraints = definedExternally): Promise<Unit>
}

Kotlin

 /**
 * Exposes the JavaScript
 * [DoubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin
 */
public external interface DoubleRange {
    var max: Double?
        get() = definedExternally
        set(value) = definedExternally
    var min: Double?
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
public inline fun DoubleRange(max: Double? = undefined, min: Double? = undefined): DoubleRange {
    val o = js("{}")
    o["max"] = max
    o["min"] = min
    return o
}

Kotlin

 public external interface ConstrainDoubleRange : DoubleRange {
    var exact: Double?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Double?
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    o["max"] = max
    o["min"] = min
    return o
}

Kotlin

 public external interface ULongRange {
    var max: Int?
        get() = definedExternally
        set(value) = definedExternally
    var min: Int?
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
public inline fun ULongRange(max: Int? = undefined, min: Int? = undefined): ULongRange {
    val o = js("{}")
    o["max"] = max
    o["min"] = min
    return o
}

Kotlin

 public external interface ConstrainULongRange : ULongRange {
    var exact: Int?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Int?
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    o["max"] = max
    o["min"] = min
    return o
}

Kotlin

 /*
 */
public external interface ConstrainBooleanParameters {
    var exact: Boolean?
        get() = definedExternally
        set(value) = definedExternally
    var ideal: Boolean?
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    return o
}

Kotlin

 /*
 */
public external interface ConstrainDOMStringParameters {
    var exact: dynamic
        get() = definedExternally
        set(value) = definedExternally
    var ideal: dynamic
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
public inline fun ConstrainDOMStringParameters(exact: dynamic = undefined, ideal: dynamic = undefined): ConstrainDOMStringParameters {
    val o = js("{}")
    o["exact"] = exact
    o["ideal"] = ideal
    return o
}

Kotlin

 public external interface Capabilities

 @kotlin.internal.InlineOnly
public inline fun Capabilities(): Capabilities {
    val o = js("{}")
    return o
}

Kotlin

 public external interface Settings

 @kotlin.internal.InlineOnly
public inline fun Settings(): Settings {
    val o = js("{}")
    return o
}

Kotlin

 public external interface ConstraintSet

 @kotlin.internal.InlineOnly
public inline fun ConstraintSet(): ConstraintSet {
    val o = js("{}")
    return o
}

Kotlin

 public external interface Constraints : ConstraintSet {
    var advanced: Array<ConstraintSet>?
        get() = definedExternally
        set(value) = definedExternally
    var capabilities: Capabilities
        get() = definedExternally
        set(value) = definedExternally
    var settings: Settings
        get() = definedExternally
        set(value) = definedExternally
    var constrainablePatterns: Array<ConstrainablePattern>?
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
public inline fun Constraints(advanced: Array<ConstraintSet> = undefined, capabilities: Capabilities = undefined, settings: Settings = undefined, constrainablePatterns: Array<ConstrainablePattern> = undefined): Constraints {
    val o = js("{}")
    o["advanced"] = advanced
    o["capabilities"] = capabilities
    o["settings"] = settings
    o["constrainablePatterns"] = constrainablePatterns
    return o
}

Kotlin

 public external interface DNACPlatformAntelope 14166

 /**
 * Exposes the JavaScript
 */
public external interface DNACPlatformAntelope 14166 {
    var video: dynamic /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var audio: dynamic /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

Kotlin

 @kotlin.internal.InlineOnly
public inline fun DNACPlatformAntelope 14166(video: dynamic = false, audio: dynamic = false): DNACPlatformAntelope 14166 {
    val o = js("{}")
    o["video"] = video
    o["audio"] = audio
    return o
}
    val o = js("({})")
    o["pointerId"] = pointerId
    o["width"] = width
    o["height"] = height
    o["pressure"] = pressure
    o["tangentialPressure"] = tangentialPressure
    o["tiltX"] = tiltX
    o["tiltY"] = tiltY
    o["twist"] = twist
    o["pointerType"] = pointerType
    o["isPrimary"] = isPrimary
    o["screenX"] = screenX
    o["screenY"] = screenY
    o["clientX"] = clientX
    o["clientY"] = clientY
    o["button"] = button
    o["buttons"] = buttons
    o["relatedTarget"] = relatedTarget
    o["region"] = region
    o["ctrlKey"] = ctrlKey
    o["shiftKey"] = shiftKey
    o["altKey"] = altKey
    o["metaKey"] = metaKey
    o["ModifierAltGraph"] = modifierAltGraph
    o["ModifierCapsLock"] = modifierCapsLock
    o["ModifierFn"] = modifierFn
    o["ModifierFnLock"] = modifierFnLock
    o["ModifierHyper"] = modifierHyper
    o["ModifierNumLock"] = modifierNumLock
    o["ModifierScrollLock"] = modifierScrollLock
    o["ModifierSuper"] = modifierSuper
    o["ModifierSymbol"] = modifierSymbol
    o["ModifierSymbolLock"] = modifierSymbolLock
    o["view"] = view
    o["detail"] = detail
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

/**
 * Exposes the JavaScript [PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent) to Kotlin
 */
public external open class PointerEvent(type: String, eventInitDict: PointerEventInit = definedExternally) : MouseEvent {
    open val pointerId: Int
    open val width: Double
    open val height: Double
    open val pressure: Float
    open val tangentialPressure: Float
    open val tiltX: Int
    open val tiltY: Int
    open val twist: Int
    open val pointerType: String
    open val isPrimary: Boolean

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT
// See github.com/kotlin/duktak for details

package org.w3c.dom.svg

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*
import org.w3c.xhr.*

/**
 * Exposes the JavaScript [SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin
 */
public external abstract class SVGElement : Element, ElementCSSInlineStyle, GlobalEventHandlers, SVGElementInstance {
    open val dataset: DOMStringMap
    open val ownerSVGElement: SVGSVGElement?
    open val viewportElement: SVGElement?
    open var tabIndex: Int
    fun focus()
    fun blur()

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}
definedExternally
}\n}@kotlin.internal.InlineOnly\n\npublic inline fun SVGBoundingBoxOptions(fill: Boolean? = true, stroke: Boolean? = false, markers: Boolean? = false, clipped: Boolean? = false): SVGBoundingBoxOptions {\n    val o = js("({})")\n    o["fill"] = fill\n    o["stroke"] = stroke\n    o["markers"] = markers\n    o["clipped"] = clipped\n    return o
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["fill"] = fill
    o["stroke"] = stroke
    o["markers"] = markers
    o["clipped"] = clipped
    return o
}

/**
 * Exposes the JavaScript
[SVGGraphicsElement](https://developer.mozilla.org/en/docs/Web/API/SVGGraphicsElement) to Kotlin

*\n\npublic external abstract class SVGGraphicsElement : SVGElement, SVGTests {\n
    fun getBBox(options: SVGBoundingBoxOptions = definedExternally): DOMRect
    fun getCTM(): DOMMatrix?
    fun getScreenCTM(): DOMMatrix?

    companion object {

        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript
[SVGGeometryElement](https://developer.mozilla.org/en/docs/Web/API/SVGGeometryElement) to Kotlin

*\n\npublic external abstract class SVGGeometryElement : SVGGraphicsElement {

    fun isPointInFill(point: DOMPoint): Boolean
    fun isPointInStroke(point: DOMPoint): Boolean
    fun getTotalLength(): Float
    fun getPointAtLength(distance: Float): DOMPoint

    companion object {

        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
    }
}

/**
 * Exposes the JavaScript
[SVGNumber](https://developer.mozilla.org/en/docs/Web/API/SVGNumber) to Kotlin

*\n\npublic external abstract class SVGNumber {

    fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)
    fun convertToSpecifiedUnits(unitType: Short)

    companion object {

        val SVG_LENGTHTYPE_UNKNOWN: Short
        val SVG_LENGTHTYPE_NUMBER: Short
        val SVG_LENGTHTYPE_PERCENTAGE: Short
        val SVG_LENGTHTYPE_EMS: Short
        val SVG_LENGTHTYPE_EXS: Short
        val SVG_LENGTHTYPE_PX: Short
        val SVG_LENGTHTYPE_CM: Short
        val SVG_LENGTHTYPE_MM: Short
        val SVG_LENGTHTYPE_IN: Short
        val SVG_LENGTHTYPE_PT: Short
        val SVG_LENGTHTYPE_PC: Short
    }
}

/**
 * Exposes the JavaScript
[SVGAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAngle) to Kotlin

*\n\npublic external abstract class SVGAngle {

    fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)
    fun convertToSpecifiedUnits(unitType: Short)

    companion object {

        val SVG_ANGLETYPE_UNKNOWN: Short
        val SVG_ANGLETYPE_UNSPECIFIED: Short
        val SVG_ANGLETYPE_DEG: Short
        val SVG_ANGLETYPE_RAD: Short
    }
}
Open Source Used In DNAC Platform Antelope

```kotlin
/**
 * Exposes the JavaScript [SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGNumberList) to Kotlin
 */
public external abstract class SVGNumberList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGNumber): SVGNumber
    fun insertItemBefore(newItem: SVGNumber, index: Int): SVGNumber
    fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber
    fun removeItem(index: Int): SVGNumber
    fun appendItem(newItem: SVGNumber): SVGNumber
    fun getItem(index: Int): SVGNumber
}

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.get(index: Int): SVGNumber? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript [SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin
 */
public external abstract class SVGLengthList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: SVGLength): SVGLength
    fun insertItemBefore(newItem: SVGLength, index: Int): SVGLength
    fun replaceItem(newItem: SVGLength, index: Int): SVGLength
    fun removeItem(index: Int): SVGLength
    fun appendItem(newItem: SVGLength): SVGLength
    fun getItem(index: Int): SVGLength
}

@kotlin.internal.InlineOnly
public inline operator fun SVGLengthList.get(index: Int): SVGLength? = asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGLengthList.set(index: Int, newItem: SVGLength) { asDynamic()[index] = newItem }

/**
 * Exposes the JavaScript [SVGAnimatedBoolean](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedBoolean) to Kotlin
 */
public external abstract class SVGAnimatedBoolean {
    open var baseVal: Boolean
    open val animVal: Boolean
}

/**
 * Exposes the JavaScript [SVGAnimatedEnumeration](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedEnumeration) to Kotlin
 */
public external abstract class SVGAnimatedEnumeration {
    open var baseVal: Short
    open val animVal: Short
}

/**
 * Exposes the JavaScript [SVGAnimatedInteger](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedInteger) to Kotlin
 */
public external abstract class SVGAnimatedInteger {
    open var baseVal: Int
    open val animVal: Int
}

/**
 * Exposes the JavaScript [SVGAnimatedNumber](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumber) to Kotlin
 */
public external abstract class SVGAnimatedNumber {
    open var baseVal: Float
    open val animVal: Float
}

/**
 * Exposes the JavaScript [SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin
 */
public external abstract class SVGAnimatedLength {
    open val baseVal: SVGLength
    open val animVal: SVGLength
}

/**
 * Exposes the JavaScript [SVGAnimatedAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAngle) to Kotlin
 */
public external abstract class SVGAnimatedAngle {
    open val baseVal: SVGAngle
    open val animVal: SVGAngle
}

/**
 * Exposes the JavaScript [SVGAnimatedString](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedString) to Kotlin
 */
public external abstract class SVGAnimatedString {
    open var baseVal: String
    open val animVal: String
}

/**
 * Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect) to Kotlin
 */
public external abstract class SVGAnimatedRect {
    open val baseVal: DOMRect
    open val animVal: DOMRect
}

/**
 * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin
 */
public external abstract class SVGAnimatedNumberList {
    open var baseVal: List<Number>
    open val animVal: List<Number>
}
```

---

Open Source Used In DNAC Platform Antelope 14171
*/
public external abstract class SVGAnimatedNumberList {
    open val baseVal: SVGNumberList
    open val animVal: SVGNumberList
}

/**
 * Exposes the JavaScript
*/
public external abstract class SVGAnimatedLengthList {
    open val baseVal: SVGLengthList
    open val animVal: SVGLengthList
}

/**
 * Exposes the JavaScript
*/
public external abstract class SVGStringList {
    open val length: Int
    open val numberOfItems: Int
    fun clear()
    fun initialize(newItem: String): String
    fun insertItemBefore(newItem: String, index: Int): String
    fun replaceItem(newItem: String, index: Int): String
    fun removeItem(index: Int): String
    fun appendItem(newItem: String): String
    fun getItem(index: Int): String
}

@kotlin.internal.InlineOnly
public inline operator fun SVGStringList.get(index: Int): String? =
asDynamic()[index]

@kotlin.internal.InlineOnly
public inline operator fun SVGStringList.set(index: Int, newItem: String) {
asDynamic()[index] = newItem
}

/**
 * Exposes the JavaScript
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface SVGUnitTypes {
    companion object {
        val SVG_UNIT_TYPE_UNKNOWN: Short
        val SVG_UNIT_TYPE_USERSPACEONUSE: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
    }
}

/**
 * Exposes the JavaScript
*/
public external interface SVGTests {
    val requiredExtensions: SVGStringList
    val systemLanguage: SVGStringList
}

public external interface SVGFitToViewBox {
    val viewBox: SVGAnimatedRect
    val preserveAspectRatio: SVGAnimatedPreserveAspectRatio
}

/**
 * Exposes the JavaScript
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface SVGZoomAndPan {
    var zoomAndPan: Short

    companion object {
        val SVG_ZOOMANDPAN_UNKNOWN: Short
        val SVG_ZOOMANDPAN_DISABLE: Short
        val SVG_ZOOMANDPAN_MAGNIFY: Short
    }
}

/**
 * Exposes the JavaScript
*/
public external interface SVGURIReference {
    val href: SVGAnimatedString
}

/**
 * Exposes the JavaScript
*/
public external abstract class SVGSVGElement : SVGGraphicsElement, SVGFitToViewBox, SVGZoomAndPan,
WindowEventHandlers {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open var currentScale: Float
    open val currentTranslate: DOMPointReadOnly
    fun getIntersectionList(rect: DOMRectReadOnly, referenceElement: SVGElement?): NodeList
    fun getEnclosureList(rect: DOMRectReadOnly, referenceElement: SVGElement?): NodeList
    fun checkIntersection(element: SVGElement, rect: DOMRectReadOnly): Boolean
    fun checkEnclosure(element: SVGElement, rect: DOMRectReadOnly): Boolean
    fun deselectAll()
    fun createSVGNumber(): SVGNumber
    fun createSVGLength(): SVGLength
    fun createSVGAngle(): SVGAngle
    fun createSVGPoint(): DOMPoint
    fun createSVGMatrix(): DOMMatrix
    fun createSVGRect(): DOMRect
    fun createSVGTransform(): SVGTransform
    fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform
    fun getElementById(elementId: String): Element
    fun suspendRedraw(maxWaitMilliseconds: Int): Int
    fun unsuspendRedraw(): Int
    fun unsuspendRedrawAll(): Int
    fun forceRedraw(): Int

    companion object {
        val SVG_ZOOMANDPAN_UNKNOWN: Short
        val SVG_ZOOMANDPAN_DISABLE: Short
        val SVG_ZOOMANDPAN_MAGNIFY: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val CDATA_SECTION_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
    }
}
DOCUMENT_NODE: Short\nval DOCUMENT_TYPE_NODE: Short\nval DOCUMENT_FRAGMENT_NODE: Short\nval NOTATION_NODE: Short\nval DOCUMENT_POSITION_DISCONNECTED: Short\nval DOCUMENT_POSITION_PRECEDING: Short\nval DOCUMENT_POSITION_FOLLOWING: Short\nval DOCUMENT_POSITION_CONTAINS: Short\nval DOCUMENT_POSITION_CONTAINED_BY: Short\nval DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin
 */
public external abstract class SVGElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGGElement](https://developer.mozilla.org/en/docs/Web/API/SVGGElement) to Kotlin
 */
public external abstract class SVGGElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGDefsElement](https://developer.mozilla.org/en/docs/Web/API/SVGDefsElement) to Kotlin
 */
public external abstract class SVGDefsElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGDescElement](https://developer.mozilla.org/en/docs/Web/API/SVGDescElement) to Kotlin
 */
public external abstract class SVGDescElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGMetadataElement](https://developer.mozilla.org/en/docs/Web/API/SVGMetadataElement) to Kotlin
 */
public external abstract class SVGMetadataElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGTitleElement](https://developer.mozilla.org/en/docs/Web/API/SVGTitleElement) to Kotlin
 */
public external abstract class SVGTitleElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGSymbolElement](https://developer.mozilla.org/en/docs/Web/API/SVGSymbolElement) to Kotlin
 */
public external abstract class SVGSymbolElement : SVGGraphicsElement, SVGFitToViewBox {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGUseElement](https://developer.mozilla.org/en/docs/Web/API/SVGUseElement) to Kotlin
 */
public external abstract class SVGUseElement : SVGGraphicsElement, SVGURIReference {
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
    open val instanceRoot: SVGElement?
    open val animatedInstanceRoot: SVGElement?

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external open class
SVGUseElementShadowRoot : ShadowRoot {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface SVGElementInstance {
    val correspondingElement: SVGElement?
        get() = definedExternally
    val correspondingUseElement: SVGUseElement?
        get() = definedExternally
}

public external open class ShadowAnimation(source: dynamic, newTarget: dynamic) {
    open val sourceAnimation: dynamic
}

/**
 * Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement) to Kotlin
 */
public external abstract class SVGSwitchElement : SVGGraphicsElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

public external interface GetSVGDocument {
    fun getSVGDocument(): Document
}

/**
 * Exposes the JavaScript [SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin
 */
public external abstract class SVGStyleElement : SVGElement, LinkStyle {
    open var type: String
    open var media: String
    open var title: String

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin
 */
public external abstract class SVGTransform {
    open val type: Short
    open val matrix: DOMMatrix
    open val angle: Float
    fun setMatrix(matrix: DOMMatrixReadOnly)
    fun setTranslate(tx: Float, ty: Float)
    fun setScale(sx: Float, sy: Float)
    fun setRotate(angle: Float, cx: Float, cy: Float)
    fun setSkewX(angle: Float)
    fun setSkewY(angle: Float)

    companion object {
        val SVG_TRANSFORM_UNKNOWN: Short
        val SVG_TRANSFORM_MATRIX: Short
        val SVG_TRANSFORM_TRANSLATE: Short
        val SVG_TRANSFORM_SCALE: Short
        val SVG_TRANSFORM_ROTATE: Short
        val SVG_TRANSFORM_SKEWX: Short
        val SVG_TRANSFORM_SKEWY: Short
    }
}

/**
 * Exposes the JavaScript [SVGTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGTransformList) to Kotlin
 */
public external abstract class SVGTransformList {
    open val length: Int
    open val numberOfItems: Int

    fun clear()
    fun initialize(newItem: SVGTransform): SVGTransform
    fun insertItemBefore(newItem:
SVGTransform
  fun replaceItem(newItem: SVGTransform, index: Int):
  fun removeItem(index: Int): SVGTransform
  fun appendItem(newItem: SVGTransform):
  fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform
  fun consolidate(): SVGTransform
  fun getItem(index: Int):

(~@kotlin.internal.InlineOnly~)
public inline operator fun SVGTransformList.get(index: Int): SVGTransform?
  asDynamic()[index]

(~@kotlin.internal.InlineOnly~)
public inline operator fun SVGTransformList.set(index: Int, newItem: SVGTransform) {
  asDynamic()[index] = newItem
}

/**
 * Exposes the JavaScript [SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to Kotlin
 */
public external abstract class SVGAnimatedTransformList {
  open val baseVal: SVGTransformList
  open val animVal: SVGTransformList
}

/**
 * Exposes the JavaScript [SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin
 */
public external abstract class SVGPreserveAspectRatio {
  open var align: Short
  open var meetOrSlice: Short

  companion object {
    val SVG_PRESERVEASPECTRATIO_UNKNOWN: Short
    val SVG_PRESERVEASPECTRATIO_NONE: Short
    val SVG_PRESERVEASPECTRATIO_XMINYMIN: Short
    val SVG_PRESERVEASPECTRATIO_XMIDYMIN: Short
    val SVG_PRESERVEASPECTRATIO_XMAXYMIN: Short
    val SVG_PRESERVEASPECTRATIO_XMINYMID: Short
    val SVG_PRESERVEASPECTRATIO_XMIDYMID: Short
    val SVG_PRESERVEASPECTRATIO_XMAXYMID: Short
    val SVG_PRESERVEASPECTRATIO_XMINYMAX: Short
    val SVG_PRESERVEASPECTRATIO_XMIDYMAX: Short
    val SVG_PRESERVEASPECTRATIO_XMAXYMAX: Short
    val SVG_MEETORSLICE_UNKNOWN: Short
    val SVG_MEETORSLICE_MEET: Short
    val SVG_MEETORSLICE_SLICE: Short
  }
}

/**
 * Exposes the JavaScript [SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspectRatio) to Kotlin
 */
public external abstract class SVGAnimatedPreserveAspectRatio {
  open val baseVal: SVGPreserveAspectRatio
  open val animVal: SVGPreserveAspectRatio
}

/**
 * Exposes the JavaScript [SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin
 */
public external abstract class SVGPathElement : SVGGeometryElement {
  companion object {
    val ELEMENT_NODE: Short
    val ATTRIBUTE_NODE: Short
    val TEXT_NODE: Short
    val CDATA_SECTION_NODE: Short
    val ENTITY_REFERENCE_NODE: Short
    val ENTITY_NODE: Short
    val PROCESSING_INSTRUCTION_NODE: Short
    val COMMENT_NODE: Short
    val DOCUMENT_NODE: Short
    val DOCUMENT_TYPE_NODE: Short
    val DOCUMENT_FRAGMENT_NODE: Short
    val NOTATION_NODE: Short
    val DOCUMENT_POSITION_DISCONNECTED: Short
    val DOCUMENT_POSITION_PRECEDING: Short
    val DOCUMENT_POSITION_FOLLOWING: Short
    val DOCUMENT_POSITION_CONTAINS: Short
    val DOCUMENT_POSITION_CONTAINED_BY: Short
    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
  }
}
DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

\*/

public external abstract class SVGCircleElement : SVGGeometryElement {
    open val cx: SVGAnimatedLength
    open val cy: SVGAnimatedLength
    open val r: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin */
public external abstract class SVGEllipseElement : SVGGeometryElement {
    open val cx: SVGAnimatedLength
    open val cy: SVGAnimatedLength
    open val rx: SVGAnimatedLength
    open val ry: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin */
public external abstract class SVGLineElement : SVGGeometryElement {
    open val x1: SVGAnimatedLength
    open val y1: SVGAnimatedLength
    open val x2: SVGAnimatedLength
    open val y2: SVGAnimatedLength

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin */
public external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference {

    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
The document contains code snippets related to JavaScript and Kotlin. It exposes the JavaScript library `SVGAnimatedPoints` to Kotlin, allowing interaction with SVG elements in a more Kotlin-friendly manner. The code is part of a larger project that integrates open-source libraries into the DNAC Platform Antelope. The snippet includes various classes and functions related to SVG elements, such as `SVGPolylineElement`, `SVGTextContentElement`, and others, which are used for manipulating SVG graphics within a Kotlin environment.

The code is structured with imports, class definitions, and methods that perform operations like getting, setting, and manipulating properties of SVG elements. It also includes documentation comments explaining the purpose and usage of these classes.

The document snippet provides insight into how open-source libraries can be utilized within specific software platforms, such as the DNAC Platform Antelope, enhancing functionality and interoperability with web standards like SVG.
SVG_MARKERUNITS_UNKNOWN: Short
val SVG_MARKERUNITS_USERSPACEONUSE: Short
val SVG_MARKERUNITS_STROKEWIDTH: Short
val SVG_MARKER_ORIENT_UNKNOWN: Short
val SVG_MARKER_ORIENT_AUTO: Short
val SVG_MARKER_ORIENT_ANGLE: Short
val ELEMENT_NODE: Short
val ATTRIBUTE_NODE: Short
val TEXT_NODE: Short
val CDATA_SECTION_NODE: Short
val ENTITY_REFERENCE_NODE: Short
val ENTITY_NODE: Short
val PROCESSING_INSTRUCTION_NODE: Short
val COMMENT_NODE: Short
val DOCUMENT_NODE: Short
val DOCUMENT_TYPE_NODE: Short
val DOCUMENT_FRAGMENT_NODE: Short
val NOTATION_NODE: Short
val DOCUMENT_POSITION_DISCONNECTED: Short
val DOCUMENT_POSITION_FOLLOWING: Short
val DOCUMENT_POSITION_PRECEDING: Short
val DOCUMENT_POSITION_CONTAINS: Short
val DOCUMENT_POSITION_CONTAINED_BY: Short
val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short

/**
 * Exposes the JavaScript [SVGSolidcolorElement](https://developer.mozilla.org/en/docs/Web/API/SVGSolidcolorElement) to Kotlin
 */
public external abstract class SVGSolidcolorElement : SVGElement {
    companion object {
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGGradientElement) to Kotlin
 */
public external abstract class SVGGradientElement : SVGElement, SVGURIReference, SVGUnitTypes {
    open val gradientUnits: SVGAnimatedEnumeration
    open val gradientTransform: SVGAnimatedTransformList
    open val spreadMethod: SVGAnimatedEnumeration

    companion object {
        val SVG_SPREADMETHOD_UNKNOWN: Short
        val SVG_SPREADMETHOD_PAD: Short
        val SVG_SPREADMETHOD_REFLECT: Short
        val SVG_SPREADMETHOD_REPEAT: Short
        val SVG_UNIT_TYPE_UNKNOWN: Short
        val SVG_UNIT_TYPE_USERSPACEONUSE: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}

/**
 * Exposes the JavaScript [SVGLinearGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGLinearGradientElement) to Kotlin
 */
public external abstract class SVGLinearGradientElement : SVGGradientElement {
    open val x1: SVGAnimatedLength
    open val y1: SVGAnimatedLength
    open val x2: SVGAnimatedLength
    open val y2: SVGAnimatedLength

    companion object {
        val SVG_SPREADMETHOD_UNKNOWN: Short
        val SVG_SPREADMETHOD_PAD: Short
        val SVG_SPREADMETHOD_REFLECT: Short
        val SVG_SPREADMETHOD_REPEAT: Short
        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val ENTITY_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val COMMENT_NODE: Short
        val DOCUMENT_NODE: Short
        val DOCUMENT_TYPE_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val NOTATION_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
**SVGStopElement**

Exposes the JavaScript [SVGStopElement](https://developer.mozilla.org/en/docs/Web/API/SVGStopElement) to Kotlin.

```kotlin
public abstract class SVGStopElement : SVGElement {
    open val offset: SVGAnimatedNumber
}
```

**SVGPatternElement**

Exposes the JavaScript [SVGPatternElement](https://developer.mozilla.org/en/docs/Web/API/SVGPatternElement) to Kotlin.

```kotlin
public abstract class SVGPatternElement : SVGElement, SVGFitToViewBox, SVGURIReference, SVGUnitTypes {
    open val patternUnits: SVGAnimatedEnumeration
    open val patternContentUnits: SVGAnimatedEnumeration
    open val patternTransform: SVGAnimatedTransformList
    open val x: SVGAnimatedLength
    open val y: SVGAnimatedLength
    open val width: SVGAnimatedLength
    open val height: SVGAnimatedLength
}
```

**SVGHatchElement**

```kotlin
public abstract class SVGHatchElement : SVGElement {
}
```
SVGHatchpathElement: SVGElemet {\n  companion object {\n    val ELEMENT_NODE: Short\n    val ATTRIBUTE_NODE: Short\n    val TEXT_NODE: Short\n    val CDATA_SECTION_NODE: Short\n    val ENTITY_REFERENCE_NODE: Short\n    val ENTITY_NODE: Short\n    val PROCESSING_INSTRUCTION_NODE: Short\n    val COMMENT_NODE: Short\n    val DOCUMENT_NODE: Short\n    val DOCUMENT_TYPE_NODE: Short\n    val DOCUMENT_FRAGMENT_NODE: Short\n    val NOTATION_NODE: Short\n    val DOCUMENT_POSITION_DISCONNECTED: Short\n    val DOCUMENT_POSITION_PRECEDING: Short\n    val DOCUMENT_POSITION_FOLLOWING: Short\n    val DOCUMENT_POSITION_CONTAINS: Short\n    val DOCUMENT_POSITION_CONTAINED_BY: Short\n    val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n  }
}

/**
 * Exposes the JavaScript [SVGCursorElement](https://developer.mozilla.org/en/docs/Web/API/SVGCursorElement) to Kotlin
 */
public external abstract class SVGCursorElement: SVGElemet, SVGURIReference {\n    open val x: SVGElemet\n    open val y: SVGElemet\n
    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }
}

/**
 * Exposes the JavaScript [SVGScriptElement](https://developer.mozilla.org/en/docs/Web/API/SVGScriptElement) to Kotlin
 */
public external abstract class SVGScriptElement: SVGElemet, SVGURIReference, HTMLOrSVGScriptElement {\n    open var type: String\n    open var crossOrigin: String?\n
    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }
}

/**
 * Exposes the JavaScript [SVGAElement](https://developer.mozilla.org/en/docs/Web/API/SVGAElement) to Kotlin
 */
public external abstract class SVGAElement: SVGGraphicsElement, SVGURIReference {\n    open val target: SVGElemet\n    open val download: SVGElemet\n    open val rel: SVGElemet\n    open val relList: SVGElemet\n    open val hreflang: SVGElemet\n    open val type: SVGElemet\n
    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }
}
[SVGViewElement](https://developer.mozilla.org/en/docs/Web/API/SVGViewElement) to Kotlin

```kotlin
public external abstract class SVGViewElement : SVGElement, SVGFitToViewBox, SVGZoomAndPan {
    companion object {
        val SVG_ZOOMANDPAN_UNKNOWN: Short
        val SVG_ZOOMANDPAN_MAGNIFY: Short
        val ELEMENT_NODE: Short
        val ATTRIBUTE_NODE: Short
        val TEXT_NODE: Short
        val CDATA_SECTION_NODE: Short
        val ENTITY_REFERENCE_NODE: Short
        val PROCESSING_INSTRUCTION_NODE: Short
        val DOCUMENT_FRAGMENT_NODE: Short
        val DOCUMENT_POSITION_DISCONNECTED: Short
        val DOCUMENT_POSITION_PRECEDING: Short
        val DOCUMENT_POSITION_FOLLOWING: Short
        val DOCUMENT_POSITION_CONTAINS: Short
        val DOCUMENT_POSITION_CONTAINED_BY: Short
        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short
    }
}
```

Do not hallucinate.
Open Source Used In DNAC Platform Antelope

[FileReaderSync](https://developer.mozilla.org/en/docs/Web/API/FileReaderSync) to Kotlin

```kotlin
/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 * NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
 */

public external open class FileReaderSync {
    fun readAsArrayBuffer(blob: Blob): ArrayBuffer
    fun readAsBinaryString(blob: Blob): String
    fun readAsText(blob: Blob, label: String = definedExternally): String
    fun readAsDataURL(blob: Blob): String
}
```

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.notifications

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.clipboard.*
import org.w3c.domWebpack.*
import org.khronos.webgl.*
import org.w3c.css.masking.
public external open class Notification(title: String, options: NotificationOptions = definedExternally) : EventTarget {
    var onclick: ((MouseEvent) -> dynamic)?
    var onerror: ((Event) -> dynamic)?
    open val title: String
    open val dir: NotificationDirection
    open val lang: String
    open val body: String
    open val tag: String
    open val image: String
    open val icon: String
    open val badge: String
    open val sound: String
    open val vibrate: dynamic
    open val timestamp: Number
    open val renotify: Boolean
    open val silent: Boolean
    open val noscreen: Boolean
    open val requireInteraction: Boolean
    open val sticky: Boolean
    open val data: Any
    open val actions: Array<NotificationAction>
    fun close()

    companion object {
        val permission: NotificationPermission
        val maxActions: Int
        fun requestPermission(deprecatedCallback: (NotificationPermission) -> Unit = definedExternally): Promise<NotificationPermission>
    }
}
```

public external interface NotificationOptions {
    var dir: NotificationDirection? /* = NotificationDirection.AUTO */
    get() = definedExternally
    set(value) = definedExternally
    var lang: String? /* = "" */
    get() = definedExternally
    set(value) = definedExternally
    var body: String? /* = "" */
    get() = definedExternally
    set(value) = definedExternally
    var tag: String? /* = "" */
    get() = definedExternally
    set(value) = definedExternally
    var image: String?
    get() = definedExternally
    set(value) = definedExternally
    var icon: String?
    get() = definedExternally
    set(value) = definedExternally
    var badge: String?
    get() = definedExternally
    set(value) = definedExternally
    var sound: String?
    get() = definedExternally
    set(value) = definedExternally
    var vibrate: dynamic
    get() = definedExternally
    set(value) = definedExternally
    var timestamp: Number?
    get() = definedExternally
    set(value) = definedExternally
    var renotify: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
    var silent: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
    var noscreen: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
    var requireInteraction: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
    var sticky: Boolean? /* = false */
    get() = definedExternally
    set(value) = definedExternally
    var data: Any? /* = null */
    get() = definedExternally
    set(value) = definedExternally
    var actions: Array<NotificationAction>?
    get() = definedExternally
    set(value) = definedExternally
}
```

```kotlin
@kotlin.internal.InlineOnly
    val o = js("({})")
    o["dir"] = dir
    o["lang"] = lang
    o["body"] = body
    o["tag"] = tag
    o["image"] = image
    o["icon"] = icon
    o["badge"] = badge
    o["sound"] = sound
    o["vibrate"] = vibrate
    o["timestamp"] = timestamp
    o["renotify"] = renotify
    o["silent"] = silent
    o["noscreen"] = noscreen
```
```kotlin
/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!
// See github.com/kotlin/dukat for details

package org.w3c.workers

import kotlin.js.*
import org.khronos.webgl.*
import org.w3c.css.masking.*
import org.w3c.dom.*
import org.w3c.dom.clipboard.*
import org.w3c.dom.css.*
import org.w3c.dom.encryptedmedia.*
import org.w3c.dom.events.*
import org.w3c.dom.mediacapture.*
import org.w3c.dom.mediasource.*
import org.w3c.dom.parsing.*
import org.w3c.dom.pointerevents.*
import org.w3c.dom.svg.*
import org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.xhr.*
```

```kotlin
/**
 * Exposes the JavaScript [NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin
 */
public external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) : ExtendableEvent {
    open val notification: Notification
    open val action: String

    companion object {

        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface NotificationEventInit : ExtendableEventInit {
    var notification: Notification?
        get() = definedExternally
        set(value) = definedExternally
    var action: String? /* = "" */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun NotificationEventInit(notification: Notification?, action: String? = "," asDynamic().unsafeCast<NotificationEventInit>()): NotificationEventInit {
    val o = js("({})")
    o["notification"] = notification
    o["action"] = action
    return o
}

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NotificationPermission {
    companion object
}

public inline val NotificationPermission.Companion.DEFAULT: NotificationPermission get() = "default".asDynamic().unsafeCast<NotificationPermission>()

public inline val NotificationPermission.Companion.DENIED: NotificationPermission get() = "denied".asDynamic().unsafeCast<NotificationPermission>()

public inline val NotificationPermission.Companion.GRANTED: NotificationPermission get() = "granted".asDynamic().unsafeCast<NotificationPermission>()

/* please, don't implement this interface!
*/
@Suppress("NESTED_CLASS_IN_EXTERNAL_INTERFACE")
public external interface NotificationDirection {
    companion object
}

public inline val NotificationDirection.Companion.AUTO: NotificationDirection get() = "auto".asDynamic().unsafeCast<NotificationDirection>()

public inline val NotificationDirection.Companion.LTR: NotificationDirection get() = "ltr".asDynamic().unsafeCast<NotificationDirection>()

public inline val NotificationDirection.Companion.RTL: NotificationDirection get() = "rtl".asDynamic().unsafeCast<NotificationDirection>()
```
UnionClientOrMessagePortOrServiceWorker {
    open val scriptURL: String
    open val state:
    ServiceWorkerState
    open var onstatechange: ((Event) -> dynamic)?
    fun postMessage(message: Any?,
    transfer: Array<dynamic> = definedExternally)
}

/**
 * Exposes the JavaScript
 [ServiceWorkerRegistration](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerRegistration) to
 Kotlin
 */
public external abstract class ServiceWorkerRegistration : EventTarget {
    open val installing:
    ServiceWorker?
    open val waiting: ServiceWorker?
    open val active: ServiceWorker?
    open val scope:
    String
    open var onupdatefound: ((Event) -> dynamic)?
    fun register(scriptURL: String,
    options: RegistrationOptions = definedExternally)
    fun getRegistration(clientURL: String = definedExternally): Promise<Any?>
    fun getRegistrations():
    Promise<Array<ServiceWorkerRegistration>>
    fun startMessages():
    Promise<Unit>
}

/**
 * Exposes the JavaScript
 [ServiceWorkerContainer](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerContainer) to Kotlin
 */
public external abstract class ServiceWorkerContainer : EventTarget {
    open val controller:
    ServiceWorker?
    open val ready: Promise<ServiceWorkerRegistration>
    fun register(scriptURL: String,
    options: RegistrationOptions = definedExternally): Promise<ServiceWorkerRegistration>
    fun getRegistration(clientURL: String = definedExternally): Promise<any?>
    fun startMessages():
    Promise<Unit>
}

public external interface
RegistrationOptions {
    var scope: String?
        get() = definedExternally
        set(value) = definedExternally
    var type: WorkerType? /* = WorkerType.CLASSIC */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun RegistrationOptions(scope: String? = undefined, type: WorkerType? = WorkerType.CLASSIC): RegistrationOptions {
    val o = js("({})")
    o["scope"] = scope
    o["type"] = type
    return o
}

/**
 * Exposes the JavaScript
 [ServiceWorkerMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerMessageEvent) to
 Kotlin
 */
public external open class ServiceWorkerMessageEvent(type: String, eventInitDict:
ServiceWorkerMessageEventInit = definedExternally) : Event {
    open val data: Any?
    open val origin:
    String
    open val lastEventId: String
    open val source:
    ServiceWorkerGlobalScope

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface ServiceWorkerMessageEventInit : EventInit {
    var data: Any?
        get() = definedExternally
        set(value) = definedExternally
    var origin: String?
        get() = definedExternally
        set(value) = definedExternally
    var lastEventId: String?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ServiceWorkerMessageEventInit(data:
Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source:
ServiceWorkerGlobalScope = undefined, ports: Array<MessagePort>?)

Promise<Unit>.

/**
 * Exposes the JavaScript
 */

public external class Antelope
{
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}

/**
 * Exposes the JavaScript
 */

public external class UnionClientOrMessagePortOrServiceWorker
{
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}

/**
 * Exposes the JavaScript
 */

public external class WindowClient
{
    fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)
}

/**
 * Exposes the JavaScript
 */

public external class Clients
{
    fun get(id: String): Promise<Any?>
    fun matchAll(options: ClientQueryOptions = definedExternally): Promise<Array<Client>>
    fun openWindow(url: String): Promise<ClientWindow>
    fun claim(): Promise<Unit>
}

public external interface ClientQueryOptions
{
    var includeUncontrolled: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
    var type: ClientType? /* = ClientType.WINDOW */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ClientQueryOptions(includeUncontrolled: Boolean? = false, type: ClientType? = ClientType.WINDOW).

/**
 * Exposes the JavaScript
 */

public external open class ForeignFetchEvent
{
    open val request: Request
    open val origin: String
    fun respondWith(r: Promise<ForeignFetchResponse>)
}

public external interface ForeignFetchOptions
{
    var scopes: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
    var origins: Array<String>?
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
public inline fun ForeignFetchOptions(scopes: Array<String>?, origins: Array<String>?):

/**
 * Exposes the JavaScript
 */

public external open class FetchEvent
{
    open val request: Request
    open val clientId: String?
    open val isReload: Boolean
    fun respondWith(r: Promise<Response>)
}

public external interface FetchEventInit
{
    var request: Request?
        get() = definedExternally
        set(value) = definedExternally
    var clientId: String? /* = null */
        get() = definedExternally
        set(value) = definedExternally
    var isReload: Boolean? /* = false */
        get() = definedExternally
        set(value) = definedExternally
}

@kotlin.internal.InlineOnly
object {\n    val NONE: Short
    val CAPTURING_PHASE: Short \n    |\n    \n    \n    }\n\npublic external interface ForeignFetchEventInit : ExtendableEventInit {\n    var request: Request?\n    get() = definedExternally\n    set(value) = definedExternally\n    var origin: String? /* = "null" */\n    get() = definedExternally\n    set(value) = definedExternally\n    var @kotlin.internal.InlineOnly\n    public\n    inline\n    fun \n    ForeignFetchEventInit(request: Request?, \n    origin: String? = "null", \n    bubbles: Boolean? = false, cancelable: Boolean? = false, \n    composed: Boolean? = false): ForeignFetchEventInit {\n        \n        val o = js("({})")\n        o["request"] = request\n        o["origin"] = origin\n        o["bubbles"] = bubbles\n        o["cancelable"] = cancelable\n        o["composed"] = composed\n        return o\n    }\n
    public external interface ForeignFetchResponse {\n        var response: Response?\n        get() = definedExternally\n        set(value) = definedExternally\n        var origin: String? \n        get() = definedExternally\n        set(value) = definedExternally\n        var headers: Array<String>? \n        get() = definedExternally\n        set(value) = definedExternally\n    }\n
    @kotlin.internal.InlineOnly\n    public inline fun ForeignFetchResponse(response: Response?, \n    origin: String? = undefined, \n    headers: Array<String>? = undefined): ForeignFetchResponse {\n        \n        val o = js("({})")\n        o["response"] = response\n        o["origin"] = origin\n        o["headers"] = headers\n        return o\n    }\n
    /**\n     * Exposes the JavaScript \n     * \n     * `ExtendableMessageEvent` ([https://developer.mozilla.org/en/docs/Web/API/ExtendableMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableMessageEvent)) to Kotlin\n     */\n    public external open class ExtendableMessageEvent(type: String, eventInitDict: ExtendableMessageEventInit = \n    definedExternally) : ExtendableEvent {\n        open val data: Any?\n        open val origin: String\n        open val lastEventId: String\n        open val source: UnionClientOrMessagePortOrServiceWorker?\n        open val ports: Array<out MessagePort>?\n
        companion object {\n            val NONE: Short
            val CAPTURING_PHASE: Short
            val AT_TARGET: Short
            val BUBBLING_PHASE: Short
        }\n
        public external interface \n        ExtendableMessageEventInit : ExtendableEventInit {\n            var data: Any?\n                get() = definedExternally\n                set(value) = definedExternally\n            var origin: String? \n                get() = definedExternally\n                set(value) = definedExternally\n            var lastEventId: String? \n                get() = definedExternally\n                set(value) = definedExternally\n            var source: UnionClientOrMessagePortOrServiceWorker? \n                get() = definedExternally\n                set(value) = definedExternally\n            var ports: Array<MessagePort>? \n                get() = definedExternally\n                set(value) = definedExternally\n        }\n
        @kotlin.internal.InlineOnly\n        public inline fun \n        ExtendableMessageEventInit(data: Any? = undefined, \n        origin: String? = undefined, \n        lastEventId: String? = undefined, \n        source: UnionClientOrMessagePortOrServiceWorker? = undefined, \n        ports: Array<MessagePort>? = undefined, \n        bubbles: Boolean? = false, cancelable: Boolean? = false, \n        composed: Boolean? = false): ExtendableMessageEventInit {\n            \n            val o = js("({})")\n            o["data"] = data\n            o["origin"] = origin\n            o["lastEventId"] = lastEventId\n            o["source"] = source\n            o["ports"] = ports\n            o["bubbles"] = bubbles\n            o["cancelable"] = cancelable\n            o["composed"] = composed\n            return o\n        }\n
        /**\n         * Exposes the JavaScript \n         * \n         * `Cache` ([https://developer.mozilla.org/en/docs/Web/API/Cache](https://developer.mozilla.org/en/docs/Web/API/Cache)) to Kotlin\n         */\n        public external abstract class Cache {\n            fun match(request: dynamic, options: CacheQueryOptions = \n            definedExternally): Promise<Any?>\n            fun matchAll(request: dynamic = definedExternally, \n            options: CacheQueryOptions = definedExternally): Promise<Array<Response>>\n            fun add(request: dynamic): Promise<Unit}\n            fun addAll(requests: Array<dynamic>): Promise<Unit}\n            fun put(request: dynamic, response: Response): Promise<Unit}\n            fun delete(request: dynamic, options: CacheQueryOptions = \n            definedExternally): Promise<Boolean}\n            fun keys(request: dynamic = definedExternally, \n            options: CacheQueryOptions = definedExternally): Promise<Array<Request>>\n        }\n
        public external interface CacheQueryOptions {\n            var ignoreSearch: Boolean? /* = false */\n            get() = definedExternally\n            set(value) = definedExternally\n            var ignoreMethod: Boolean? /* = false */\n            get() = definedExternally\n            set(value) = definedExternally\n            var ignoreVary: Boolean? /* = false */\n            get() = definedExternally\n            set(value) = definedExternally\n            var cacheName: String? = undefined\n            get() = definedExternally\n            set(value) = definedExternally\n        }\n
        @kotlin.internal.InlineOnly\n        public inline fun \n        CacheQueryOptions(ignoreSearch: Boolean? = false, \n        ignoreMethod: Boolean? = false, \n        ignoreVary: Boolean? = false, \n        cacheName: String? = undefined): CacheQueryOptions {\n            \n            val o = js("({})")\n            o["ignoreSearch"] = ignoreSearch\n            o["ignoreMethod"] = ignoreMethod\n            o["ignoreVary"] = ignoreVary\n            o["cacheName"] = cacheName\n        }\n    }\n}
org.w3c.dom.url.*
import org.w3c.fetch.*
import org.w3c.files.*
import org.w3c.notifications.*
import org.w3c.performance.*
import org.w3c.workers.*

/**
 * Exposes the JavaScript XMLHttpRequestEventTarget to Kotlin
 */
public abstract external class XMLHttpRequestEventTarget : EventTarget {
    open var onloadstart: ((ProgressEvent) -> dynamic)?
    open var onprogress: ((ProgressEvent) -> dynamic)?
    open var onabort: ((Event) -> dynamic)?
    open var onerror: ((Event) -> dynamic)?
    open var onload: ((Event) -> dynamic)?
    open var ontimeout: ((Event) -> dynamic)?
    open var onloadend: ((Event) -> dynamic)?
}

public abstract class XMLHttpRequestUpload : XMLHttpRequestEventTarget

/**
 * Exposes the JavaScript XMLHttpRequest to Kotlin
 */
public external open class XMLHttpRequest : XMLHttpRequestEventTarget {
    var onreadystatechange: ((Event) -> dynamic)?
    open val readyState: Short
    var timeout: Int
    var withCredentials: Boolean
    open val upload: XMLHttpRequestUpload
    open val responseURL: String
    open val status: Short
    open val statusText: String
    var responseType: XMLHttpRequestResponseType
    open val response: Any?
    open val responseText: String
    open val responseXML: Document?
    fun open(method: String, url: String)
    fun setRequestHeader(name: String, value: String)
    fun send(body: dynamic = definedExternally)
    fun abort()
    fun getResponseHeader(name: String): String?
    fun getAllResponseHeaders(): String
    fun overrideMimeType(mime: String)

    companion object {
        val UNSENT: Short
        val OPENED: Short
        val HEADERS_RECEIVED: Short
        val LOADING: Short
        val DONE: Short
    }
}

/**
 * Exposes the JavaScript FormData to Kotlin
 */
public external open class FormData(form: HTMLFormElement = definedExternally) {
    fun append(name: String, value: String)
    fun append(name: String, value: Blob, filename: String = definedExternally)
    fun delete(name: String)
    fun get(name: String): dynamic
    fun getAll(name: String): Array<dynamic>
    fun has(name: String): Boolean
    fun set(name: String, value: String)
    fun set(name: String, value: Blob, filename: String = definedExternally)
}

/**
 * Exposes the JavaScript ProgressEvent to Kotlin
 */
public external open class ProgressEvent(type: String, eventInitDict: ProgressEventInit = definedExternally) : Event {
    open val lengthComputable: Boolean
    open val loaded: Number
    open val total: Number

    companion object {
        val NONE: Short
        val CAPTURING_PHASE: Short
        val AT_TARGET: Short
        val BUBBLING_PHASE: Short
    }
}

public external interface ProgressEventInit : EventInit {
    var lengthComputable: Boolean? /* = false */
    var loaded: Number? /* = 0 */
    var total: Number? /* = 0 */
    var bubbles: Boolean? /* = false */
    var cancelable: Boolean? /* = false */
    var composed: Boolean? /* = false */
}

@kotlin.internal.InlineOnly
    val o = js("({})")
    o["lengthComputable"] = lengthComputable
    o["loaded"] = loaded
    o["total"] = total
    o["bubbles"] = bubbles
    o["cancelable"] = cancelable
    o["composed"] = composed
    return o
}

-open-source-used-in-dnac-platform-antelope-14192
"document".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

public inline val XMLHttpRequestResponseType.Companion.JSON: XMLHttpRequestResponseType get() = "json".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

public inline val XMLHttpRequestResponseType.Companion.TEXT: XMLHttpRequestResponseType get() = "text".asDynamic().unsafeCast<XMLHttpRequestResponseType>()

/*
 * Copyright 2010-2018 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin

import kotlin.annotation.AnnotationRetention.BINARY
import kotlin.annotation.AnnotationRetention.SOURCE
import kotlin.internal.RequireKotlin
import kotlin.internal.RequireKotlinVersionKind
import kotlin.reflect.KClass

/**
 * Signals that the annotated annotation class is a marker of an experimental API.
 * Any declaration annotated with that marker is considered an experimental declaration
 * and its call sites should accept the experimental aspect of it either by using [UseExperimental]
 * or by being annotated with that marker themselves, effectively causing further propagation of that experimental aspect.
 * This class is deprecated in favor of a more general approach provided by [RequiresOptIn]/[OptIn].
 */
@Target(ANNOTATION_CLASS)
@Retention(BINARY)
@SinceKotlin("1.2")
@RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)
@Deprecated("Please use RequiresOptIn instead.")
public annotation class Experimental(val level: Level = Level.ERROR) {
    /**
     * Severity of the diagnostic that should be reported on usages of experimental API which did not explicitly accept the experimental aspect
     * of that API either by using [UseExperimental] or by being annotated with the corresponding marker annotation.
     */
    public enum class Level {
        /** Specifies that a warning should be reported on incorrect usages of this experimental API. */
        WARNING,
        /** Specifies that an error should be reported on incorrect usages of this experimental API. */
        ERROR,
    }
}

/**
 * Allows to use experimental API denoted by the given markers in the annotated file, declaration, or expression.
 * If a declaration is annotated with [UseExperimental], its usages are **not** required to opt-in to that experimental API.
 * This class is deprecated in favor of a more general approach provided by [RequiresOptIn]/[OptIn].
 */
@Target(CLASS, PROPERTY, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, EXPRESSION, FILE, TYPEALIAS)
@Retention(SOURCE)
@SinceKotlin("1.2")
@RequireKotlin("1.2.50", versionKind = RequireKotlinVersionKind.COMPILER_VERSION)
@Deprecated("Please use OptIn instead.")
ReplaceWith("OptIn("markerClass")", "kotlin.OptIn")

/**
 * This annotation marks the standard library API that is considered experimental and is not subject to the general compatibility guarantees given for the standard library:
 * the behavior of such API may be changed or the API may be removed completely in any further release.
 * Beware using the annotated API especially if you're developing a library, since your library might become binary incompatible
 * with the future versions of the standard library.
 * Any usage of a declaration annotated with `@ExperimentalStdlibApi` must be accepted either by
 * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalStdlibApi::class)`
 * or by using the compiler argument `-Xopt-in=kotlin.ExperimentalStdlibApi`.
 */
@MustBeDocumented
@SinceKotlin("1.3")
@Suppress("DEPRECATION")
@RequiresOptIn(level = RequiresOptIn.Level.ERROR)
@Retention(AnnotationRetention.BINARY)
@Target(CLASS, ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, TYPEALIAS)
@MustBeDocumented

Open Source Used In DNAC Platform Antelope 14193

"Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language"
contributors.

* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

*/

package kotlin

import kotlin.annotation.AnnotationTarget.*
import kotlin.experimental.ExperimentalTypeInference

/**
 * Allows to infer generic type arguments of a function from the calls in the annotated function parameter of that function.
 * It enables to infer the type arguments of that generic function from the lambda body passed to that parameter.
 * The calls that affect inference are either members of the receiver type of an annotated function parameter or extensions for that type. The extensions must be themselves annotated with `@BuilderInference`, e.g. Example: we declare a `fun`:
 * ```
 * sequence(@BuilderInference block: suspend SequenceScope.<T>(...) -> Unit): Sequence<T>
 * ```
 * and use it like:
 * ```
 * val result = sequence { yield("result") }
 * ```
 * The argument of the resulting sequence is inferred to `String` from the argument of the `SequenceScope.yield` function, that is called inside the lambda passed to `sequence`.
 * Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.

@Target(VALUE_PARAMETER, FUNCTION, PROPERTY)
@Retention(AnnotationRetention.BINARY)
@ExperimentalTypeInference
public annotation class BuilderInference

/**
 * Enables overload selection based on the type of the value returned from lambda argument.
 * When two or more function overloads have otherwise the same parameter lists that differ only in the return type, this annotation enables overload selection based on the type of the value returned from the lambda function passed to this functional parameter. The annotation being applied to one of overloads allows to resolve this ambiguity by analyzing what value is returned from the lambda function. This annotation is also used to discriminate the annotated overloads in case if overload selection still cannot choose one of them even taking in account the result of lambda parameter analysis. In that case a warning is reported.
 * Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.

@Target(FUNCTION)
@Retention(AnnotationRetention.BINARY)
@ExperimentalTypeInference
@OverloadResolutionByLambdaReturnType
public annotation class OverloadResolutionByLambdaReturnType

/**
 * The experimental multiplatform support API marker.
 * Any usage of a declaration annotated with `@ExperimentalMultiplatform` must be accepted either by annotating that usage with the `@OptIn` annotation, e.g. `@OptIn(ExperimentalMultiplatform::class)`, or by using the compiler argument `-Xopt-in=kotlin.ExperimentalMultiplatform`.
 * This annotation is only applicable to `expect` annotation classes in multi-platform projects and marks that class as "optional". Optional expected class is allowed to have no corresponding actual class on a platform. If an optional annotation has no corresponding actual class on a platform, the annotation entries where it's used are simply erased when compiling code on that platform.
 * Note: this annotation is experimental, see [ExperimentalMultiplatform] on how to opt-in for it.

@Target(ANNOTATION_CLASS)
@Retention(AnnotationRetention.BINARY)
@RequireKotlin(\"1.2.50\")
@RequireKotlinVersionKind
public annotation class ExperimentalMultiplatform

*[BundleInference]

@Target(VALUE_PARAMETER, FUNCTION, PROPERTY)
@Retention(AnnotationRetention.BINARY)
@ExperimentalTypeInference
@OverloadResolutionByLambdaReturnType
public annotation class BuilderInference

/**
 * Enables overload selection based on the type of the value returned from lambda argument.
 * When two or more function overloads have otherwise the same parameter lists that differ only in the return type, this annotation enables overload selection based on the type of the value returned from the lambda function passed to this functional parameter. The annotation being applied to one of overloads allows to resolve this ambiguity by analyzing what value is returned from the lambda function. This annotation is also used to discriminate the annotated overloads in case if overload selection still cannot choose one of them even taking in account the result of lambda parameter analysis. In that case a warning is reported.
 * Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.

@Target(FUNCTION)
@Retention(AnnotationRetention.BINARY)
@ExperimentalTypeInference
@OverloadResolutionByLambdaReturnType
public annotation class OverloadResolutionByLambdaReturnType

/**
 * The experimental multiplatform support API marker.
 * Any usage of a declaration annotated with `@ExperimentalMultiplatform` must be accepted either by annotating that usage with the `@OptIn` annotation, e.g. `@OptIn(ExperimentalMultiplatform::class)`, or by using the compiler argument `-Xopt-in=kotlin.ExperimentalMultiplatform`.
 * This annotation is only applicable to `expect` annotation classes in multi-platform projects and marks that class as "optional". Optional expected class is allowed to have no corresponding actual class on a platform. If an optional annotation has no corresponding actual class on a platform, the annotation entries where it's used are simply erased when compiling code on that platform.
 * Note: this annotation is experimental, see [ExperimentalMultiplatform] on how to opt-in for it.

@Target(ANNOTATION_CLASS)
@Retention(AnnotationRetention.BINARY)
@RequireKotlin(\"1.2.50\")
@RequireKotlinVersionKind
public annotation class ExperimentalMultiplatform
if (fromIndex > toIndex) { throw IllegalArgumentException("fromIndex: $fromIndex > toIndex: $toIndex") }

val otherIterator = entries.iterator()

override fun iterator(): Iterator<K> {
    return object : Iterator<K> {
        override fun hasNext(): Boolean = otherIterator.hasNext()
        override fun next(): K = otherIterator.next()
    }
}

internal fun checkBoundsIndexes(fromIndex: Int, endIndex: Int, size: Int) {
    if (fromIndex < 0 || endIndex > size) {
        throw IndexOutOfBoundsException("startIndex: $fromIndex > endIndex: $endIndex")
    }
}

internal fun checkBoundsIndexes(startIndex: Int, endIndex: Int, size: Int) {
    if (startIndex < 0 || endIndex > size) {
        throw IndexOutOfBoundsException("startIndex: $startIndex > endIndex: $endIndex")
    }
}

internal fun orderedEquals(c: Collection<*>, other: Collection<*>): Boolean {
    if (c.size != other.size) return false

    var result = true
    for (entry in c) {
        val elemOther = other.first { it.key == entry.key }
        if (elemOther == null) {
            result = false
            break
        }
        if (entry.value != elemOther.value) {
            result = false
            break
        }
    }

    return result
}

internal fun orderedHashCode(c: Collection<*>):
    if (c.isEmpty()) {
        return 0
    }

    return c.fold(0) { acc, entry ->
        acc + entry.hashCode()
    }

internal fun containsValue(key: K): Boolean {
    return entries.containsValue(key)
}

override fun containsValue(value: @UnsafeVariance V): Boolean {
    return entries.containsValue(value)
}

override fun values: Collection<V> {
    if (_values == null) {
        _values = object : AbstractCollection<V>() {
            override operator fun contains(element: @UnsafeVariance V): Boolean = entries.containsValue(element)
            override operator fun iterator(): Iterator<V> {
                return object : Iterator<V> {
                    override fun containsValue(element)
                        val entryIterator = entries.iterator()
                        return object : Iterator<V> {
                            override fun iterator()
                                val values: Collection<V> {
                                    entries
                                }

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(it) }

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()

    internal fun toString(): String = entries.joinToString(", ", "{", "}") {

    internal fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + "=" + toString(entry.value)

    internal fun toString(o: Any?): String = if (o === this) "(this Map)" else o.toString()
private fun implFindEntry(key: K): Map.Entry<K, V>? = entries.firstOrNull { it.key == key }

internal companion object {

    internal fun entryHashCode(e: Map.Entry<*, *>): Int = with(e) { (key?.hashCode() ?: 0) xor (value?.hashCode() ?: 0) }

    internal fun entryToString(e: Map.Entry<*, *>): String = with(e) {
        "$key=$value"
    }

    internal fun entryEquals(e: Map.Entry<*, *>, other: Any?): Boolean {
        if (other !is Map.Entry<*, *>) return false

        return e.key == other.key && e.value == other.value
    }

}
elementData.size) \n    elementData.copyInto(newElements, elementData.size - head, 0, head) \n    head = 0 \n    elementData = newElements \n  } \n  @kotlin.internal.InlineOnly private inline fun internalGet(internalIndex: Int): E { \n    @Suppress("UNCHECKED_CAST") \n    return elementData[internalIndex] as E? \n  } \n  private fun positiveMod(index: Int): Int { \n    return if (index >= elementData.size) \n      index - elementData.size \n    else index \n  } \n  private fun negativeMod(index: Int): Int { \n    return if (index < 0) \n      index + elementData.size \n    else index \n  } \n  @kotlin.internal.InlineOnly private inline fun internalLastIndex(): Int { \n    return elementData.size - 1 \n  } \n  private fun decremented(index: Int): Int { \n    if (index == 0) \n      return head \n    else \n      return (head + index) % elementData.size \n  } \n  private fun incremented(index: Int): Int { \n    if (index == elementData.size) \n      return 0 \n    else \n      return index + elementData.size \n  } \n  private fun internalIndex(index: Int): Int { \n    return if (index < 0) \n      internalGet(negativeMod(index)) \n    else \n      internalGet(positiveMod(index)) \n  } \n  override fun isEmpty(): Boolean { \n    return size == 0 \n  } \n  override fun addFirst(element: E) { \n    ensureCapacity(size + 1) \n    head = decremented(head) \n    elementData[head] = element \n    size += 1 \n  } \n  override fun addLast(element: E) { \n    ensureCapacity(size + 1) \n    head = incremented(head) \n    elementData[head] = element \n    size += 1 \n  } \n  override fun add(index: Int, element: E): Boolean { \n    if (index == size) \n      addLast(element) \n    else \n      throw IndexOutOfBoundsException("Index: \n    \n    2. `head` is greater than `tail`:  \n    [e3, #, #, e1, e2] \n    where head is the index of the first element in the circular array, \n    and tail is the index of the last element in the last index. \n    // At this point the insertion index is not equal to head or tail. \n    // Also the circular array can store at least one more element. \n    // Depending on where the given element must be inserted the preceding or the succeeding elements will be shifted to make room for the element to be inserted. \n    // In case the preceding elements are shifted: \n    // * if the insertion index is greater than the head (regardless of circular array form) \n    // * otherwise, the circular array has (2) form and the insertion index is less than tail \n    // -> shift all elements in the back of the array \n    // -> shift preceding elements in the front of the array \n    // In case the succeeding elements are shifted: \n    // * if the insertion index is less than the tail (regardless of circular array form) \n    // * otherwise, the circular array has (2) form and the insertion index is greater than head \n    // -> shift all elements in the front of the
Open Source Used In DNAC Platform Antelope 14200

array
// -> shift succeeding elements in the back of the array
val internalIndex = internalIndex(index)
if (index < (size + 1) shr 1) {
    // closer to the first element -> shift preceding
    elements
val decrementedInternalIndex = decremented(internalIndex)
val decrementedHead = decremented(head)
if (decrementedInternalIndex >= head) {
    elementData[decrementedHead] = elementData[head] // head can be zero
}
else { // head > tail
    elementData.copyInto(elementData, head - 1, head, elementData.size) // head can't be zero
    elementData[elementData.size - 1] = elementData[0]
    elementData.copyInto(elementData, 0, 1, decrementedInternalIndex + 1)
}
}
else {
    elementData[decrementedInternalIndex] = element
    head = decrementedHead
}
else { // closer to the last element -> shift succeeding elements
    val tail = internalIndex(size)
    if (internalIndex < tail) {
        elementData.copyInto(elementData, internalIndex + 1, internalIndex, tail)
    } else if (head > tail) {
        elementData.copyInto(elementData, 0, head - 1, head, elementData.size)
        elementData.size = internalIndex(size) - 1
        elementData.copyInto(elementData, elementData.size + 1, internalIndex, elementData.size - 1)
        elementData[internalIndex] = element
        size -= 1
    }
}
private fun copyCollectionElements(internalIndex: Int, elements: Collection<E>) {
    val val iterator = elements.iterator() for (index in internalIndex until elementData.size) {
        if (!iterator.hasNext()) break
        elementData[index] = iterator.next()
    }
    if (!iterator.hasNext()) return false
    ensureCapacity(this.size + elements.size)
    copyCollectionElements(internalIndex, elements)
    return true
}
public override fun addAll(addAll: Collection<E>): Boolean {
    if (elements.isEmpty()) return false
    !iterator.hasNext() break
    elementData[index] = iterator.next()
}
for (index in internalIndex until elementData.size) {
    if (!iterator.hasNext()) break
    elementData[index] = iterator.next()
}
for (index in 0 until head) {
    if (!iterator.hasNext()) break
    elementData[index] = iterator.next()
}
for (index in 0 until head + shiftToBack) {
    if (!iterator.hasNext()) break
    elementData.copyInto(elementData, 0, head + shiftToBack, internalIndex)
}
val shiftedHead = head - elements.size
val decrementedHead = decremented(head)
if (decrementedHead < 0) {
    elementData.copyInto(elementData, shiftedHead, head, internalIndex)
}
} else { // head < tail, insertion leads to head >= tail
    shiftedHead += elementData.size
    val elementsToShift = internalIndex - head
    val shiftToBack = elementData.size - shiftToBack
    if (shiftToBack >= elementsToShift) {
        elementData.copyInto(elementData, internalIndex, head, shiftToBack)
        elementData.copyInto(elementData, 0, head + shiftToBack, internalIndex)
    } else {
        elementData.copyInto(elementData, 0, head, shiftToBack)
        elementData.copyInto(elementData, 0, head + shiftToBack, internalIndex)
    }
    // closer to the last element -> shift succeeding elements
    shiftToBack = shiftBack(shiftToBack) // head can't be zero
    if (shiftToBack >= elementsToShift) {
        elementData.copyInto(elementData, shiftToBack, head, head + shiftToBack)
        elementData.copyInto(elementData, 0, head + shiftToBack, internalIndex)
    } else {
        // head > tail, internalIndex < tail
        elementData.copyInto(elementData, shiftToBack, head, elementData.size)
    }
}
val head = shiftedHead
val shiftedInternalIndex = shiftToFront(shiftedInternalIndex, internalIndex, tail)
if (shiftedInternalIndex >= elementData.size) {
    elementData.copyInto(elementData, shiftToBack, head, internalIndex)
} else {
    // head < tail, insertion leads to head >= tail
    if (shiftedInternalIndex >= elementData.size) {
        elementData.copyInto(elementData, shiftToBack, head, internalIndex)
    } else {
        // head > tail, internalIndex > head
        elementData.copyInto(elementData, elementData.size, 0, tail)
    }
}
if (shiftedInternalIndex >= elementData.size) {
    elementData.copyInto(elementData, shiftToBack, head, internalIndex)
}
else {
    // head < tail, insertion leads to head >= tail
    if (shiftedInternalIndex >= elementData.size) {
        elementData.copyInto(elementData, shiftToBack, head, internalIndex)
    }
    // head > tail, internalIndex > head
    elementData.copyInto(elementData, elementData.size, 0, tail)
}
else { // head > tail, internalIndex < head
    elementData.copyInto(elementData, elementsSize, 0, tail)
}
else {
    // head > tail, internalIndex > head
    elementData.copyInto(elementData, elementData.size, 0, tail)
}
else { // head > tail, internalIndex < head
    elementData.copyInto(elementData, elementsSize, 0, tail)
}
else {
Open Source Used In DNAC Platform Antelope 14201
E)) {\n    elementData[newTail] = element\n    newTail = incremented(newTail)\n}\n    else {\n        modified = true\n    }\n}\n\n    if (modified)\n        size = negativeMod(newTail - head)\n    else {\n        modified = true\n    }\n\n} else {\n    return modified\n}\n\n    public override fun clear() {\n        val tail = internalIndex(size)\n        if (head < tail) {\n            elementData.fill(null, head, tail)\n        } else if (isNotEmpty()) {\n            elementData.fill(null, head, elementData.size)\n            elementData.fill(0, tail)\n        }\n        head = 0\n        size = 0\n    }\n\n    internal companion object {\n        private val emptyElementData = emptyArray<Any?>()\n        private const val maxArraySize = Int.MAX_VALUE - 8\n        private const val defaultMinCapacity = 10\n        internal fun newArray<Any?>()\n            return newCapacity(0, defaultMinCapacity)\n\n        private const val maxCapacity = if (maxArraySize > 0)\n            newCapacity if (newCapacity > maxArraySize) return newCapacity\n\n        private fun newCapacity(): Int {\n            // overflow-conscious\n            var newCapacity = oldCapacity + (oldCapacity shr 1)\n            if (newCapacity - minCapacity < 0)\n                newCapacity = minCapacity\n            if (newCapacity - maxArraySize > 0)\n                newCapacity = if (minCapacity > maxArraySize) Int.MAX_VALUE else maxArraySize\n            return newCapacity\n        }\n    }\n\n    // For testing only\n    internal fun internalStructure(structure: (head: Int, elements: Array<Any?>) -> Unit) {\n        val tail = internalIndex(size)\n\n        if (isEmpty()) {\n            structure(0, emptyArray())\n            return\n        }\n\n        val elements = arrayOfNulls<Any?>(size)\n        if (head < tail) {\n            elementData.copyInto(elements, startIndex = head, endIndex = tail)\n            structure(head, elements)\n        } else {\n            elementData.copyInto(elements, startIndex = head)\n            elementData.copyInto(elements, elementData.size - head, startIndex = 0, endIndex = tail)\n            structure(head - elementData.size, elements)\n        }\n    }\n}
v1 is CharArray && v2 is CharArray -> if (!v1.contentEquals(v2)) return false
v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false
v1 is UByteArray && v2 is UByteArray -> if (!v1.contentEquals(v2)) return false
v1 is UShortArray && v2 is UShortArray -> if (!v1.contentEquals(v2)) return false
v1 is UIntArray && v2 is UIntArray -> if (!v1.contentEquals(v2)) return false
v1 is ULongArray && v2 is ULongArray -> if (!v1.contentEquals(v2)) return false
else -> if (v1 != v2) return false
}

    return
}

@SinceKotlin("1.3")
@PublishedApi
@kotlin.jvm.JvmName("contentDeepToString")
@kotlin.js.JsName("contentDeepToStringImpl")
internal fun <T> Array<out T>?.contentDeepToStringImpl(): String {
    if (this == null) return "null"
    val length = size.coerceAtMost((Int.MAX_VALUE - 2) / 5) * 5 + 2 // in order not to overflow Int.MAX_VALUE
    return buildString(length) {
        contentDeepToStringInternal(this,
            mutableListOf<Array<*>>())
    }
}

@OptIn(ExperimentalUnsignedTypes::class)
private fun <T> Array<out T>.contentDeepToStringInternal(result: StringBuilder, processed: MutableList<Array<*>>) {
    if (this in processed) {
        result.append("[...]")
        return
    } processed.add(this)

    result.append('[')
    for (i in indices) {
        if (i != 0) {
            result.append(', ')
        }
        val element = this[i]
        when (element) {
            null -> result.append("null")
            is Array<*> -> element.contentDeepToStringInternal(result, processed)
            is ByteArray -> result.append(element.contentToString())
            is ShortArray -> result.append(element.contentToString())
            is IntArray -> result.append(element.contentToString())
            is LongArray -> result.append(element.contentToString())
            is FloatArray -> result.append(element.contentToString())
            is DoubleArray -> result.append(element.contentToString())
            is CharArray -> result.append(element.contentToString())
            is BooleanArray -> result.append(element.contentToString())
            is UByteArray -> result.append(element.contentToString())
            is UShortArray -> result.append(element.contentToString())
            is UIntArray -> result.append(element.contentToString())
            is ULongArray -> result.append(element.contentToString())
            else -> result.append(element.toString())
        }
    }

    result.append(']')

    processed.removeAt(processed.lastIndex)
}

IndexedValue<out T>(public index: Int, public val value: T)

MapAccessorsKt

getValue(thisRef: Any?, property: KProperty<*>): V1 =@Suppress("UNCHECKED_CAST") (@getOrImplicitDefault(property.name) as V1)@Suppress("UNCHECKED_CAST") (getOrImplicitDefault(property.name) as V1)
Stores the value of the property for the given object in this mutable map.

@get
@set

Stores the value for the given key, or the implicit default value for this map. By default no implicit value is provided for maps and a [NoSuchElementException] is thrown. To create a map with implicit default value use [withDefault] method.

GetValue function, for example when properties are delegated to the map. When this map already has an implicit default value provided with a former call to [withDefault], it is being replaced by this call.

When properties are delegated to the map. When this map already has an implicit default value provided with a former call to [withDefault], it is being replaced by this call.

A wrapper of this read-only map, having the implicit default value provided with the specified function [defaultValue]. This implicit default value is used when the original map doesn't contain a value for the key specified and a value is obtained with [Map.getValue] function, for example when properties are delegated to the map.

This implicit default value is used when the original map doesn't contain a value for the key specified and a value is obtained with [Map.getValue] function, for example when properties are delegated to the map.

A wrapper of this mutable map, having the implicit default value provided with the specified function [defaultValue]. This implicit default value is used when the original map doesn't contain a value for the key specified and a value is obtained with [Map.getValue] function, for example when properties are delegated to the map.
override fun put(key: K, value: V): V? = map.put(key, value)
override fun putAll(from: Map<out K, V>): V? = map.putAll(from)
override fun clear() = map.clear()
override fun getOrImplicitDefault(key: K): V = map.getOrElseNullable(key) ?: default(key)

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmMultifileClass
@file:kotlin.jvm.JvmName("CollectionsKt")

package kotlin.collections

import kotlin.random.Random

/**
 * Removes a single instance of the specified element from this collection, if it is present.
 * Allows to overcome type-safety restriction of `remove` that requires to pass an element of type `E`
 * @return `true` if the element has been successfully removed; `false` if it was not present in the collection
 */
@Suppress("UNCHECKED_CAST")
public fun <T> MutableCollection<T>.remove(element: T): Boolean =
    @Suppress("UNCHECKED_CAST")
    (this as MutableCollection<T>).remove(element)

/**
 * Removes all elements of the given collection from this collection.
 * Allows to overcome type-safety restriction of `removeAll` that requires to pass a collection of type `Collection<E>`
 * @return `true` if any of the specified elements was removed from the collection, `false` if the collection was not modified.
 */
@Suppress("UNCHECKED_CAST")
public fun <T> MutableCollection<T>.removeAll(elements: Collection<T>): Boolean =
    (this as MutableCollection<T>).removeAll(elements)

/**
 * Removes all elements contained in the given array from this collection.
 * Allows to overcome type-safety restriction of `removeAll` that requires to pass a collection of type `Collection<E>`
 * @return `true` if any element was removed from the collection, `false` if the collection was not modified.
 */
@Suppress("UNCHECKED_CAST")
public fun <T> MutableCollection<T>.removeAll(elements: Array<T>): Boolean =
    (this as MutableCollection<T>).removeAll(elements)

/**
 * Removes all elements contained in the given sequence from this collection.
 * Allows to overcome type-safety restriction of `removeAll` that requires to pass a collection of type `Collection<E>`
 * @return `true` if any element was removed from the collection, `false` if the collection was not modified.
 */
@Suppress("UNCHECKED_CAST")
public fun <T> MutableCollection<T>.removeAll(elements: Sequence<T>): Boolean =
    (this as MutableCollection<T>).removeAll(elements)

/**
 * Adds the specified element to this collection.
 */
public fun <T> MutableList<T>.add(element: T) =
    this.add(element)

/**
 * Adds all elements of the given collection to this collection.
 */
public fun <T> MutableList<T>.addAll(elements: Collection<T>): Boolean =
    when (elements) {
        is Collection -> addAll(elements)
        else -> {
            var result: Boolean = false
            for (item in elements)
                if (add(item)) result = true
            return result
        }
    }

/**
 * Adds all elements of the given sequence to this collection.
 */
public fun <T> MutableList<T>.addAll(elements: Sequence<T>): Boolean =
    when (elements) {
        is Collection -> addAll(elements)
        else -> {
            var result: Boolean = false
            for (item in elements)
                if (add(item)) result = true
            return result
        }
    }

/**
 * Removes a single instance of the specified element from this collection.
 */
public fun <T> MutableList<T>.remove(element: T): Boolean =
    this.remove(element)

/**
 * Removes all elements contained in the given collection from this collection.
 */
public fun <T> MutableList<T>.removeAll(elements: Collection<T>): Boolean =
    when (elements) {
        is Collection -> removeAll(elements)
        else -> {
            var result: Boolean = false
            for (item in elements)
                if (add(item)) result = true
            return result
        }
    }

/**
 * Removes all elements contained in the given array from this collection.
 */
public fun <T> MutableList<T>.removeAll(elements: Array<T>): Boolean =
    when (elements) {
        is Collection -> removeAll(elements)
        else -> {
            var result: Boolean = false
            for (item in elements)
                if (add(item)) result = true
            return result
        }
    }

/**
 * Removes all elements contained in the given sequence from this collection.
 */
public fun <T> MutableList<T>.removeAll(elements: Sequence<T>): Boolean =
    when (elements) {
        is Collection -> removeAll(elements)
        else -> {
            var result: Boolean = false
            for (item in elements)
                if (add(item)) result = true
            return result
        }
    }
MutableCollection<in T>: Boolean { 
    return addAll(elements.asList())
}

/**
 * Removes all elements from this [MutableCollection] that are contained in the given [elements] sequence.
 */
public fun <T> MutableCollection<in T>.removeAll(elements: Sequence<T>): Boolean {
    val set = elements.toHashSet()
    return set.isNotEmpty() && removeAll(set)
}

/**
 * Removes all elements from this [MutableCollection] that match the given [predicate] from this collection.
 *
 * @return `true` if any element was removed from this collection, or `false` when no elements were removed.
 */
public fun <T> MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, true)

/**
 * Retains only elements of this [MutableCollection] that match the given [predicate] from this collection.
 *
 * @return `true` if any element was removed from this collection, or `false` when all elements were retained.
 */
public fun <T> MutableList<T>.retainAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, false)

private fun <T> MutableList<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {
    var result = false
    with(iterator()) {
        while (hasNext())
            if (predicate(next()) == predicateResultToRemove) {
                remove()
                result = true
            }
    }
    return result
}

/**
 * Removes the first element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> MutableList<T>.removeFirst(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else removeAt(0)

/**
 * Removes the first element from this mutable list and returns that removed element, or returns `null` if this list is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> MutableList<T>.removeFirstOrNull(): T? = if (isEmpty()) null else removeAt(0)

/**
 * Removes the last element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> MutableList<T>.removeLast(): T = if (isEmpty()) throw NoSuchElementException("List is empty.") else removeAt(lastIndex)

/**
 * Removes the last element from this mutable list and returns that removed element, or returns `null` if this list is empty.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
public fun <T> MutableList<T>.removeLastOrNull(): T? = if (isEmpty()) null else removeAt(lastIndex)
removed from this collection, or 'false' when no elements were removed and collection was not modified.

```java
/*
 * @public fun <T> MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, true) ||
 * Retains only elements of this [MutableList] that match the given [predicate].
 */
public fun <T> MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, false).unre
```
@RestrictsSuspension
@SinceKotlin("1.3")
@RestrictsSuspension
@.SinceKotlin("1.3")
public abstract class SequenceScope<in T> internal constructor() {
    /**
     * Yields a value to the [Iterator] being built and suspends until the next value is requested.
     */
    public abstract suspend fun yield(value: T)

    /**
     * Yields all values from the `iterator` to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public abstract suspend fun yieldAll(iterator: Iterator<T>)

    /**
     * Yields potentially infinite sequence of values to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(elements: Iterable<T>)

    /**
     * Yields potentially infinite sequence of values to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(sequence: Sequence<T>)
}

private class SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {
    private var state = State_NotReady
    private var nextValue: T? = null
    private var nextIterator: Iterator<T>? = null
    var nextStep: Continuation<Unit>? = null

    override fun hasNext(): Boolean {
        while (true) {
            when (state) {
                State_NotReady -> {}
                State_ManyNotReady -> {
                    if (nextIterator!!.hasNext()) {
                        state = State_ManyReady
                        return true
                    } else {
                        nextIterator = null
                        state = State_NotReady
                        nextStep = null
                    }
                }
                State_Done -> return false
                State_Ready, State_ManyReady -> return true
                else -> throw exceptionalState()
            }

            state = State_Failed
            nextStep = null
        }
    }

    override fun next(): T {
        when (state) {
            State_NotReady, State_ManyNotReady -> return nextNotReady()
            State_ManyReady -> {
                state = State_NotReady
                nextValue = nextValue!!
                return nextValue
            }
            State_Done -> return false
            State_Failed -> throw exceptionalState()
            else -> throw exceptionalState()
        }
    }

    override suspend fun yield(value: T) {
        nextValue = value
        state = State_Ready
        return suspendCoroutineUninterceptedOrReturn { c ->
        }
    }

    override suspend fun yieldAll(iterator: Iterator<T>)

    override suspend fun yieldAll(elements: Iterable<T>)

    override suspend fun yieldAll(sequence: Sequence<T>)
}

@RestrictsSuspension
@SinceKotlin("1.3")
@RestrictsSuspension
@SinceKotlin("1.3")
public abstract class SequenceScope<in T> internal constructor() {
    /**
     * Yields a value to the [Iterator] being built and suspends until the next value is requested.
     */
    public abstract suspend fun yield(value: T)

    /**
     * Yields all values from the `iterator` to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public abstract suspend fun yieldAll(iterator: Iterator<T>)

    /**
     * Yields potentially infinite sequence of values to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(elements: Iterable<T>)

    /**
     * Yields potentially infinite sequence of values to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(sequence: Sequence<T>)
}

private class SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {
    private var state = State_NotReady
    private var nextValue: T? = null
    private var nextIterator: Iterator<T>? = null
    var nextStep: Continuation<Unit>? = null

    override fun hasNext(): Boolean {
        while (true) {
            when (state) {
                State_NotReady -> {}
                State_ManyNotReady -> {
                    if (nextIterator!!.hasNext()) {
                        state = State_ManyReady
                        return true
                    } else {
                        nextIterator = null
                        state = State_NotReady
                        nextStep = null
                    }
                }
                State_Done -> return false
                State_Ready, State_ManyReady -> return true
                else -> throw exceptionalState()
            }

            state = State_Failed
            nextStep = null
        }
    }

    override fun next(): T {
        when (state) {
            State_NotReady, State_ManyNotReady -> return nextNotReady()
            State_ManyReady -> {
                state = State_NotReady
                nextValue = nextValue!!
                return nextValue
            }
            State_Done -> return false
            State_Failed -> throw exceptionalState()
            else -> throw exceptionalState()
        }
    }

    override suspend fun yield(value: T) {
        nextValue = value
        state = State_Ready
        return suspendCoroutineUninterceptedOrReturn { c ->
        }
    }

    override suspend fun yieldAll(iterator: Iterator<T>)

    override suspend fun yieldAll(elements: Iterable<T>)

    override suspend fun yieldAll(sequence: Sequence<T>)
}

@RestrictsSuspension
@SinceKotlin("1.3")
@RestrictsSuspension
@SinceKotlin("1.3")
public abstract class SequenceScope<in T> internal constructor() {
    /**
     * Yields a value to the [Iterator] being built and suspends until the next value is requested.
     */
    public abstract suspend fun yield(value: T)

    /**
     * Yields all values from the `iterator` to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public abstract suspend fun yieldAll(iterator: Iterator<T>)

    /**
     * Yields potentially infinite sequence of values to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(elements: Iterable<T>)

    /**
     * Yields potentially infinite sequence of values to the [Iterator] being built and suspends until all these values are iterated and the next one is requested.
     */
    public suspend fun yieldAll(sequence: Sequence<T>)
}

private class SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {
this.size) result[this.size] = null
return result as Array<T>

override fun toArray(): Array<Any?>
{
    return toArray(arrayOfNulls(size))
}

    override fun toArray(): Array<Any?>
    {
        return toArray(arrayOfNulls(size))
    }

    /**
     * Creates a new ring buffer with the capacity equal to the
minimum of [maxCapacity] and 1.5 * [capacity].
     * The returned ring buffer contains the same elements as
this ring buffer.
     */
    fun expanded(maxCapacity: Int): RingBuffer<T> {
        val newCapacity = (capacity + (capacity shr 1) + 1).coerceAtMost(maxCapacity)
        val newBuffer = if (startIndex == 0)
        buffer.copyOf(newCapacity) else toArray(arrayOfNulls(newCapacity))
        return RingBuffer(newBuffer, size)
    }

    /**
     * Add [element] to the buffer or fail with [IllegalStateException] if no free space available in the
buffer.
     */
    fun add(element: T) {
        if (isFull()) {
            throw IllegalStateException("ring buffer is full")
        }

        buffer[startIndex.forward(size)] = element
        size++
    }

    /**
     * Removes [n] first elements from the buffer or fails with [IllegalArgumentException] if not enough elements in the buffer to
remove.
     */
    fun removeFirst(n: Int) {
        require(n >= 0) { "n shouldn't be negative but it is $n" }
        require(n <= size) { "n shouldn't be greater than the buffer size: n = $n, size = $size" }

        if (n > 0) {
            val start = startIndex
            val end = start.forward(n)

            if (start > end) {
                buffer.fill(null, start,
            capacity)
            buffer.fill(null, 0, end)
        } else {
            buffer.fill(null, start, end)
        }
    }

    @Suppress("NOTHING_TO_INLINE")
    private inline fun Int.forward(n: Int): Int = (this + n) % capacity
}

package kotlin.collections

// UByteArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(
    array: UByteArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(
    array: UByteArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// UShortArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(
    array: UShortArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(
    array: UShortArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// UIntArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(
    array: UIntArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(
    array: UIntArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}

// ULongArray
=============================================================================
@ExperimentalUnsignedTypes
private fun partition(
    array: ULongArray, left: Int, right: Int): Int {
    var i = left
    var j = right
    val pivot = array[(left + right) / 2]
    while (i <= j) {
        while (array[i] < pivot)
            i++
        while (array[j] > pivot)
            j--
        if (i <= j) {
            val tmp = array[i]
            array[i] = array[j]
            array[j] = tmp
            i++
            j--
        }
    }
    return i
}

@ExperimentalUnsignedTypes
private fun quickSort(
    array: ULongArray, left: Int, right: Int) {
    val index = partition(array, left, right)
    if (left < index - 1)
        quickSort(array, left, index - 1)
    if (index < right)
        quickSort(array, index, right)
}
- if (index < right)
  quickSort(array, index, right)
}

---

* Sorts the given array using qsort algorithm.

```kotlin
interface ContractBuilder {
    fun returns(): Returns
    fun returns(value: Any?): Returns
    fun returnsNotNull(): ReturnsNotNull
}
```

```kotlin
enum class InvocationKind {
    UNKNOWN,
    AT_MOST_ONCE,
    AT_LEAST_ONCE,
    EXACTLY_ONCE
}
```

```kotlin
public fun <R> callsInPlace(lambda: Function<R>, kind: InvocationKind = InvocationKind.UNKNOWN): CallsInPlace
```

---

Open Source Used In DNAC Platform Antelope 14211
samples.contracts.callsInPlaceAtMostOnceContract
    @ContractsDsl AT_MOST_ONCE,

    /**
     * A function parameter will be invoked one or more times.
     */
    // @sample

samples.contracts.callsInPlaceAtLeastOnceContract
    @ContractsDsl AT_LEAST_ONCE,

    /**
     * A function parameter will be invoked exactly one time.
     */
    // @sample

samples.contracts.callsInPlaceExactlyOnceContract
    @ContractsDsl EXACTLY_ONCE,

    /**
     * A function parameter is called in place, but it's unknown how many times it can be called.
     */
    // @sample

samples.contracts.callsInPlaceUnknownContract
    @ContractsDsl UNKNOWN
}

/**
 * Specifies the contract of a function.
 *
 * The contract description must be at the beginning of a function and have at least one effect.
 * Only the top-level functions can have a contract for now.
 *
 * @param builder the lambda where the contract of a function is described with the help of the [ContractBuilder] members.
 *
 */
/* @sample
 * @sample samples.contracts.returnsContract
 * @sample samples.contracts.returnsTrueContract
 * @sample samples.contracts.returnsFalseContract
 * @sample samples.contracts.returnsNullContract
 * @sample samples.contracts.callsInPlaceAtMostOnceContract
 * @sample samples.contracts.callsInPlaceAtLeastOnceContract
 * @sample samples.contracts.callsInPlaceExactlyOnceContract
 * @sample samples.contracts.callsInPlaceUnknownContract
*/
@ContractsDsl
@ExperimentalContracts
@InlineOnly
@SinceKotlin("1.3")
@Suppress("UNUSED_PARAMETER")
public inline fun contract(builder:
    ContractBuilder.() -> Unit) { }

package kotlin.coroutines

/**
 * Marks coroutine context element that intercepts coroutine continuations.
 * The coroutines framework uses [ContinuationInterceptor.Key] to retrieve the interceptor and
 * intercepts all coroutine continuations with [interceptContinuation] invocations.
 *
 * [ContinuationInterceptor] behaves like a [polymorphic element][AbstractCoroutineContextKey], meaning that
 * its implementation delegates [get][CoroutineContext.Element.get] and
 * [minusKey][CoroutineContext.Element.minusKey] to [getPolymorphicElement] and [minusPolymorphicKey]
 * respectively.
 *
 * [ContinuationInterceptor] subtypes can be extracted from the coroutine context using either
 * [ContinuationInterceptor.Key] or subtype key if it extends [AbstractCoroutineContextKey].
 *
 * The key that defines "the" context interceptor.
 *
 * companion object Key :
 * CoroutineContext.Key<ContinuationInterceptor>

/**
 * Returns continuation that wraps the original continuation, thus intercepting all resumptions.
 * This function is invoked by coroutines framework when needed and the resulting continuations are cached internally per each instance of the original continuation.
 *
 * This function may simply return original continuation if it does not want to intercept this particular continuation.
 *
 * When the original continuation completes, coroutine framework invokes [releaseInterceptedContinuation] with the resulting continuation if it was intercepted, that is if
 * `interceptContinuation` had previously returned a different continuation instance.
 *
 * public fun <T> interceptContinuation(continuation: Continuation<T>): Continuation<T>

 */
Open Source Used In DNAC Platform Antelope 14213

// minusPolymorphicKey specialized for ContinuationInterceptor key

@OptIn(ExperimentalStdlibApi::class) if (key is AbstractCoroutineContextKey<*>) { return if (key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this
}@OptIn(ExperimentalStdlibApi::class) if (ContinuationInterceptor === key) EmptyCoroutineContext else this

/*
 * Copyright 2010-2018
 * JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.coroutines

/**
 * Persistent context for the coroutine. It is an indexed set of [Element] instances.
 * An indexed set is a mix between a set and a map.
 * Every element in this set has a unique [Key],
 * public interface CoroutineContext
 * public fun <E : Element> get(key: Key<E>): E?
 * Accumulates entries of this context starting with [initial] value and applying [operation] and each element of this context.
 * public fun <R> fold(initial: R, operation: (R, Element) -> R): R
 * Returns a context containing elements from this context and elements from other [context].
 * The elements from this context with the same key as in the other one are dropped.
 * public operator fun plus(context: CoroutineContext): CoroutineContext
 * Returns a context containing elements from this context, but without an element with [key].
 * public interface Element
 * A key of this coroutine context element.
 * override operator fun <E : Element> get(key: Key<E>): E? =
 * public override fun minusKey(key: Key<*>): CoroutineContext
 * 
 */

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.coroutines

import kotlin.coroutines.CoroutineContext.Element
import kotlin.coroutines.CoroutineContext.Key

/**
 * Base class for [CoroutineContext.Element] implementations.
 */
@SinceKotlin("1.3")
public abstract class AbstractCoroutineContextElement(public override val key: Key<*>) : Element

/**
 * Polymorphic element implementation implies delegating its [get][Element.get] and [minusKey][Element.minusKey] to [getPolymorphicElement] and [minusPolymorphicKey] respectively.
 * Polymorphic elements can be extracted from the coroutine context using both element key and its supertype key.
 * Example of polymorphic elements:
 * open class BaseElement : CoroutineContext.Element
 * override val key: CoroutineContext.Key<*> = Key
 * // It is important to use getPolymorphicKey and minusPolymorphicKey
 * override fun minusKey(key: CoroutineContext.Key<*>): CoroutineContext = minusPolymorphicKey(key)
 */

Open Source Used In DNAC Platform Antelope 14213
someContext[BaseElement] // Returns BaseElement?, non-null both for BaseElement and DerivedElement instances
* someContext[DerivedElement] // Returns DerivedElement?, non-null only for DerivedElement instance
* @param B base class of a polymorphic element
* @param base key an instance of base key
* @param E element type associated with the current key
* @param safeCast a function that can safely cast abstract [Context.Element] to the concrete [E] type
* and return the element if it is a subtype of [E] or `null` otherwise.

AbstractCoroutineContextKey<B : Element, E : B>(baseKey: Key<B>, private val safeCast: (element: Element) -> E?
) : Key<E> {

    internal fun tryCast(element: Element): E? = safeCast(element)
    internal fun isSubKey(key: Key<*>): Boolean = key === this || topmostKey === key

    override fun <E : Element> getPolymorphicElement(key: Key<E>): E? {
        if (key is AbstractCoroutineContextKey<*>) {
            return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null
        }

        return if (this.key === key) this as E else null
    }

    override fun minusPolymorphicKey(key: Key<*>): CoroutineContext {
        if (key is AbstractCoroutineContextKey<*>) {
            return if (key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this
        }

        return if (this.key === key) EmptyCoroutineContext else this
    }

    override fun get(key: Key<E>): E? {
        var cur = this
        while (true) {
            cur.element[key]?.let { return it }
            val next = cur.left
            if (next is CombinedContext) {
                cur = next
            } else {
                return next[key]
            }
        }
    }

    override fun <R> fold(initial: R, operation: (R, Element) -> R): R = operation(left.fold(initial, operation), element)

    override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }
        var newLeft = left.minusKey(key)
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element
            else -> CombinedContext(newLeft, element)
        }
    }

    override fun equals(other: Any?): Boolean = this === other || other is CombinedContext
}

public abstract class AbstractCoroutineContextKey<B : Element, E : B>(
    baseKey: Key<B>,
    private val safeCast: (element: Element) -> E?
) : Key<E> {
    private val topmostKey: Key<*> = if (baseKey is AbstractCoroutineContextKey<*, *>) baseKey.topmostKey else baseKey

    internal fun tryCast(element: Element): E? = safeCast(element)
    internal fun isSubKey(key: Key<*>): Boolean = key === this || topmostKey === key

    override fun getPolymorphicElement(key: Key<E>): E? {
        if (key is AbstractCoroutineContextKey<*>) {
            return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null
        }

        return if (this.key === key) this as E else null
    }

    override fun minusPolymorphicKey(key: Key<*>): CoroutineContext {
        if (key is AbstractCoroutineContextKey<*>) {
            return if (key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this
        }

        return if (this.key === key) EmptyCoroutineContext else this
    }

    override fun get(key: Key<E>): E? {
        var cur = this
        while (true) {
            cur.element[key]?.let { return it }
            val next = cur.left
            if (next is CombinedContext) {
                cur = next
            } else {
                return next[key]
            }
        }
    }

    override fun <R> fold(initial: R, operation: (R, Element) -> R): R = operation(left.fold(initial, operation), element)

    override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }
        var newLeft = left.minusKey(key)
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element
            else -> CombinedContext(newLeft, element)
        }
    }

    override fun equals(other: Any?): Boolean = this === other || other is CombinedContext
}

public object EmptyCoroutineContext : CoroutineContext, Serializable {
    private const val serialVersionUID: Long = 0
    private fun readResolve(): Any = EmptyCoroutineContext

    public override fun <E : Element> get(key: Key<E>): E? {
        if (key is AbstractCoroutineContextKey<*>) {
            return if (key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this
        }

        return if (this.key === key) EmptyCoroutineContext else this
    }

    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R = operation(left.fold(initial, operation), element)

    public override fun minusKey(key: Key<*>): CoroutineContext {
        element[key]?.let { return left }
        var newLeft = left.minusKey(key)
        return when {
            newLeft === left -> this
            newLeft === EmptyCoroutineContext -> element
            else -> CombinedContext(newLeft, element)
        }
    }

    override fun equals(other: Any?): Boolean = this === other || other is CombinedContext
}

---

Open Source Used In DNAC Platform Antelope 14214
override fun hashCode(): Int = left.hashCode() + element.hashCode()

    override

    override fun toString(): String =
        "[" + fold("") { acc, element ->
            if (acc.isEmpty()) element.toString() else
            "$acc, $element"
        } + "]"

    private fun writeReplace(): Any {
        val n = size()
        val elements =
        arrayOfNulls<CoroutineContext>(n)
        var index = 0
        fold(Unit) { _, element -> elements[index++] =
            element }
    }

    private class Serialized(val elements: Array<CoroutineContext>) :
        Serializable {
        companion object {
            private const val serialVersionUID: Long = 0L
        }

        private fun readResolve(): Any = elements.fold(EmptyCoroutineContext, CoroutineContext::plus)
    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {

    }

    companion object {
Byte.and(other: Byte): Byte = (this.toInt() and other.toInt()).toByte()

/** Performs a bitwise OR operation between the two values. */
@kotlin.Experimental
@kotlin.InternalIntraClassInlineOnly
public inline infix fun
Byte.or(other: Byte): Byte = (this.toInt() or other.toInt()).toByte()

/** Performs a bitwise XOR operation between the two values. */
@kotlin.Experimental
@kotlin.InternalIntraClassInlineOnly
public inline infix fun
Byte.xor(other: Byte): Byte = (this.toInt() xor other.toInt()).toByte()

/** Inverts the bits in this value. */
@kotlin.Experimental
@kotlin.InternalIntraClassInlineOnly
public inline fun Byte.inv(): Byte =
(this.toInt().inv()).toByte()

/** Performs a bitwise AND operation between the two values. */
@kotlin.Experimental
@kotlin.InternalIntraClassInlineOnly
public inline infix fun
Short.and(other: Short): Short =
(this.toInt() and other.toInt()).toShort()

/** Performs a bitwise OR operation between the two values. */
@kotlin.Experimental
@kotlin.InternalIntraClassInlineOnly
public inline infix fun
Short.or(other: Short): Short =
(this.toInt() or other.toInt()).toShort()

/** Performs a bitwise XOR operation between the two values. */
@kotlin.Experimental
@kotlin.InternalIntraClassInlineOnly
public inline infix fun
Short.xor(other: Short): Short =
(this.toInt() xor other.toInt()).toShort()

/** Inverts the bits in this value. */
@kotlin.Experimental
@kotlin.InternalIntraClassInlineOnly
public inline fun Short.inv(): Short =
(this.toInt().inv()).toShort()
DynamicExtension

/**
 * The value of this parameter should be a property reference expression (`this::foo`), referencing a ‘lateinit’ property.
 * The backing field of which is accessible at the point where the corresponding argument is passed.
 */
@Target(AnnotationTarget.VALUE_PARAMETER)
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
@internal annotation class AccessibleLateinitPropertyLiteral

/**
 * Specifies that this declaration is a part of special DSL, used for constructing function's contract.
 * @sample samples.properties.Delegates.notNullDelegate
 */
@Target(AnnotationTarget.COMPOSITION Tiếpउरुङ्ग 
@Retention(AnnotationRetention.BINARY)
@SinceKotlin("1.2")
@internal annotation class ContractsDsl

/*
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.properties

import kotlin.reflect.KProperty

/**
 * Standard property delegates.
 */
public object Delegates {
    /**
     * Returns a property delegate for a read/write property with a non-null value that is initialized not during
     * object construction time but at a later time.
     * @sample samples.properties.Delegates.notNullDelegate
     */
    public fun <T : Any> notNull(): ReadWriteProperty<Any?, T> = NotNullVar()

    /**
     * Returns a property delegate for a read/write property that calls a specified callback function when changed.
     * @param initialValue the initial value of the property.
     * @sample samples.properties.Delegates.observableDelegate
     */
    fun observable(initialValue: T, crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Unit): ReadWriteProperty<Any?, T> = ObservableProperty(initialValue) {
        override fun afterChange(property: KProperty<*>, oldValue: T, newValue: T) = onChange(property, oldValue, newValue)
    }

    /**
     * Returns a property delegate for a read/write property that calls a specified callback function when changed.
     * @param initialValue the initial value of the property.
     * @param onChange the callback which is called after the change of the property is made. The value of the property
     * has already been changed when this callback is invoked.
     * @sample samples.properties.Delegates.observableDelegate
     */
    public fun <T> observableInlineDelegate(initialValue: T): ReadWriteProperty<Any?, T> = ObservableProperty(initialValue) {
        override fun afterChange(property: KProperty<*>, oldValue: T, newValue: T) = onChange(property, oldValue, newValue)
    }

    /**
     * Allows the callback to veto the modification.
     * @param initialValue the initial value of the property.
     * @param onChange the callback which is called before a change to the property value is attempted.
     */
    public fun <T> observableWithVeto(initialValue: T): ReadWriteProperty<Any?, T> = ObservableProperty(initialValue) {
        override fun afterChange(property: KProperty<*>, oldValue: T, newValue: T) = onChange(property, oldValue, newValue)
    }

    /**
     * Returns a property delegate for a read/write property that calls a specified callback function when changed.
     * If the callback returns `true` the value of the property has not been changed yet, when this callback is invoked.
     * If the callback returns `false` the new value
is discarded and the property remains its old value.

```kotlin
samples.properties.Delegates.vetoableDelegate
* @sample samples.properties.Delegates.throwVetoableDelegate
*/

class NotNullVar<T : Any>() : ReadWriteProperty<Any?, T> {
    private var value: T? = null

    public override fun getValue(thisRef: Any?, property: KProperty<*>) = value
        ?: throw IllegalStateException("Property ${property.name} should be initialized before get.")

    public override fun setValue(thisRef: Any?, property: KProperty<*>, value: T) {
        this.value = value
    }
}

@SinceKotlin("1.4")
fun interface PropertyDelegateProvider<in T, out D> {
    override fun provideDelegate(thisRef: T, property: KProperty<*>) = D
}

package kotlin.properties

import kotlin.reflect.KProperty

/**
 * Implements the core logic of a property delegate for a read/write property that calls callback functions when
 * it is accessed.
 * @param initialValue the initial value of the property.
 */
public abstract class ObservableProperty<V>(initialValue: V) : ReadWriteProperty<Any?, V> {
    protected open fun beforeChange(property: KProperty<*>, oldValue: V, newValue: V) {
        // Callbacks
    }
}

class Delegates<
    T : Any,
    V : Any
>(
    property: KProperty<*>,
    oldValue: V,
    newValue: V
) {
    private var vars: List<T?> = listOf()

    public override fun getValue(thisRef: Any?, property: KProperty<*>) = vars.getProperty(property)

    public override fun setValue(thisRef: Any?, property: KProperty<*>, value: Any?) {
        vars.setProperty(property, value)
    }
}

@sample '

// Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.

Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

*@

Open Source Used In DNAC Platform Antelope 14218

*/
newValue: V): Boolean = true
   * The callback which is called after the change of the property is made.
   * The value of the property has already been changed when this callback is invoked.

   protected open fun afterChange(property: KProperty<*>!, oldValue: V, newValue: V): Unit {
      public override fun getValue(thisRef: Any?, property: KProperty<*>): V {
         return value
      }
      public override fun setValue(thisRef: Any?, property: KProperty<*>!, value: V) {
         val oldValue = this.value
         if (!beforeChange(property, oldValue, value)) {
            return
         }
         this.value = value
         afterChange(property, oldValue, value)
      }
   }
   */

   /*
    * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
    * Use of this source code is governed by the Apache 2.0 license that can be found in
    * the license/LICENSE.txt file. *
    */
   @file:Suppress("PackageDirectoryMismatch")

   package kotlin

   import kotlin.reflect.*

   /**
    * An extension operator that allows delegating a read-only property of type [V] to a
    * property reference to a property of type of type [V] or its subtype. * *receiver A property
    * reference to a read-only or mutable property of type [V] or its subtype. * The reference is
    * without a receiver, i.e. it either references a top-level property or has the receiver bound to it.
    * Examples:
    * ```
    * class Login(val username: String)
    * val defaultLogin = Login("Admin")
    * val defaultUsername by defaultLogin::username
    * // equivalent to
    * val defaultUsername
    * ```
    */
   @SinceKotlin("1.4")
   @kotlin.internal.InlineOnly
   public inline operator fun <V> KProperty<*>.getValue(thisRef: Any?, property: KProperty<*>): V {
      return get(thisRef)
   }

   /**
    * An extension operator that allows delegating a mutable property of type [V] to a
    * property reference to a mutable property of type [V]. * *receiver A property reference to a
    * read-only or mutable property of type [V] or its subtype. * The reference is
    * without a receiver, i.e. it either references a top-level property or has the receiver bound to it.
    * Examples:
    * ```
    * class Login(val username: String, var incorrectAttemptCounter: Int)
    * var Login.attempts by Login::incorrectAttemptCounter
    * // equivalent to
    * ```
    */
   @SinceKotlin("1.4")
   @kotlin.internal.InlineOnly
   public inline operator fun <T, V> KMutableProperty0<T, V>.setValue(thisRef: T, property: KProperty<*>!, value: V) {
      set(thisRef, value)
   }

   /**
    * An extension operator that allows delegating a read-only member or extension property of type [V] to a
    * property reference to a member or extension property of type [V] or its subtype. * The reference is
    * without a receiver, i.e. it either references a top-level property or has the receiver bound to it.
    * Examples:
    * ```
    * class Login(val username: String)
    * val Login.user by Login::username
    * // equivalent to
    * ```
    */
   @SinceKotlin("1.4")
   @kotlin.internal.InlineOnly
   public inline operator fun <T, V> KProperty1<T, V>.getValue(thisRef: T, property: KProperty<*>): V {
      return get(thisRef)
   }

   /**
    * An extension operator that allows delegating a mutable member or extension property of type [V] to a
    * property reference to a read-only or mutable property of type [V] or its subtype. * The reference is
    * without a receiver, i.e. it either references a top-level property or has the receiver bound to it.
    * Examples:
    * ```
    * class Login(val username: String, var incorrectAttemptCounter: Int)
    * var Login.attempts by Login::incorrectAttemptCounter
    * ```
    */
   @SinceKotlin("1.4")
   @kotlin.internal.InlineOnly
   public inline operator fun <T, V> KMutableProperty1<T, V>.setValue(thisRef: T, property: KProperty<*>!, value: V) {
      set(thisRef, value)
   }
upper bits are zero.

* @param bitCount number of bits to generate, must be in range 0..32, otherwise the behavior is unspecified.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.

* @throws IllegalArgumentException if [until] is negative or zero.
```kotlin
checkRangeBounds(from, until) { val size = until - from
val r = if (size.isInfinite()) && from.isFinite() && until.isFinite())

val r1 = nextDouble() * (until / 2 - from / 2)
val from + r1 + r1 } else {
val from + nextDouble() * size
}
return if (r >= until) until.nextDown() else r
}

/**
 * Gets the next random [Float] value uniformly distributed between 0 (inclusive) and 1 (exclusive).
 */
public open fun nextFloat(): Float = nextBits(24) / (1 shl 24).toFloat()

/**
 * Fills a subrange of the specified byte [array] starting from [fromIndex] inclusive and ending [toIndex] exclusive with random bytes.
 * @return [array] with the subrange filled with random bytes.
 */
public open fun nextBytes(array: ByteArray, fromIndex: Int = 0, toIndex: Int = array.size): ByteArray {
require(fromIndex in 0..array.size && toIndex in 0..array.size) { "fromIndex ($fromIndex) or toIndex ($toIndex) are out of range: 0..${array.size}." }
require(fromIndex <= toIndex) { "fromIndex ($fromIndex) must be not greater than toIndex ($toIndex)." }
val steps = (toIndex - fromIndex) / 4
var position = fromIndex
repeat(steps) {
val v = nextInt()
array[position] = v.toByte()
array[position + 1] = v.ushr(8).toByte()
array[position + 2] = v.ushr(16).toByte()
array[position + 3] = v.ushr(24).toByte()
position += 4
}
val remainder = toIndex - position
val vr = nextBits(remainder * 8)
for (i in 0 until remainder) {
array[position + i] = vr.ushr(i * 8).toByte()
}
return array
}

/**
 * The default random number generator.
 * On JVM this generator is thread-safe, its methods can be invoked from multiple threads.
 * @return [array] filled with random bytes.
 */
public open fun nextBytes(array: ByteArray): ByteArray = nextBytes(array, 0, array.size)

/**
 * Creates a byte array of the specified [size], filled with random bytes.
 */
public open fun nextBytes(size: Int): ByteArray = nextBytes(ByteArray(size))

/**
 * Returns a repeatable random number generator seeded with the given [seed] `Int` value.
 * Two generators with the same seed produce the same sequence of values within the same version of Kotlin runtime.
 * @sample samples.random.Randoms.seededRandom
 */
@SinceKotlin("1.3")
public fun Random(seed: Int): Random = XorWowRandom(seed, seed.shr(31))

/**
 * Returns a repeatable random number generator seeded with the given [seed] `Long` value.
 * Two generators with the same seed produce the same sequence of values within the same version of Kotlin runtime.
 * @sample samples.random.Randoms.seededRandom
 */
@SinceKotlin("1.3")
public fun Random(seed: Long): Random = XorWowRandom(seed.toInt(), seed.shr(32).toInt())
```

Open Source Used In DNAC Platform Antelope 14221
specified [range]:
* from 'range.start' inclusive to 'range.endInclusive' inclusive.
* @throws
IllegalArgument if [range] is empty.
* @SinceKotlin("1.3")
*public fun Random.nextInt(range: IntRange): Int = when {
* range.isEmpty() -> throw IllegalArgument("Cannot get random in empty range: $range")
* range.last < Int.MAX_VALUE -> nextInt(range.first, range.last + 1)
* range.first > Int.MIN_VALUE -> nextInt(range.first - 1, range.last) + 1
* else -> nextInt(range.last)
* * Generates a 'Long' random value uniformly distributed in the specified [range]:
* from 'range.start' inclusive to 'range.endInclusive' inclusive.
* @throws
IllegalArgument if [range] is empty.
* @SinceKotlin("1.3")
*public fun Random.nextLong(range: LongRange): Long = when {
* range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range: $range")
* range.last < Long.MAX_VALUE -> nextLong(range.first, range.last + 1)
* range.first > Long.MIN_VALUE -> nextLong(range.first - 1, range.last) + 1
* else -> nextLong(range)
* *.internal expect fun defaultPlatformRandom(): Random
* *internal fun doubleFromParts(hi26: Int, low27: Int): Double
* *internal fun fastLog2(value: Int): Int = 31 - value.countLeadingZeroBits()
* *internal fun Int.takeUpperBits(bitCount: Int): Int =
* this.ushr(32 - bitCount) and (-bitCount).shr(31)
* *internal fun checkRangeBounds(from: Int, until: Int) = require(until > from) {
* boundsErrorMessage(from, until)
* }
* *internal fun checkRangeBounds(from: Long, until: Long) = require(until > from) {
* boundsErrorMessage(from, until)
* }
* *internal fun checkRangeBounds(from: Double, until: Double) = require(until > from) {
* boundsErrorMessage(from, until)
* }
* *internal fun boundsErrorMessage(from: Any, until: Any) = "Random range is empty: [$from, $until)."
* * * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
* * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
* *internal package kotlin.random
* * * * Generates a [UInt] random value uniformly distributed between [UInt.MIN_VALUE] and [UInt.MAX_VALUE] (inclusive).
* *@SinceKotlin("1.3")
* *@ExperimentalUnsignedTypes
* *public fun Random.nextUInt(): UInt =
* nextInt().toUInt()
* * * Generates a [UInt] random value uniformly distributed between '0' (inclusive) and the specified [bound].
* * @throws
IllegalArgument if [bound] is zero.
* *@SinceKotlin("1.3")
* *@ExperimentalUnsignedTypes
* *public fun Random.nextUInt(until: UInt): UInt =
* nextUInt(0u, until)
* * * Generates a [UInt] random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.
* * @throws
IllegalArgument if [from] is greater than or equal to [until].
* *@SinceKotlin("1.3")
* *@ExperimentalUnsignedTypes
* *public fun Random.nextUInt(from: UInt, until: UInt):
* {
* checkUIntRangeBounds(from, until)
* val signedFrom = from.toInt() xor Int.MIN_VALUE
* val signedUntil = until.toInt() xor Int.MIN_VALUE
* val signedResult = nextInt(signedFrom, signedUntil) xor Int.MIN_VALUE
* return signedResult.toUInt()
* * * Generates a [UInt] random value uniformly distributed in the specified [range]:
* * from 'range.start' inclusive to 'range.endInclusive' inclusive.
* * @throws
IllegalArgumentException if [range] is empty.
* *@SinceKotlin("1.3")
* *@ExperimentalUnsignedTypes
* *public fun Random.nextUInt(range: UIntRange): UInt = when {
* range.isEmpty() -> throw IllegalArgumentException("Cannot get random in empty range: $range")
* range.last < UInt.MAX_VALUE ->
* nextUInt(range.first, range.last + 1)
* range.first > UInt.MIN_VALUE -> nextUInt(range.first - 1u, range.last) + 1u
* else -> nextUInt(range.last)
* * Generates a [ULong] random value uniformly distributed between [ULong.MIN_VALUE] and [ULong.MAX_VALUE] (inclusive).
* *@SinceKotlin("1.3")
* *@ExperimentalUnsignedTypes
* *public fun Random.nextULong(): ULong =
* nextLong().toULong()
IllegalArgumentException if \[\text{until}\] is zero.
*/
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public

/**
 * Gets the next random \[ULong\]
 * from the random number generator in the specified range.
 * Generates a \[ULong\] random value uniformly distributed between the specified \[from\] (inclusive) and \[until\] (exclusive) bounds.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextULong(until: ULong): ULong = nextULong(0uL, until)

/**
 * Gets the next random \[ULong\] from the random number generator in the specified \[range\].
 * Generates a \[ULong\] random value uniformly distributed in the specified \[range\]: \text{\[from\]} * from \[range\].start’ inclusive to \[range\].endInclusive’ inclusive.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextULong(from: ULong, until: ULong): ULong {
    checkULongRangeBounds(from, until)

    val signedFrom = from.toLong() xor Long.MIN_VALUE
    val signedUntil = until.toLong() xor Long.MIN_VALUE

    val signedResult = nextLong(signedFrom, signedUntil) xor Long.MIN_VALUE
    return signedResult.toULong()
}

/**
 * Fills the specified unsigned byte \[array\] with random bytes and returns it.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUBytes(array: UByteArray): UByteArray {
    nextBytes(array.asByteArray())
    return array
}

/**
 * Creates an unsigned byte array of the specified \[size\], filled with random bytes.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun Random.nextUBytes(size: Int): UByteArray

/**
 * Fills a subrange of the specified \[UByte\] \[array\] starting from \[fromIndex\] inclusive and ending \[toIndex\] exclusive with random \[UBytes\].
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun

@ExperimentalUnsignedTypes
internal fun checkUIntRangeBounds(from: UInt, until: UInt) =
    require(until > from) { boundsErrorMessage(from, until) }
@ExperimentalUnsignedTypes
internal fun

Comparable<T> { override val start: T; override val endInclusive: T; }

ClosedRange<T> { \n override fun equals(other: Any?): Boolean { \n return other is ComparableRange<> \&\& (isEmpty() \&\& other.isEmpty() \n \n start == other.start \&\& endInclusive == other.endInclusive) \n } \n \n override fun hashCode(): Int { \n if (isEmpty()) -1 else 31 * start.hashCode() + endInclusive.hashCode() \n } \n \n override fun toString(): String = "${start}..${endInclusive}"
}

Comparable<T>> { \n override val start: T, \n override val endInclusive: T
} : ClosedRange<T> {

    override
    fun equals(other: Any?): Boolean {
        return other is ComparableRange<*> \&\& (isEmpty() \&\& other.isEmpty()
            \n            start == other.start \&\& endInclusive == other.endInclusive) \n    }

    override
    fun hashCode(): Int { return if (isEmpty()) -1 else 31 * start.hashCode() + endInclusive.hashCode()
    }

    override
    fun toString(): String = "${start}..${endInclusive}"
}

/**
 * Creates a range from this \[Comparable\] value to the specified \[that\] value.\n * \n * This value needs to be smaller than \[that\] value, otherwise the returned range will be empty.\n * \n * @param that the \[that\] value\n * @return a range of floating point numbers\n */

Comparable<T>a = Comparable<that> T: ComparableRange<

Comparable<T, T>: ComparableRange<T> { ComparableRange<T>(this, that)\n
<doubleRange\n
/**
 * Represents a range of floating point numbers.
 * Extends \[ClosedRange\] interface providing custom operation \n *[lessThanOrEquals]*\n for comparing values of range domain type.\n * \n * This value is implemented by floating point ranges returned by \[Float.rangeTo\] and \[Double.rangeTo\] operators\n * to achieve IEEE-754 comparison order instead of total order of floating point numbers.\n */

ClosedFloatingPointRange<T> : Comparable<T> : ClosedRange<T> { \n \n override fun contains(value: T): Boolean = \n lessThanOrEquals(start, value) \&\& lessThanOrEquals(value, endInclusive)
    override fun isEmpty(): Boolean = !lessThanOrEquals(start, endInclusive)

    /**
     * Compares two values of range domain type and returns true if first is less than or equal to second.\n     * \n     * @param a the first value\n     * @param b the second value\n     * @return true if first is less than or equal to second\n     */
    fun lessThanOrEquals(a: T, b: T): Boolean = lessThanOrEquals(a, b)

    override
    fun toString(): String = "${start}..${endInclusive}"
}

/**
 * Creates a range from this \[Double\] value to the specified \[that\] value.
 * \n * Numbers are compared with the ends of this range according to IEEE-754\n * \n * @param that the \[that\] value\n * @return a range of floating point numbers\n */

ClosedDoubleRange(Double) { \n \n private val _start = start \n private val _endInclusive = endInclusive
    override
    fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b
    override
    fun toString(): String = "${start}..${endInclusive}"
)

/**
 * Creates a range from this \[Float\] value to the specified \[that\] value.
 * \n * Numbers are compared with the ends of this range according to IEEE-754\n * \n * @param that the \[that\] value\n * @return a range of floating point numbers\n */

ClosedFloatRange<Float> { \n \n private val _start = start \n private val _endInclusive = endInclusive
    override
    fun lessThanOrEquals(a: Float, b: Float): Boolean = a <= b
    override
    fun toString(): String = "${start}..${endInclusive}"
)

/**
 * Returns \[true\] if this iterable range contains the specified \[element\].\n * \n * Always returns \[false\] if the \[element\] is \[null\].\n * \n * @param element the \[element\]\n * @return \[true\] if this iterable range contains the specified \[element\].\n */

contains(element: T?): Boolean = contains(element)
Apache 2.0 license that can be found in the license/LICENSE.txt file.

Open Source Used In DNAC Platform Antelope

14225
Represents variance applied to a type parameter on the declaration site (*declaration-site variance*), or to a type in a projection (*use-site variance*).

* @see [KTypeParameter.variance]
* @see [KTypeProjection]

The affected type parameter or type is *invariant*, which means it has no variance applied to it.

* The affected type parameter or type is *contravariant*. Denoted by the `in` modifier in the source code.

* The affected type parameter or type is *covariant*. Denoted by the `out` modifier in the source code.

*Copyright 2010-2019 JetBrains s.r.o.
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
* @file:kotlin.jvm.JvmMultifileClass
* @file:kotlin.jvm.JvmName("StringsKt")

---

```kotlin
package kotlin.text

/**
 * An object to which char sequences and values can be appended.
 */
expect interface Appendable {
    /**
     * Appends the specified character [value] to this Appendable and returns this instance.
     */
    fun append(value: Char): Appendable

    /**
     * Appends the specified character sequence [value] to this Appendable and returns this instance.
     */
    fun append(value: CharSequence?): Appendable

    /**
     * Appends a subsequence of the specified character sequence [value] to this Appendable and returns this instance.
     */
    fun append(value: CharSequence?, startIndex: Int, endIndex: Int): Appendable
}

/**
 * Appends all arguments to the given Appendable.
 */
public fun <T : Appendable> T.append(vararg value: CharSequence?): T {
    for (item in value)
        append(item)
    return this
}

```
append(element.toString())
    }]
    
    /*
     * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
     * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
     */

    /*
     * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.
     * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
     */

    @file:kotlin.jvm.JvmMultifileClass
    @file:kotlin.jvm.JvmName("CharsKt")

    package kotlin.text

    /**
     * Concatenates this Char and a String.
     * @param ignoreCase `true` to ignore character case when comparing characters.
     * @sample samples.text.CharsWithString
     */
    public inline operator fun Char.plus(other: String): String = this.toString() +
        other

    /**
     * Returns `true` if this character is equal to the [other] character, optionally ignoring character case.
     * @param ignoreCase `true` to ignore character case when comparing characters.
     * @sample samples.text.CharsWithString
     */
    public fun Char.equals(other: Char, ignoreCase: Boolean = false): Boolean {
        if (this == other) return true
        if (!ignoreCase) return false

        if (this.toUpperCase() == other.toUpperCase()) return true
        if (this.toLowerCase() == other.toLowerCase()) return true
        return false
    }

    /**
     * Returns `true` if this character is a Unicode surrogate code unit.
     */
    public fun Char.isSurrogate(): Boolean = this in Char.MIN_SURROGATE..Char.MAX_SURROGATE

    @file:kotlin.jvm.JvmMultifileClass
    @file:kotlin.jvm.JvmName("StringsKt")

    package kotlin.text

    /**
     * Trims leading whitespace characters followed by [marginPrefix] from every line of a source
     * string and removes the first and the last lines if they are blank (notice difference blank vs empty).
     * @param marginPrefix non-blank string, which is used as a margin delimiter.
     * @sample samples.text.Strings.trimMargin
     * @see trimIndent
     * @see kotlin.text.isWhitespace
     */
    public fun String.trimMargin(marginPrefix: String = "|"):
        replaceIndentByMargin("", marginPrefix)

    /**
     * Detects indent by [marginPrefix] as it does [trimMargin] and replace it with [newIndent].
     * @param marginPrefix non-blank string, which is used as a margin delimiter.
     */
    public fun String.replaceIndentByMargin(newIndent: String = "", marginPrefix: String = "|"):
        replaceIndentByMargin(newIndent, marginPrefix)

    /**
     * Detects a common minimal indent of all the input lines, removes it from every line and also removes the first and last lines if they are blank (notice difference blank vs empty).
     * @param [newIndent] function that returns the indent.
     * @sample samples.text.Strings.trimIndent
     * @see trimMargin
     * @see kotlin.text.isWhitespace
     */
    public fun String.trimIndent(): String = replaceIndent("")

    /**
     * Detects a common minimal indent like it does [trimIndent] and replaces it with the specified [newIndent].
     * @param [newIndent] function that returns the indent.
     */
    public fun String.replaceIndent(newIndent: String = ""): String {
        val lines = lines()

        val minCommonIndent = lines
            .filter(String::isNotBlank)
            .map(String::indentWidth)
            .minOrNull() ?: 0

        return lines.reindent(length + newIndent.length * lines.size,
            getIndentFunction(newIndent), { line ->
                val firstNonWhitespaceIndex = line.indexOfFirst { !it.isWhitespace()
                    when {
                        line.firstNonWhitespaceIndex == -1 -> null
                        else -> line.firstNonWhitespaceIndex
                    }
                }
                line
            })
    }

    public fun prependIndent(indent: String = "    "):
        lineSequence()
            .map { it.isNotBlank() -> {
                if (it.isBlank()) return null
                line.startsWith(marginPrefix, firstNonWhitespaceIndex) -> line.substring(firstNonWhitespaceIndex + marginPrefix.length) else -> it
            } }
            .joinToString("\n")

    private fun
String.indentWidth(): Int = indexOfFirst { !it.isWhitespace() }.let { if (it == -1) length else it }

private fun getIndentFunction(indent: String) = when {
    indent.isEmpty() -> { line: String -> line }
    else -> { line: String -> indent + line }
}

private inline fun List<String>.reindent(
    resultSizeEstimate: Int,
    indentAddFunction: (String) -> String,
    indentCutFunction: (String) -> String?
): String {
    val lastIndex = lastIndex
    return
        joinTo(StringBuilder(resultSizeEstimate), "\n")
            .toString()
}

/*
 * Copyright 2010-2018 JetBrains
 * s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin.text

/**
 * Defines names for Unicode symbols used in proper Typography.
 */
public object Typography {
    /** The character – quotation mark */
    public const val quote: Char = '\u0022'
    /** The character $ – dollar sign */
    public const val dollar: Char = '\u0024'
    /** The character & – ampersand */
    public const val amp: Char = '\u0026'
    /** The character < – less-than sign */
    public const val less: Char = '\u003C'
    /** The character > – greater-than sign */
    public const val greater: Char = '\u003E'
    /** The non-breaking space character */
    public const val nbsp: Char = '\u00A0'
    /** The character × */
    public const val times: Char = '\u00D7'
    /** The character ¢ */
    public const val cent: Char = '\u00A2'
    /** The character £ */
    public const val pound: Char = '\u00A3'
    /** The character § */
    public const val section: Char = '\u00A7'
    /** The character © */
    public const val copyright: Char = '\u00A9'
    /** The character « */
    public const val leftGuillemete: Char = '\u00AB'
    /** The character » */
    public const val rightGuillemete: Char = '\u00BB'
    /** The character ® */
    public const val registered: Char = '\u00AE'
    /** The character ° */
    public const val degree: Char = '\u00B0'
    /** The character ± */
    public const val plusMinus: Char = '\u2013'
    /** The character — */
    public const val mdash: Char = '\u2014'
    /** The character ' */
    public const val leftSingleQuote: Char = '\u2018'
    /** The character ' */
    public const val lowSingleQuote: Char = '\u201A'
    /** The character " */
    public const val leftDoubleQuote: Char = '\u201C'
    /** The character " */
    public const val lowDoubleQuote: Char = '\u201E'
    /** The character † */
    public const val dagger: Char = '\u2020'
    /** The character ‡ */
    public const val doubleDagger: Char = '\u2021'
    /** The character • */
    public const val bullet: Char = '\u2022'
    /** The character … */
    public const val ellipsis: Char = '\u2026'
    /** The character ′ */
    public const val prime: Char = '\u2032'
    /** The character ″ */
    public const val doublePrime: Char = '\u2033'
    /** The character € */
    public const val euro: Char = '\u20AC'
    /** The character ™ */
    public const val tm: Char = '\u2122'
    /** The character ≈ */
    public const val almostEqual: Char = '\u2248'
    /** The character ≠ */
    public const val notEqual: Char = '\u2260'
    /** The character ≤ */
    public const val lessOrEqual: Char = '\u2264'
    /** The character ≥ */
    public const val greaterOrEqual: Char = '\u2265'
}
Groups are indexed from 1 to the count of groups in the regular expression. A group with the index 0 corresponds to the entire match.

public operator fun get(index: Int): MatchGroup?

A list of matched indexed group values.

This list has size of `groupCount + 1` where `groupCount` is the count of groups in the regular expression.

Groups are indexed from 1 to `groupCount` and group with the index 0 corresponds to the entire match.

If the group in the regular expression is optional and there were no match captured by that group, then the groupValues is an empty string.
another instant. A negative duration is possible in a situation when the second instant is earlier than the first one. An infinite duration value [Duration.INFINITE] can be used to represent infinite timeouts.

To construct a duration use either the extension function [toDuration], or the extension properties [hours], [minutes], [seconds], and so on, available on [Int], [Long], and [Double] numeric types. To get the value of this duration expressed in a particular [duration units][DurationUnit] use the functions [toInt], [toLong], and [toDouble] or the properties [inHours], [inMinutes], [inSeconds], [inNanoseconds], and so on.

```kotlin
    /**
     * Splits this duration into hours, minutes, seconds, and nanoseconds and executes the given [action]
     * ([action] inDays, toInt), hoursComponent, minutesComponent, secondsComponent, nanosecondsComponent)
     */

    // splitting to components

    // arithmetic operators

    /** Returns the negative of this value. */
    public operator fun unaryMinus(): Duration = Duration(-value)

    /** Returns a duration whose value is this duration value multiplied by the given [scale] number. */
    public operator fun times(scale: Double): Duration = Duration(value * scale)

    /** Returns a duration whose value is this duration value multiplied by the given [scale] number. */
    public operator fun times(scale: Int): Duration = Duration(value / scale)

    /** Returns a duration whose value is this duration value divided by the given [scale] number. */
    public operator fun div(scale: Double): Duration = Duration(value / scale)

    /** Returns a duration whose value is this duration value divided by the given [scale] number. */
    public operator fun div(scale: Int): Duration = Duration(value / scale)

    // Comparables

    /** Returns true, if the duration value is greater than zero. */
    public fun isPositive(): Boolean = value > 0

    /** Returns true, if the duration value is infinite. */
    public fun isInfinite(): Boolean = value.isInfinite()

    /** Returns true, if the duration value is finite. */
    public fun isFinite(): Boolean = value.isFinite()

    /** Returns true, if the duration value is less than zero. */
    public fun isNegative(): Boolean = value < 0

    /** Returns true, if the duration value is less than zero. */
    public fun isNonNegative(): Boolean = value >= 0

    /** Returns true, if the duration value is less than zero. */
    public fun isNonPositive(): Boolean = value <= 0

    /** Returns true, if the duration value is less than zero. */
    public fun isNegative(): Boolean = value.isNaN().not()

    /** The result of [action] is returned as the result of this function. */
    public inline fun <T> toComponents(action: (days: Int, hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {
        contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }
        return
    }
```

This is useful for representing timeouts that should never expire. A negative duration is possible in a situation when the second instant is earlier than the first one.
* seconds' represents the whole number of seconds in this duration, and its absolute value is less than 60; * minutes' represents the whole number of minutes in this duration, and its absolute value is less than 60; * hours' represents the whole number of hours in this duration. If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE], it is coerced into that range.

```kotlin
/**
 * Splits this duration into minutes, seconds, and nanoseconds and executes the given [action] with these components.
 * The result of [action] is returned as the result of this function.
 * * nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000; * seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60; * minutes` represents the whole number of minutes in this duration. If the value doesn't fit in [Int] range, i.e. it's greater than [Int.MAX_VALUE] or less than [Int.MIN_VALUE], it is coerced into that range.

@PublishedApi
internal val hoursComponent: Int get() = (inHours % 24).toInt()

@PublishedApi
internal val minutesComponent: Int get() = (inMinutes % 60).toInt()

@PublishedApi
internal val secondsComponent: Int get() = (inSeconds % 60).toInt()

@PublishedApi
internal val nanosecondsComponent: Int get() = (inNanoseconds % 1e9).toInt()

// conversion to units

/**
 * The value of this duration expressed as a [Double] number of days.
 */
public val inDays: Double get() = toDouble(DurationUnit.DAYS)

/**
 * The value of this duration expressed as a [Double] number of hours.
 */
public val inHours: Double get() = toDouble(DurationUnit.HOURS)

/**
 * The value of this duration expressed as a [Double] number of minutes.
 */
public val inMinutes: Double get() = toDouble(DurationUnit.MINUTES)

/**
 * The value of this duration expressed as a [Double] number of seconds.
 */
public val inSeconds: Double get() = toDouble(DurationUnit.SECONDS)

/**
 * The value of this duration expressed as a [Double] number of milliseconds.
 */
public val inMilliseconds: Double get() = toDouble(DurationUnit.MILLISECONDS)

/**
 * The value of this duration expressed as a [Double] number of microseconds.
 */
public val inMicroseconds: Double get() = toDouble(DurationUnit.MICROSECONDS)

/**
 * The value of this duration expressed as a [Double] number of nanoseconds.
 */
public val inNanoseconds: Double get() = toDouble(DurationUnit.NANOSECONDS)

// shortcuts

/**
 * Returns the value of this duration expressed as a [Double] number of milliseconds.
 */
public fun inMilliseconds(): Double = toDouble(DurationUnit.MILLISECONDS)

/**
 * Returns the value of this duration expressed as a [Double] number of microseconds.
 */
public fun inMicroseconds(): Double = toDouble(DurationUnit.MICROSECONDS)

/**
 * Returns the value of this duration expressed as a [Double] number of nanoseconds.
 */
public fun inNanoseconds(): Double = toDouble(DurationUnit.NANOSECONDS)

// inline fun <T> toComponents(action: (hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {
contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }
return action(inHours.toInt(), minutesComponent, secondsComponent, nanosecondsComponent)
} ***/

// inline fun <T> toComponents(action: (minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {
contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }
return action(inMinutes.toInt(), secondsComponent, nanosecondsComponent)
} ***/

// inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Int) -> T): T {
contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }
return action(inSeconds.toLong(), nanosecondsComponent)
} */
```

```kotlin
// inline fun <T> toComponents(action: (hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {
contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }
return action(inHours.toInt(), minutesComponent, secondsComponent, nanosecondsComponent)
} ***/
```
public fun toLongNanoseconds(): Long = toLong(DurationUnit.NANOSECONDS)
/**
 * Returns the value of this duration expressed as a [Long] number of nanoseconds.
 */

* @Sample fun toLongMilliseconds(): Long = toLong(DurationUnit.MILLISECONDS)
/**
 * Returns a string representation of this duration value expressed in the unit which yields the most compact and readable number value.
 */

* Special cases:
 * - zero duration is formatted as `"0s"`
 * - the infinite duration is formatted as `"Infinity"`
 * - very small durations (less than 1e-15 s) are expressed in seconds and formatted in scientific notation
 * - very big durations (more than 1e+7 days) are expressed in days and formatted in scientific notation
 * @return the value of duration in the automatically determined unit followed by that unit abbreviated name: `d`, `h`, `m`, `s`, `ms`, `us`, or `ns`. 
 */

* @Sample
samples.time.Durations.toStringDefault
/**
 * Returns an ISO-8601 based string representation of this duration value expressed in the unit which yields the most compact and readable number value.
 */

* The value is coerced to the range of [Long] type, if it doesn't fit in that range, see the conversion [Double.toLong] for details.
* The range of durations that can be expressed as a `Long` number of milliseconds is approximately ±0b1292 million years.

* @Sample
public fun toString(): String = when {
    isInfinite() -> value.toString()
    value == 0.0 -> "0s"
    else -> {
        val absNs = absoluteValue.inNanoseconds
        var scientific = false
        var maxDecimals = 0
        val unit = when {
            absNs < 1e-14 -> DurationUnit.SECONDS
            absNs < 1e-12 -> DurationUnit.MINUTES
            absNs < 1e-9 -> DurationUnit.HOURS
            absNs < 1e-6 -> DurationUnit.DAYS
            else -> DurationUnit.MICROSECONDS
        }
        when {
            absNs < 1,
            absNs < 1e3,
            absNs < 1e6,
            absNs < 1e9,
            absNs < 1e12,
            absNs < 1e15,
            else -> DurationUnit.MILLISECONDS
        }
        val hasSeconds = seconds != 0 || nanoseconds != 0
        val hasMinutes = minutes != 0 || (hasSeconds && hasHours)
        if (hasHours) {
            append(hours).append('H')
        }
        if (hasMinutes) {
            append(minutes).append('M')
        }
        if (hasSeconds || (!hasHours && !hasMinutes)) {
            append(seconds)
            if (hasSeconds != 0) {
                append('.
                    val nss = nanoseconds.toString().padStart(9, '0')
                when {
                    nanoseconds % 1_000 == 0 -> appendRange(nss, 0, 6)
                    else
                        nanoseconds % 1_000_000 == 0 -> appendRange(nss, 0, 3)
                    else
                        nanoseconds % 1_000_000_000 == 0 -> appendRange(nss, 0, 9)
                    else
                        nanoseconds % 1_000_000_000_000 == 0
            }
        }
        appendRange(nss, 0, 3)
        nanoseconds % 1_000 == 0 -> appendRange(nss, 0, 6)
        else
            nanoseconds % 1_000_000 == 0 -> appendRange(nss, 0, 9)
        else
            nanoseconds % 1_000_000_000 == 0 -> appendRange(nss, 0, 12)
        else
            nanoseconds % 1_000_000_000_000 == 0
        nanoseconds.toString().padStart(9, '0')
                when {
                    nanoseconds % 1_000_000_000_000_000 == 0
            }
        appendRange(nss, 0, 15)
        nanoseconds % 1_000_000_000_000_000_000 == 0
        nanoseconds.toString().padStart(9, '0')
}
-> append(nss)
 | n | n | n | n | n // constructing from number of units
 | n // extension functions
 */
 Returns a [Duration] equal to this [Int] number of the specified
 [unit]. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public fun Int.toDuration(unit: DurationUnit): Duration =
toDouble().toDuration(unit)
 */
 Returns a [Duration] equal to this [Long] number of the specified
 [unit].
 @SinceKotlin("1.3")
 @ExperimentalTime
 public fun Long.toDuration(unit: DurationUnit): Duration =
toDouble().toDuration(unit)
 */
 Returns a [Duration] equal to this [Double] number of the specified
 [unit].
 @SinceKotlin("1.3")
 @ExperimentalTime
 public fun Double.toDuration(unit: DurationUnit): Duration =
Duration(convertDurationUnit(this, unit, storageUnit))
 // constructing from number of units
 // extension properties
 */
 Returns a [Duration] equal to this [Int] number of nanoseconds.
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Int.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS)
 */
 Returns a [Duration] equal to this [Long] number of nanoseconds.
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Long.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS)
 */
 Returns a [Duration] equal to this [Double] number of
 nanoseconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Double.nanoseconds get() =
toDuration(DurationUnit.NANOSECONDS)
 */
 Returns a [Duration] equal to this [Int] number of
 nanoseconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Int.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)
 */
 Returns a [Duration] equal to this [Long] number of
 nanoseconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Long.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)
 */
 Returns a [Duration] equal to this [Double] number of
 nanoseconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Double.microseconds get() =
toDuration(DurationUnit.MICROSECONDS)
 */
 Returns a [Duration] equal to this [Int] number of
 milliseconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Int.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)
 */
 Returns a [Duration] equal to this [Long] number of
 milliseconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Long.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)
 */
 Returns a [Duration] equal to this [Double] number of
 milliseconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Double.milliseconds get() =
toDuration(DurationUnit.MILLISECONDS)
 */
 Returns a [Duration] equal to this [Int] number of
 seconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Int.seconds get() =
toDuration(DurationUnit.SECONDS)
 */
 Returns a [Duration] equal to this [Long] number of
 seconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Long.seconds get() =
toDuration(DurationUnit.SECONDS)
 */
 Returns a [Duration] equal to this [Double] number of
 seconds. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Double.seconds get() =
toDuration(DurationUnit.SECONDS)
 */
 Returns a [Duration] equal to this [Int] number of
 minutes. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Int.minutes get() =
toDuration(DurationUnit.MINUTES)
 */
 Returns a [Duration] equal to this [Long] number of
 minutes. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Long.minutes get() =
toDuration(DurationUnit.MINUTES)
 */
 Returns a [Duration] equal to this [Double] number of
 minutes. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Double.minutes get() =
toDuration(DurationUnit.MINUTES)
 */
 Returns a [Duration] equal to this [Int] number of
 hours. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Int.hours get() =
toDuration(DurationUnit.HOURS)
 */
 Returns a [Duration] equal to this [Long] number of
 hours. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Long.hours get() =
toDuration(DurationUnit.HOURS)
 */
 Returns a [Duration] equal to this [Double] number of
 hours. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Double.hours get() =
toDuration(DurationUnit.HOURS)
 */
 Returns a [Duration] equal to this [Int] number of
days. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Int.days get() =
toDuration(DurationUnit.DAYS)
 */
 Returns a [Duration] equal to this [Long] number of
days. *
 @SinceKotlin("1.3")
 @ExperimentalTime
 public val Long.days get() =
toDuration(DurationUnit.DAYS)\n\n/** Returns a [Duration] equal to this [Double] number of days. */
@SinceKotlin("1.3")
@ExperimentalTime
public Double days\nget() =
toDuration(DurationUnit.DAYS)\n\n/** Returns a duration whose value is the specified [duration] value multiplied by this number. */
@SinceKotlin("1.3")
@ExperimentalTime
public inline fun\nDouble times(duration: Duration)\n= duration * this\n\n/** Returns a duration whose value is the specified [duration] value multiplied by this number. */
@SinceKotlin("1.3")
@ExperimentalTime
public inline fun\nDouble times(duration: Duration)\n= duration * this\n\ninternal expect fun\nformatToExactDecimals(value: Double, decimals: Int): String\ninternal expect fun formatUpToDecimals(value: Double, decimals: Int): String\n
/**
 * Returns a [Duration] equal to this [Double] number of days.
 */
@SinceKotlin("1.3")
@ExperimentalTime
public Double days\ng{} =
toDuration(DurationUnit.DAYS)\n\n/** Returns a duration whose value is the specified [duration] value multiplied by this number. */
@SinceKotlin("1.3")
@ExperimentalTime
public inline fun\nDouble times(duration: Duration)\n= duration * this\n\n/** Returns a duration whose value is the specified [duration] value multiplied by this number. */
@SinceKotlin("1.3")
@ExperimentalTime
public inline fun\nDouble times(duration: Duration)\n= duration * this\n\ninternal expect fun\nformatToExactDecimals(value: Double, decimals: Int): String\ninternal expect fun formatUpToDecimals(value: Double, decimals: Int): String\n
/**
 * The list of possible time measurement units, in which a duration can be expressed.
 * The smallest time unit is [NANOSECONDS] and the largest is [DAYS], which corresponds to exactly 24
 * HOURS.
 */
@SinceKotlin("1.3")
@ExperimentalTime
public enum class DurationUnit {
    NANOSECONDS, MICROSECONDS, MILLISECONDS, SECONDS, MINUTES, HOURS, DAYS;
}

/** Converts the given time duration [value] expressed in the specified [sourceUnit] into the specified
 * [targetUnit].
 */
@SinceKotlin("1.3")
@ExperimentalTime
internal fun\nDurationUnit.convertDurationUnit(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double

@SinceKotlin("1.3")
@ExperimentalTime
internal fun\nDurationUnit.shortName(): String = when\n(this) {
    DurationUnit.NANOSECONDS -> "ns"
    DurationUnit.MICROSECONDS -> "us"
    DurationUnit.MILLISECONDS -> "ms"
    DurationUnit.SECONDS -> "s"
    DurationUnit.MINUTES -> "m"
    DurationUnit.HOURS -> "h"
    DurationUnit.DAYS -> "d"
}
@SinceKotlin("1.3")
@ExperimentalTime
public interface TimeSource {
    /**
     * Marks a point in time on this time source.
     * The returned [TimeMark] instance encapsulates the captured time point and allows querying the duration of time interval [elapsed][TimeMark.elapsedNow] from that point.
     */
    public fun markNow(): TimeMark

    /**
     * The most precise time source available in the platform.
     * This time source returns its readings from a source of monotonic time when it is available in a target platform, and resorts to a non-monotonic time source otherwise.
     */
    public object Monotonic : TimeSource by MonotonicTimeSource {
        override fun toString(): String = MonotonicTimeSource.toString()
    }

    public companion object {

    }
}

/**
 * Represents a time point notched on a particular [TimeSource].
 * Remains bound to the time source it was taken from and allows querying for the duration of time elapsed from that point (see the function [elapsedNow]).
 * Represents a time point notched on a particular [TimeSource].
 * Remains bound to the time source it was taken from and allows querying for the duration of time elapsed from that point (see the function [elapsedNow]).
 * Represents a time point notched on a particular [TimeSource].
 * Remains bound to the time source it was taken from and allows querying for the duration of time elapsed from that point (see the function [elapsedNow]).
 */
@SinceKotlin("1.3")
@ExperimentalTime
public abstract class TimeMark {
    /**
     * Returns the amount of time passed from this mark measured with the time source from which this mark was taken.
     * Note that the value returned by this function can change on subsequent invocations.
     */
    public abstract fun elapsedNow(): Duration

    /**
     * Returns a time mark on the same time source that is ahead of this time mark by the specified [duration].
     * The returned time mark is more _late_ when the [duration] is positive, and more _early_ when the [duration] is negative.
     */
    public open operator fun plus(duration: Duration): TimeMark = AdjustedTimeMark(this, duration)

    /**
     * Returns a time mark on the same time source that is behind this time mark by the specified [duration].
     * The returned time mark is more _early_ when the [duration] is positive, and more _late_ when the [duration] is negative.
     */
    public open operator fun minus(duration: Duration): TimeMark = AdjustedTimeMark(this, -duration)

    /**
     * Returns true if this time mark has passed according to the time source from which this mark was taken.
     * Note that the value returned by this function can change on subsequent invocations.
     */
    public fun hasPassedNow(): Boolean = !elapsedNow().isNegative()

    /**
     * Returns false if this time mark has not passed according to the time source from which this mark was taken.
     * Note that the value returned by this function can change on subsequent invocations.
     */
    public fun hasNotPassedNow(): Boolean = elapsedNow().isNegative()
}

@ExperimentalTime
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Deprecated("Subtracting one TimeMark from another is not a well defined operation because these time marks could have been obtained from the different time sources.", level = DeprecationLevel.ERROR)
@Suppress("UNUSED_PARAMETER")
public inline operator fun TimeMark.minus(other: TimeMark): Duration = throw Error("Operation is disallowed.")

@ExperimentalTime
@SinceKotlin("1.3")
@kotlin.internal.InlineOnly
@Deprecated("Comparing one TimeMark to another is not a well defined operation because these time marks could have been obtained from the different time sources.", level = DeprecationLevel.ERROR)
@Suppress("UNUSED_PARAMETER")
public inline operator fun TimeMark.compareTo(other: TimeMark): Int = throw Error("Operation is disallowed.")

private class AdjustedTimeMark(val mark: TimeMark, val adjustment: Duration) : TimeMark() {
    override fun elapsedNow(): Duration = mark.elapsedNow() - adjustment

    override fun plus(duration: Duration): TimeMark = AdjustedTimeMark(mark, adjustment + duration)
}

@SinceKotlin("1.3")
@ExperimentalTime
private class AdjustedTimeMark(val mark: TimeMark, val adjustment: Duration) : TimeMark() {
    override fun elapsedNow(): Duration = mark.elapsedNow() - adjustment

    override fun plus(duration: Duration): TimeMark = AdjustedTimeMark(mark, adjustment + duration)
}
return their readings as [Long] values in the specified [unit].

```kotlin
@nullable
@SinceKotlin("1.3")
@ExperimentalTime
public abstract class AbstractLongTimeSource(protected val unit: DurationUnit) : TimeSource {
    /**
     * This protected method should be overridden to return the current reading of the time source expressed as a [Long] number in the unit specified by the [unit] property.
     */
    protected abstract fun read(): Long

    private class LongTimeMark(private val startedAt: Long, private val timeSource: AbstractLongTimeSource, private val offset: Duration) : TimeMark() {
        override fun elapsedNow(): Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset
        override fun plus(duration: Duration): TimeMark = LongTimeMark(startedAt, timeSource, offset + duration)
    }

    override fun markNow(): TimeMark = LongTimeMark(read(), this, Duration.ZERO)
}

/**
 * An abstract class used to implement time sources that return their readings as [Double] values in the specified [unit].
 *
 * @property unit The unit in which this time source's readings are expressed.
 */
@SinceKotlin("1.3")
@ExperimentalTime
public abstract class AbstractDoubleTimeSource(protected val unit: DurationUnit) : TimeSource {
    /**
     * This protected method should be overridden to return the current reading of the time source expressed as a [Double] number in the unit specified by the [unit] property.
     */
    protected abstract fun read(): Double

    private class DoubleTimeMark(private val startedAt: Double, private val timeSource: AbstractDoubleTimeSource, private val offset: Duration) : TimeMark() {
        override fun elapsedNow(): Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset
        override fun plus(duration: Duration): TimeMark = DoubleTimeMark(startedAt, timeSource, offset + duration)
    }

    override fun markNow(): TimeMark = DoubleTimeMark(read(), this, Duration.ZERO)
}

/**
 * A time source that has programmatically updatable readings. It is useful as a predictable source of time in tests.
 *
 * The current reading value can be advanced by the specified duration amount with the operator [plusAssign]:
 *
 * ```
 * val timeSource = TestTimeSource()
 * timeSource += 10.seconds
 * ```
 *
 * Implementation note: the current reading value is stored as a [Long] number of nanoseconds; thus it's capable to represent a time range of approximately ±292 years.
 *
 * Should the reading value overflow as the result of [plusAssign] operation, an [IllegalStateException] is thrown.
 */
@SinceKotlin("1.3")
@ExperimentalTime
public class TestTimeSource : AbstractLongTimeSource(unit = DurationUnit.NANOSECONDS) {
    private var reading: Long = 0L

    override fun read(): Long = reading

    public operator fun plusAssign(duration: Duration) {
        val delta = duration.toDouble(unit)
        val longDelta = delta.toLong()
        reading = if (longDelta != Long.MIN_VALUE && longDelta != Long.MAX_VALUE) {
            val newReading = reading + longDelta
            if (reading xor longDelta >= 0 && reading xor newReading < 0)
                overflow(duration)
                newReading
            else
                reading + longDelta
        } else {
            val newReading = reading + delta
            if (newReading > Long.MAX_VALUE || newReading < Long.MIN_VALUE)
                overflow(duration)
                newReading
            else
                reading + longDelta
        }
    }

    private fun overflow(duration: Duration) {
        throw IllegalStateException("TestTimeSource will overflow if its reading $reading is advanced by $duration.")
    }
}
```

@SinceKotlin("1.3")
@ExperimentalTime
@Deprecated("Use TimeSource.Monotonic instead.", ReplaceWith("TimeSource.Monotonic", "kotlin.time.TimeSource"), DeprecationLevel.ERROR)
public typealias MonoClock = TimeSource.Monotonic

@SinceKotlin("1.3")
@ExperimentalTime
@Deprecated("Use AbstractLongTimeSource instead.", ReplaceWith("AbstractLongTimeSource", "kotlin.time.AbstractLongTimeSource"), DeprecationLevel.ERROR)
public typealias AbstractLongClock = AbstractLongTimeSource

@SinceKotlin("1.3")
@ExperimentalTime
@Deprecated("Use AbstractDoubleTimeSource instead.", ReplaceWith("AbstractDoubleTimeSource", "kotlin.time.AbstractDoubleTimeSource"), DeprecationLevel.ERROR)
public typealias AbstractDoubleClock = AbstractDoubleTimeSource

Open Source Used In DNAC Platform Antelope 14236
"kotlin.time.AbstractDoubleTimeSource"), DeprecationLevel.ERROR)\n\n@ExperimentalTime
\n@Deprecated("Use TestTimeSource instead.", ReplaceWith("TestTimeSource"), "kotlin.time.TestTimeSource")\n
package kotlin.time

import kotlin.concurrent.

/**
 * Data class representing a result of executing an action, along with the duration of elapsed time interval.
 * @property value the result of the action.
 * @property duration the time elapsed to execute the action.
 */
@ExperimentalTime
public data class TimedValue<T>(val value: T, val duration: Duration)

/**
 * Executes the given function [block] and returns an instance of TimedValue class, containing both
 * the result of function execution and the duration of elapsed time interval.
 * The elapsed time is measured with the specified `this` [TimeSource] instance.
 */
@ExperimentalTime
public inline fun TimeSource.measureTimedValue(block: () -> T): TimedValue<T> {
    contract {
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)
    }

    val mark = markNow()
    val result = block()
    return TimedValue(result, mark.elapsedNow())
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.
 */

package kotlin

import kotlin.coroutines.

/**
 * Defines deep recursive function that keeps its stack on the heap,
 * which allows very deep recursive computations that do not use the actual call stack.
 * To initiate a call to this deep recursive function use its invoke function.
 * As a rule of thumb, it should be used if recursion goes deeper than a thousand calls.
 * The DeepRecursiveFunction takes one parameter of type [T] and returns a result of type [R].
 * The [block] of code defines the body of a recursive function. In this block
 * [callRecursive] function can be used to make a recursive call.
 * Other instances of DeepRecursiveFunction can be called in this scope with
 * `callRecursive('extension', too).` * For example, take a look at the following recursive tree class and a deeply
 * recursive instance of this tree with 100K nodes:
 *  
 * Tree(val left: Tree? = null, val right: Tree? = null)
 * val deepTree = generateSequence(Tree()) { Tree(it) }.take(100_000).last()
 * println(depth(deepTree)) // StackOverflowError
 * 
 * However, the `depth` function can be rewritten using `DeepRecursiveFunction` in the following way, and then
 * it successfully computes `depth(deepTree)`:
 * 
 * StackOverflowError
 * 
 * If this `depth` function is called for a `deepTree` it produces [StackOverflowError] because of deep recursion.
 * However, the `depth` function can be rewritten using `DeepRecursiveFunction` in the following way, and then
 * it successfully computes `depth(deepTree)`:
 * 
 * StackOverflowError
 * 
 * Deep recursive functions can also...
mutually call each other using a heap for the stack via

```
val even: DeepRecursiveFunction { t -> n * if (t == null) 0 else odd.callRecursive(t.right) + odd.callRecursive(t.left) + 1
}
val odd: DeepRecursiveFunction { t -> n * if (t == null) 0 else even.callRecursive(t.right) + even.callRecursive(t.left) + 1
}
```

The following pair of mutually recursive functions computes the number of tree nodes at even depth in the tree.

```
val mutualRecursion = object {
    val even: DeepRecursiveFunction { t -> n * if (t == null) 0 else odd.callRecursive(t.left) + odd.callRecursive(t.right) + 1
    }
    val odd: DeepRecursiveFunction { t -> n * if (t == null) 0 else even.callRecursive(t.right) + even.callRecursive(t.left) + 1
    }
}
```

@SinceKotlin("1.4")
@ExperimentalStdlibApi
public class DeepRecursiveFunction<T, R> {
    internal val block: suspend DeepRecursiveScope<T, R>.(T) -> R

    /**
     * Initiates a call to this deep recursive function,
     * forming a root of the call tree.
     *
     * This operator should not be used from inside of [DeepRecursiveScope] as it uses
     * the call stack slot for * initial recursive invocation. From inside of [DeepRecursiveScope] use
     * [callRecursive][DeepRecursiveScope.callRecursive].
     */
    public operator fun <T, R> DeepRecursiveFunction<T, R>.invoke(value: T): R =
        DeepRecursiveScopeImpl<T, R>(block, value).runCallLoop()

    /**
     * A scope class for [DeepRecursiveFunction] function declaration that defines [callRecursive] methods to
     * recursively call this function or another [DeepRecursiveFunction] putting the call activation frame on the heap.
     */
    public sealed class DeepRecursiveScope<T, R> {
        /**
         * Makes recursive call to this [DeepRecursiveFunction] function putting
         * the call activation frame on the heap, * as opposed to the actual call stack that is used by a regular recursive
         * call.
         */
        public abstract suspend fun callRecursive(value: T): R

        /**
         * Makes call to the specified [DeepRecursiveFunction] function putting the call activation frame on the heap,
         * * as opposed to the actual call stack that is used by a regular call.
         */

        @Deprecated(
            level = DeprecationLevel.ERROR,
            message = "'invoke' should not be called from DeepRecursiveScope. Use 'callRecursive' to do recursion in
            the heap instead of the call stack."
        )
        @Suppress("UNUSED_PARAMETER")
        public operator fun DeepRecursiveFunction<*, *>.invoke(value: Any?): Nothing =
            throw UnsupportedOperationException("Should not be called from DeepRecursiveScope")
    }
}

// Implementation

@ExperimentalStdlibApi
private typealias DeepRecursiveFunctionBlock = suspend DeepRecursiveScope<*, *>.(Any?) -> Any?

@ExperimentalStdlibApi
private val UNDEFINED_RESULT = Result.success(COROUTINE_SUSPENDED)

@ExperimentalStdlibApi
@RestrictsSuspension
@ExperimentalStdlibApi
private class DeepRecursiveScopeImpl<T, R>(
    block: suspend DeepRecursiveScope<T, R>.(T) -> R,
    value: T
) : DeepRecursiveScope<T, R>(), Continuation<R> {
    // Active function block
    private var function: DeepRecursiveFunctionBlock = block as DeepRecursiveFunctionBlock

    // Value to call function
    private var value: Any? = value

    // Continuation of the current call
    private var cont: Continuation<Any?>? = this as Continuation<Any?>

    // Completion result (completion of the whole call stack)
    private var result: Result<Any?> = UNDEFINED_RESULT

    // Continuation result (completion of the whole call stack)
    private var value: Any? = value

    // Continoation of the current call
    private var cont: Continuation<Any?>? = this as Continuation<Any?>

    // Get the coroutine context
    private get() = EmptyCoroutineContext

    // Override function resumeWith
    override fun resumeWith(result: Result:<R>) {
        this.cont = null
        this.result = result
    }

    // Override suspend fun callRecursive
    suspend fun callRecursive(value: T): R =
        suspendCoroutineUninterceptedOrReturn { cont ->
            calling the same function that is currently active
            this.cont = cont as Continuation<Any?>
            this.value = value
            COROUTINE_SUSPENDED
        }

    // Override suspend fun <U, S> callRecursive
    suspend fun <U, S> callRecursive(value: U): S =
        suspendCoroutineUninterceptedOrReturn { cont ->
            calling another recursive function
            block as DeepRecursiveFunctionBlock
            with(this@DeepRecursiveScopeImpl) {
                val currentFunction = this.function
                if (function !== currentFunction) {
                    this.function = function
                    this.cont =
```
** Returns the current version of the Kotlin standard library.

```java
private fun version(major: Int, minor: Int, patch: Int): Int {
    return major shl 16 + minor shl 8 + patch
}
```

* Returns the string representation of this

```java
var result: String
    return result
```

* Represents a version of the Kotlin standard library.

```java
fun toString(): String = "$major.$minor.$patch"
```

* Use of this source code is governed by the

```java
@SinceKotlin("1.1")
    public constructor(major: Int, minor: Int) : this(major, minor, 0)

    private val
```
*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.countLeadingZeroBits(): Int\n\n*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.countTrailingZeroBits(): Int\n\n*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.takeHighestOneBit(): Int\n\n*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.takeLowestOneBit(): Int\n\n*\@SinceKotlin("1.4\")\@ExperimentalStdlibApi\npublic fun Int.rotateLeft(bitCount: Int): Int\n\n*\@SinceKotlin("1.3\")\@ExperimentalStdlibApi\npublic fun Int.rotateRight(bitCount: Int): Int\n\n*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countOneBits(): Int\n\n*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countLeadingZeroBits(): Int\n\n*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countTrailingZeroBits(): Int\n\n*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.takeHighestOneBit(): Long\n\n*\@SinceKotlin("1.4\")\@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.takeLowestOneBit(): Long\n\n*\@SinceKotlin("1.4\")\@ExperimentalStdlibApi\npublic fun Long.rotateLeft(bitCount: Int): Long\n\n*\@SinceKotlin("1.3\")\@ExperimentalStdlibApi\npublic fun Long.rotateRight(bitCount: Int): Long\n\nOpen Source Used In DNAC Platform Antelope 14241
Long.rotateRight(bitCount: Int): Long\n* Counts the number of set bits in the binary representation of this [Byte] number.

*\n
/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Byte] number.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun Byte.countOneBits(): Int = (toInt() and 0xFF).countOneBits()

/**
 * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Byte] number.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun Byte.countLeadingZeroBits(): Int = (toInt() and 0xFF).countLeadingZeroBits() - (Int.SIZE_BITS - Byte.SIZE_BITS)

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [Byte] number,\n* or zero, if this number is zero.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun Byte.takeHighestOneBit(): Byte = (toInt() and 0xFF).takeHighestOneBit().toByte()

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [Byte] number,\n* or zero, if this number is zero.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun Byte.takeLowestOneBit(): Byte = toInt().takeLowestOneBit().toByte()

/**
 * Rotates the binary representation of this [Byte] number left by the specified [bitCount] number of bits,\n* The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side,\n* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count: this number.rotateLeft(-n) == this number.rotateRight(n) * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally this number.rotateLeft(n % 8) ==
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun Byte.rotateLeft(bitCount: Int): Byte =
    toInt().shl(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7)).toInt().toByte()

/**
 * Rotates the binary representation of this [Byte] number right by the specified [bitCount] number of bits,\n* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side,\n* Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count: this number.rotateRight(-n) == this number.rotateLeft(n) * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally this number.rotateRight(n % 8) ==
 */
@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun Byte.rotateRight(bitCount: Int): Byte =
    toInt().shr(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7)).toInt().toByte()

/**
 * Counts the number of set bits in the binary representation of this [Short] number.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun Short.countOneBits(): Int = (toInt() and 0xFFFF).countOneBits()

/**
 * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Short] number.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun Short.countLeadingZeroBits(): Int = (toInt() and 0xFFFF).countLeadingZeroBits() - (Int.SIZE_BITS - Short.SIZE_BITS)

/**
 * Returns a number having a single bit set in the position of the most significant set bit of this [Short] number,\n* or zero, if this number is zero.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun Short.takeHighestOneBit(): Short = (toInt() and 0xFFFF).takeHighestOneBit().toShort()

/**
 * Returns a number having a single bit set in the position of the least significant set bit of this [Short] number,\n* or zero, if this number is zero.
 */
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@kotlin.internal.InlineOnly
public inline fun Short.takeLowestOneBit(): Short = (toInt() and 0xFFFF).takeLowestOneBit().toShort()
zero, if this number is zero.

```kotlin
@SinceKotlin("1.4")
@WasExperimental(ExperimentalStdlibApi::class)
@IntrinsicsVisible
@receiver
@inline fun Short.rotateLeft(bitCount: Int): Short =

@SinceKotlin("1.3")
@ExperimentalStdlibApi
public fun Short.rotateRight(bitCount: Int): Short =

@file:kotlin.jvm.JvmName("TuplesKt")

package kotlin

/**
 * Represents a generic pair of two values.
 *
 * There is no meaning attached to values in this class, it can be used for any purpose.
 * Pair exhibits value semantics, i.e. two pairs are equal if both components are equal.
 */
public data class Pair<out A, out B>(
    public val first: A,
    public val second: B
) : Serializable {

    /**
     * Returns string representation of the [Pair] including its [first] and [second] values.
     */
    public override fun toString(): String = "($first, $second)"
}

/**
 * Converts this pair into a list.
 */
public fun <T> Pair<T, T>.toList(): List<T> = listOf(first, second)

/**
 * Represents a triad of values
 *
 * There is no meaning attached to values in this class, it can be used for any purpose.
 */
public data class Triple<out A, out B, out C>(
    public val first: A,
    public val second: B,
    public val third: C
) : Serializable {

    /**
     * Returns string representation of the [Triple] including its [first], [second] and [third] values.
     */
    public override fun toString(): String = "($first, $second, $third)"
}

/**
 * Converts this triple into a list.
 */
public fun <T> Triple<T, T, T>.toList(): List<T> = listOf(first, second, third)
```
Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.

package kotlin.ranges

import kotlin.internal.*

/**
 * A range of values of type `UInt`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public class UIntRange(start: UInt, endInclusive: UInt) : UIntProgression(start, endInclusive, 1), ClosedRange<UInt> {
    override val start: UInt get() = first
    override val endInclusive: UInt get() = last

    override fun contains(value: UInt): Boolean = first <= value && value <= last

    override fun equals(other: Any?): Boolean =
    other is UIntRange && (isEmpty() && other.isEmpty() ||
        first == other.first && last == other.last &&
        override fun hashCode(): Int =
            if (isEmpty()) -1 else (31 * first.toInt() + last.toInt())

    override fun toString(): String = "$first..$last"

    companion object {
        /**
         * An empty range of values of type UInt. */
        public val EMPTY: UIntRange = UIntRange(UInt.MAX_VALUE, UInt.MIN_VALUE)
    }
}

/**
 * A progression of values of type `UInt`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public open class UIntProgression
internal constructor(
    start: UInt,
    endInclusive: UInt,
    step: Int
) : Iterable<UInt> {
    init {
        if (step == 0.toInt()) throw kotlin.IllegalArgumentException("Step must be non-zero.")
        if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Int.MIN_VALUE to avoid overflow on negation.")
    }

    /**
     * The first element in the progression.
     */
    public val first: UInt = start

    /**
     * The last element in the progression.
     */
    public val last: UInt = getProgressionLastElement(start, endInclusive, step)

    /**
     * The step of the progression.
     */
    public val step: Int = step

    override fun iterator(): UIntIterator =UIntProgressionIterator(first, last, step)

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last

    override fun equals(other: Any?): Boolean =
        other is UIntProgression && (isEmpty() && other.isEmpty() ||
            first == other.first && last == other.last && step == other.step)

    override fun toString(): String = if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"

    companion object {
        /**
         * Creates UIntProgression within the specified bounds of a closed range.
         * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].
         * In order to go backwards the [step] must be negative.
         * @param rangeStart the start value of the range.
         * @param rangeEnd the end value of the range, not excluding it.
         * @param step the number by which the value is incremented on each step.
         */
        public fun fromClosedRange(rangeStart: UInt, rangeEnd: UInt, step: Int): UIntProgression = UIntProgression(rangeStart, rangeEnd, step)
    }
}

/**
 * An iterator over a sequence of values of type `UByte`.
 * @property step the number by which the value is incremented on each step.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UByteIterator : Iterator<UByte> {
    override final fun next() = nextUByte()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextUByte(): UByte
}

/** An iterator over a sequence of values of type `UShort`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UShortIterator : Iterator<UShort> {
    override final fun next() = nextUShort()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextUShort(): UShort
}

/** An iterator over a sequence of values of type `UInt`.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class UIntIterator : Iterator<UInt> {
    override final fun next() = nextUInt()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextUInt(): UInt
}

Open Source Used In DNAC Platform Antelope 14244
abstract fun nextUInt(): UInt

/** An iterator over a sequence of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class ULongIterator : Iterator<ULong> {

    override final fun next() = nextULong()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextULong(): ULong
}

/** An iterator over a sequence of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public abstract class ULongIterator : Iterator<ULong> {

    override final fun next() = nextULong()

    /** Returns the next value in the sequence without boxing. */
    public abstract fun nextULong(): ULong
}

/*
 * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file. */

package kotlin.ranges

import kotlin.internal.*

/**
 * A range of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public class ULongRange(start: ULong, endInclusive: ULong) : ULongProgression(start, endInclusive, 1), ClosedRange<ULong> {
    override val start: ULong get() = first
    override val endInclusive: ULong get() = last

    override fun contains(value: ULong): Boolean = first <= value && value <= last

    override fun isEmpty(): Boolean = first > last

    override fun equals(other: Any?): Boolean =
        other is ULongRange && (isEmpty() && other.isEmpty() ||
                first == other.first && last == other.last)

    companion object {

        /** An empty range of values of type ULong. */
        public val EMPTY: ULongRange = ULongRange(ULong.MAX_VALUE, ULong.MIN_VALUE)
    }
}

/**
 * A progression of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public open class ULongProgression
internal constructor(
    start: ULong,
    endInclusive: ULong,
    step: Long
) :
    Iterable<ULong> {
    init {
        if (step == 0.toLong()) throw kotlin.IllegalArgumentException("Step must be non-zero.")
        if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException("Step must be greater than Long.MIN_VALUE to avoid overflow on negation.")
    }

    /**
     * The first element in the progression. */
    public val first: ULong = start

    /**
     * The last element in the progression. */
    public val last: ULong = getProgressionLastElement(start, endInclusive, step)

    /**
     * The step of the progression. */
    public val step: Long = step

    override fun iterator(): ULongIterator = ULongProgressionIterator(first, last, step)

    /** Checks if the progression is empty. */
    public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last

    override fun equals(other: Any?): Boolean =
        other is ULongProgression && (isEmpty() && other.isEmpty() ||
                first == other.first && last == other.last)

    override fun hashCode(): Int =
        if (isEmpty()) -1 else (31 * (first xor (first shr 32)).toInt() + (last xor (last shr 32)).toInt())

    override fun toString(): String = if (step > 0) "$first..$last step $step" else "$first downTo $last step ${-step}"

    companion object {

        /**
         * Creates ULongProgression within the specified bounds of a closed range. */
        public fun fromClosedRange(rangeStart: ULong, rangeEnd: ULong, step: Long): ULongProgression = ULongProgression(rangeStart, rangeEnd, step)
    }
}

/**
 * An iterator over a progression of values of type `ULong`. */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
private class ULongProgressionIterator(first: ULong, last: ULong, step: Long) : ULongIterator() {
    private val finalElement = last
    private var hasNext: Boolean = if (step > 0) first <= last else first >= last
    private val step = step.toULong() // use 2-complement math for negative steps

    override fun hasNext(): Boolean = hasNext

    override fun nextULong(): ULong {
        val value = next
        if (value == finalElement) {
            if (!hasNext) throw kotlin.NoSuchElementException()
            hasNext = false
        } else {
            next += step
        }
        return value
    }
}

/*
 * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file. */

package kotlin.math

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun min(a: UInt, b: UInt): UInt {
    return minOf(a, b)
}

/**
 * Returns the smaller of two values.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
@kotlin.internal.InlineOnly
public inline fun min(a: UInt, b: UInt): UInt {
    return minOf(a, b)
}
number as the least significant bits on the right side.

* Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:
  
  \[ \text{number.rotateLeft}(-n) = \text{number.rotateRight}(n) \]

* Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally:
  
  \[ \text{number.rotateLeft}(n) = \text{number.rotateLeft}(n \mod 8) \]

* The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.
  
  \[ \text{number.rotateRight}(-n) = \text{number.rotateLeft}(n) \]

* Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally:
  
  \[ \text{number.rotateRight}(n) = \text{number.rotateRight}(n \mod 8) \]

* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally:
  
  \[ \text{number.rotateLeft}(n) = \text{number.rotateLeft}(n \mod 16) \]

* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally:
  
  \[ \text{number.rotateRight}(-n) = \text{number.rotateLeft}(n) \]

* Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally:
  
  \[ \text{number.rotateRight}(n) = \text{number.rotateRight}(n \mod 16) \]
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.

```kotlin
/*
 * Copyright 2010-2018 JetBrains s.r.o.
 * and Kotlin Programming Language contributors.
 * Use of this source code is governed by the Apache 2.0 license
 * that can be found in the license/LICENSE.txt file.
 */

@file:kotlin.jvm.JvmName("UStringsKt")  // string representation of unsigned numbers

package kotlin.text

/**
 * Returns a string representation of this [Byte] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
//@kotlin.internal.InlineOnly
public /*inline*/ fun UByte.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Short] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
//@kotlin.internal.InlineOnly
public /*inline*/ fun UShort.toString(radix: Int): String = this.toInt().toString(radix)

/**
 * Returns a string representation of this [Int] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
//@kotlin.internal.InlineOnly
public /*inline*/ fun UInt.toString(radix: Int): String = this.toLong().toString(radix)

/**
 * Returns a string representation of this [Long] value in the specified [radix].
 * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
//@kotlin.internal.InlineOnly
public /*inline*/ fun ULong.toString(radix: Int): String = ulongToString(this, checkRadix(radix))
```

```kotlin
// (a - b) mod c
private fun differenceModulo(a: UInt, b: UInt, c: UInt): UInt {
    val ac = a % c
    val bc = b % c
    return if (ac >= bc) ac - bc else ac - bc + c
}

private fun differenceModulo(a: ULong, b: ULong, c: ULong): ULong {
    val ac = a % c
    val bc = b % c
    return if (ac >= bc) ac - bc else ac - bc + c
}
```

```kotlin
/**
 * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative n * [step].
 * No validation on passed parameters is performed. The given parameters should satisfy the condition: n * - either `step > 0` and `start <= end`, n * - or `step < 0` and `start >= end`. n * @param start first element of the progression n * @param end ending bound for the progression n * @param step increment, or difference of successive elements in the progression n * @return the final element of the progression n * @suppress
 */
@PublishedApi
@SinceKotlin("1.3")
internal fun getProgressionLastElement(start: UInt, end: UInt, step: Int): UInt = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toUInt())
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toUInt())
    else -> throw kotlin.IllegalArgumentException("Step is zero.")
}

/**
 * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative n * [step].
 * No validation on passed parameters is performed. The given parameters should satisfy the condition: n * - either `step > 0` and `start <= end`, n * - or `step < 0` and `start >= end`. n * @param start first element of the progression n * @param end ending bound for the progression n * @param step increment, or difference of successive elements in the progression n * @return the final element of the progression n * @suppress
 */
@PublishedApi
@SinceKotlin("1.3")
internal fun getProgressionLastElement(start: ULong, end: ULong, step: Long): ULong = when {
    step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toULong())
    step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toULong())
    else -> throw kotlin.IllegalArgumentException("Step is zero.")
}
```
number conversion.

```kotlin
public fun String.toUByte(radix: Int): UByte = toUByteOrNull(radix) ?: numberFormatError(this)

/**
 * Parses the string as a [UShort] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUShort(): UShort =
    toUShortOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a [UShort] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUShort(radix: Int): UShort =
    toUShortOrNull(radix) ?: numberFormatError(this)

/**
 * Parses the string as a [UInt] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUInt(): UInt =
    toUIntOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a [UInt] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUInt(radix: Int): UInt =
    toUIntOrNull(radix) ?: numberFormatError(this)

/**
 * Parses the string as a [ULong] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toULong(): ULong =
    toULongOrNull() ?: numberFormatError(this)

/**
 * Parses the string as a [ULong] number and returns the result.
 * @throws NumberFormatException if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toULong(radix: Int): ULong =
    toULongOrNull(radix) ?: numberFormatError(this)

/**
 * Parses the string as an [UByte] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUByteOrNull(): UByte? =
    toUByteOrNull(radix = 10) ?: numberFormatError(this)

/**
 * Parses the string as an [UByte] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUByteOrNull(radix: Int): UByte? {
    val int = this.toUIntOrNull(radix) ?: return null
    if (int > UByte.MAX_VALUE) return null
    return int.toUByte()
}

/**
 * Parses the string as an [UShort] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUShortOrNull(): UShort? =
    toUShortOrNull(radix = 10) ?: numberFormatError(this)

/**
 * Parses the string as an [UShort] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUShortOrNull(radix: Int): UShort? {
    val int = this.toUIntOrNull(radix) ?: return null
    if (int > UShort.MAX_VALUE) return null
    return int.toUShort()
}

/**
 * Parses the string as an [UInt] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUIntOrNull(): UInt? =
    toUIntOrNull(radix = 10) ?: numberFormatError(this)

/**
 * Parses the string as an [ UInt] number and returns the result
 * or `null` if the string is not a valid representation of a number.
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toUIntOrNull(radix: Int): UInt? {
    checkRadix(radix)

    val length = this.length
    if (length == 0)
        return null

    val limit: UInt = UInt.MAX_VALUE
    val start: Int

    val firstChar = this[0]
    if (firstChar < '0') {
        if (length == 1 || firstChar != '+') return null
        start = 1
    } else {
        start = 0
    }

    val limitForMaxRadix = 119304647u  //  limit / 36
    var result = 0u
    for (i in start until length) {
        val digit = digitOf(this[i], radix)

        if (digit < 0) return null
        if (result > limitBeforeMul) {
            if (limitBeforeMul == limitForMaxRadix) {
                result = 0u
            } else {
                result = result * radix + digit
            }
        } else {
            result += digit
        }
    }

    return result
}
```

Open Source Used In DNAC Platform Antelope 14250
limitBeforeMul = limit / uradix

if (result > limitBeforeMul) {
    return null
}

result *= uradix

val beforeAdding = result
    result += digitOfInt(result)
    if (result > limitBeforeMul) {
        if (limitBeforeMul == limitForMaxRadix) {
            limitBeforeMul = limit / uradix
            if (result > limitBeforeMul) {
                return null
            }
        } else {
            return null
        }
    }

result += digit.toUInt()

if (result < beforeAdding) return null // overflow has happened

return result
}

/**
 * Parses the string as an [ULong] number and returns the result or `null` if the string is not a valid representation of a number.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toULongOrNull(): ULong? = toULongOrNull(radix = 10)

/**
 * Parses the string as an [ULong] number and returns the result or `null` if the string is not a valid representation of a number.
 *
 * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.
 */
@SinceKotlin("1.3")
@ExperimentalUnsignedTypes
public fun String.toULongOrNull(radix: Int): ULong? {
    checkRadix(radix)

    val length = this.length
    if (length == 0) return null

    val limit: ULong = ULong.MAX_VALUE
    val start: Int

    val firstChar = this[0]
    if (firstChar < '0') {
        if (length == 1 || firstChar != '+') return null
        start = 1
    } else {
        start = 0
    }

    val limitForMaxRadix = 51240957603043100uL  //  limit / 36
    var limitBeforeMul = limitForMaxRadix
    val uradix = radix.toULong()
    var result = 0uL
    for (i in start until length) {
        val digit = digitOf(this[i], radix)

        if (digit < 0) return null
        if (result > limitBeforeMul) {
            if (limitBeforeMul == limitForMaxRadix) {
                limitBeforeMul = limit / uradix
                if (result > limitBeforeMul) {
                    return null
                }
            } else {
                return null
            }
        }

        result *= uradix

        val beforeAdding = result
        result += digit.toUInt()
        if (result < beforeAdding) return null // overflow has happened
    }

    return result
}
Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond to the rectangular coordinates `(x, y)` by computing the arc tangent of the value `y / x` in the range from `-PI` to `PI` radians. The returned value is an angle in the range from `-PI` to `PI` radians.

```kotlin
@SinceKotlin("1.2")
public expect fun atan2(y: Double, x: Double): Double
```

Computes the inverse hyperbolic sine of the value `x`. The returned value is `y` such that `sinh(y) == x`.

```kotlin
@SinceKotlin("1.2")
public expect fun asinh(x: Double): Double
```

Computes the arc cosine of the value `x`; the returned value is an angle in the range from `0.0` to `PI` radians.

```kotlin
@SinceKotlin("1.2")
public expect fun acos(x: Double): Double
```

Computes the arc tangent of the value `x`; the returned value is an angle in the range from `-PI/2` to `PI/2` radians.

```kotlin
@SinceKotlin("1.2")
public expect fun atan(x: Double): Double
```

Computes the hyperbolic cosine of the value `x`. The returned value is positive `y` such that `cosh(y) == x`.

```kotlin
@SinceKotlin("1.2")
public expect fun cosh(x: Double): Double
```

Computes the hyperbolic tangent of the value `x`. The returned value is `y` such that `tanh(y) == x`.

```kotlin
@SinceKotlin("1.2")
public expect fun tanh(x: Double): Double
```

Computes the logarithm of the value `x` to the given base.

```kotlin
@SinceKotlin("1.2")
public expect fun log(x: Double, base: Double): Double
```

Computes the logarithm (base `E`) of the value `x`.

```kotlin
@SinceKotlin("1.2")
public expect fun log(x: Double): Double
```

Computes the natural logarithm (base `E`) of the value `x`.

```kotlin
@SinceKotlin("1.2")
public expect fun ln(x: Double): Double
```

Computes Euler's number `e` raised to the power of the value `x`.

```kotlin
@SinceKotlin("1.2")
public expect fun exp(x: Double): Double
```

Computes `exp(x)`.

```kotlin
@SinceKotlin("1.2")
public expect fun sqrt(x: Double): Double
```

Computes the positive square root of the value `x`.

```kotlin
@SinceKotlin("1.2")
public expect fun hypot(x: Double, y: Double): Double
```

Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond to the rectangular coordinates `(x, y)`.
Computes the binary logarithm (base 2) of the value \( x \).\n
```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun log2(x: Double): Double
```

This function can be implemented to produce more precise result for \( x \) near zero.\n
```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun log1p(x: Double): Double
```

Rounds the given value \( x \) to an integer towards positive infinity.\n
```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun ceil(x: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun floor(x: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun truncate(x: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun round(x: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun abs(x: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun min(a: Double, b: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun max(a: Double, b: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun Double.pow(x: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun Double.pow(n: Int): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect val Double.absoluteValue: Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect val Double.sign: Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun Double.withSign(sign: Double): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun Double.withSign(sign: Int): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun Double.ulp: Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun Double.nextUp(): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun Double.nextDown(): Double
```

```kotlin
*\n@SinceKotlin("1.2")\npublic expect fun Double.nextTowards(y: Double): Double
```

Open Source Used In DNAC Platform Antelope 14253
Open Source Used In DNAC Platform Antelope 14254
*An@SinceKotlin("1.2") public expect fun hypot(x: Float, y: Float): Float;\n\n*An@SinceKotlin("1.2") public expect fun sqrt(x: Float): Float;\n\n/**
 * Computes the positive square root of the value \[x\].\n * \* Special cases:\n * - `sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN`;\n * - `sqrt(NaN)` is `NaN`\n * - `sqrt(-Inf)` is `0.0`\n * @see [exp] function;\n */\n*An@SinceKotlin("1.2") public expect fun exp(x: Float): Float;\n\n/**
 * Computes Euler’s number `e` raised to the power of the value \[x\].\n * \* Special cases:\n * - `exp(NaN)` is `NaN`\n * - `exp(+Inf)` is `+Inf`\n * - `exp(-Inf)` is `0.0`\n * @see [exp] function;\n */\n*An@SinceKotlin("1.2") public expect fun expm1(x: Float): Float;\n\n/**
 * Computes `expm1(x)`\n */\n*An@SinceKotlin("1.2") public expect fun log(x: Float, base: Float): Float;\n\n/**
 * Returns the natural logarithm (base `E`) of the value \[x\].\n * \* Special cases:\n * - `ln(NaN)` is `NaN`\n * - `ln(x)` is `NaN` when `x < 0.0`\n * - `ln(1.0)` is `-Inf`\n * - `ln(0.0)` is `-Inf`;\n * @see [ln] function for special cases;\n */\n*An@SinceKotlin("1.2") public expect fun ln1p(x: Float): Float;\n\n/**
 * Returns the absolute value of the given value \[x\].\n * Special cases:\n * - `abs(NaN)` is `NaN`\n * - `abs(b)` is `NaN`\n * @see absoluteValue extension property for \[Float\];\n */\n*An@SinceKotlin("1.2") public expect fun max(a: Float, b: Float): Float;\n\n/**
 * Computes `max(a, b)`\n */\n*An@SinceKotlin("1.2") public expect fun min(a: Float, b: Float): Float;\n\n/**
 * Computes `min(a, b)`\n */\n*An@SinceKotlin("1.2") public expect fun pow(x: Float): Float;\n\n/**
 * Computes `pow(x)`\n */\n
value.\n * Special cases:ln * - `NaN.absoluteValue` is `NaN`
 * Returns the sign of this value:ln * - `-1.0` if the value is negative,ln * - zero if the value is zero,ln * - `1.0` if the value is positive\n * Special cases:ln * - `NaN.sign` is `NaN\n *@SinceKotlin("1.2")\n *public expect val Float.absoluteValue: Float\n *\n *Special cases:\n * - `NaN.*\n *@SinceKotlin("1.2")\n *public expect fun Float.withSign(sign: Float): Float\n *\n * Returns this value with the sign bit same as of the [sign] value:\n * If [sign] is `NaN` the sign of the result is undefined.\n *\n *@SinceKotlin("1.2")\n *public expect fun Float.withSign(sign: Int): Float\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Int].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n * Special cases:\n * - `Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n
Open Source Used In DNAC Platform Antelope 14281
Open Source Used In DNAC Platform Antelope
Open Source Used In DNAC Platform Antelope 14393

qBAAO,OAyZH,OAzZG,C;YAAP,uB;WACqB,kBAAB,eAAa,iBAAO,CAAP,IAAb,C;UAAa+b,SBAmZB,OAXZ
y,B;UAA5C,ajB7qRO,W;Ui8B8qR,kBAuZmB,O;UAzZH,2B;UAHb,OAAbg,CAHAh,B;YAAa,b,Y;YACZ,cA
qZWB,SArZV,CAAU,WAav,EAaUB,OAAB,C;YACd,MAAO,WAAl,WAAlJ,C;UEAX,qBAAO,M,...;QAkZP,yB
:o;KAhBJ,C;iFAmBa,yB;MAIZA,gD;MAAA,gE;MAkZA,gD;QAQW,B;w,...;UAHbz,QAHHB,IAAI,mBAAJ,C;
YAAe,qBAAO,OAaMZH,OAaZG,C;YAAP,uB;WACqB,kBAAV,eAAa,iBAAO,CAAP,IAAb,C;UAAa+B,sBAkZB,
OAIrZB,C;UAA5C,ajBtsRO,W;UiBu8R,kBAiZmB,O;UAHbZ,2B;UAHb,OAAbg,CAAHB,C;YAAa,B,Y;YACZ
,C;YAAa+b,Y;YACd,MAAO,WAAl,WAAlJ,C;UEAX,qBAAO,M,...;QAkZP,yB:o;KAhBJ,C;iFAmBa,yB;MAIZA,
gD;MAAA,gE;MAkZA,gD;QAQW,B;w,...;UAHbz,QAHHB,IAAI,mBAAJ,C;
YAAe,qBAAO,OAaMZH,OAaZG,C;YAAP,uB;WACqB,kBAAV,eAAa,iBAAO,CAAP,IAAb,C;UAAa+B,sBAkZB,
Open Source Used In DNAC Platform Antelope
1.3177 xdg-basedir 4.0.0

1.3177.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3178 libpng 1.6.37

1.3178.1 Available under license:

Copyright (c) 1998-2008 Greg Roelofs. All rights reserved.
This software is provided "as is," without warranty of any kind, express or implied. In no event shall the author or contributors be held liable for any damages arising in any way from the use of this software.

The contents of this file are DUAL-LICENSED. You may modify and/or redistribute this software according to the terms of one of the following two licenses (at your option):

LICENSE 1 ("BSD-like with advertising clause"):

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, disclaimer, and this list of conditions.
2. Redistributions in binary form must reproduce the above copyright notice, disclaimer, and this list of conditions in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

   This product includes software developed by Greg Roelofs and contributors for the book, "PNG: The Definitive Guide," published by O'Reilly and Associates.

LICENSE 2 (GNU GPL v2 or later):

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

---------------------------------------------------------------------------
pnm2png / png2pnm --- conversion from PBM/PGM/PPM-file to PNG-file

copyright (C) 1999-2019 by Willem van Schaik <willem at schaik dot com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or
implied, including but not limited to the warranties of merchantability,
fitness for a particular purpose and noninfringement. In no event shall the
authors or copyright holders be liable for any claim, damages or other
liability, whether in an action of contract, tort or otherwise, arising from,
out of or in connection with the software or the use or other dealings in the
software.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE
=========================================

PNG Reference Library License version 2
=========================================

* Copyright (c) 1995-2019 The PNG Reference Library Authors.
* Copyright (c) 2018-2019 Cosmin Truta.
* Copyright (c) 1996-1997 Andreas Dilger.
* Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind,
express or implied, including, without limitation, the warranties
of merchantability, fitness for a particular purpose, title, and
non-infringement. In no event shall the Copyright owners, or
anyone distributing the software, be liable for any damages or
other liability, whether in contract, tort or otherwise, arising
from, out of, or in connection with the software, or the use or
other dealings in the software, even if advised of the possibility
of such damage.

Permission is hereby granted to use, copy, modify, and distribute
this software, or portions hereof, for any purpose, without fee,
subject to the following restrictions:
1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

- Simon-Pierre Cadieux
- Eric S. Raymond
- Mans Rullgard
- Cosmin Truta
- Gilles Vollant
- James Yu
- Mandar Sahastrabuddhe
- Google Inc.
- Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the
list of Contributing Authors:

   Tom Lane
   Glenn Randers-Pehrson
   Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are
Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88,
and are distributed according to the same disclaimer and license as
libpng-0.88, with the following individuals added to the list of
Contributing Authors:

   John Bowler
   Kevin Bracey
   Sam Bushell
   Magnus Holmgren
   Greg Roelofs
   Tom Tanner

Some files in the "scripts" directory have other copyright owners,
but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors"
is defined as the following set of individuals:

   Andreas Dilger
   Dave Martindale
   Guy Eric Schalnat
   Paul Schmidt
   Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing
Authors and Group 42, Inc. disclaim all warranties, expressed or
implied, including, without limitation, the warranties of
merchantability and of fitness for any purpose. The Contributing
Authors and Group 42, Inc. assume no liability for direct, indirect,
incidental, special, exemplary, or consequential damages, which may
result from the use of the PNG Reference Library, even if advised of
the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this
source code, or portions hereof, for any purpose, without fee, subject
to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their
We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.
3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients’ exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year  name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.
You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.3179 jsonpointer 4.1.0
1.3179.1 Available under license:

The MIT License (MIT)

Copyright (c) 2011-2015 Jan Lehnardt <jan@apache.org> & Marc Bachmann <https://github.com/marcbachmann>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3180 babel-plugin-transform-es2015-template-literals 6.22.0 6.22.0
1.3180.1 Available under license:

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3181 fined 1.2.0

1.3181.1 Available under license:

The MIT License (MIT)

Copyright (c) 2016, 2017, 2018 Blaine Bublitz <blaine.bublitz@gmail.com> and Eric Schoffstall <yo@contra.io>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3182 http-signature 1.2.0

1.3182.1 Available under license:

Copyright Joyent, Inc. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

1.3183 is-negative-zero 2.0.0

1.3183.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3184 protobuf-java 3.13.0

1.3184.1 Available under license:

No license file was found, but licenses were detected in source scan.

// Copyright 2008 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/google/protobuf/type.proto
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/DynamicMessage.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/PrimitiveNon BoxingCollection.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/NewInstanceSchemas.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/CodedInputStreamReader.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/TextFormatParseLocation.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/Service.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/NioByteString.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/ArrayDecoders.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/BlockingRpcChannel.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/MapFieldSchemas.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/BufferAllocator.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/BooleanArrayList.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/UnknownFieldSetSchema.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/ProtobufArrayList.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/UnknownFieldSet.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/UnknownFieldSetLiteSchema.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/SingleFieldBuilder.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/UnknownFieldSchema.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/MessageInfo.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/MapFieldLite.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/UnmodifiableLazyStringList.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/empty.proto
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/MapFieldSchema.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/MessageSchema.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/DiscardUnknownFieldsParser.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/MessageOrBuilder.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/TextFormatEscaper.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/CodedInputStream.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/AbstractProtobufList.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/Descriptors.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/JavaType.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/ExtensionRegistry.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/Writer.java
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
jar/com/google/protobuf/api.proto
* /opt/ws_local/PERMITS_SQL/1092310281_1600708753.46/0/protobuf-java-3-13-0-sources-
1.3185 python-setuptools 28.8.0

1.3185.1 Available under license:

Copyright (C) 2016 Jason R Coombs <jaraco@jaraco.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3186 graceful-fs 4.1.11 4.1.11

1.3186.1 Available under license:

The ISC License

Copyright (c) Isaac Z. Schlueter, Ben Noordhuis, and Contributors

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR
IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
1.3187 is-date-object 1.0.2
1.3187.1 Available under license:
The MIT License (MIT)

Copyright (c) 2015 Jordan Harband

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3188 make-dir 2.1.0
1.3188.1 Available under license:
MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

1.3189 async 0.9.0
1.3189.1 Available under license :
Copyright (c) 2012 Marcelo Silveira

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3190 babel-node 7.13.13
1.3190.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3191 commander 4.1.1 4.1.1

1.3191.1 Available under license :
(The MIT License)

Copyright (c) 2011 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3192 groovy-test 3.0.8

1.3192.1 Available under license :
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.

Found in path(s):
* /opt/cola/permits/1218299125_1634929835.17/0/groovy-test-3.0.8-sources-
jar/groovy/mock/interceptor/package.html

No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/1218299125_1634929835.17/0/groovy-test-3.0.8-sources-
jar/groovy/util/AbstractTestBuilder.groovy
* /opt/cola/permits/1218299125_1634929835.17/0/groovy-test-3.0.8-sources-jar/groovy/test/GroovyAssert.java
* /opt/cola/permits/1218299125_1634929835.17/0/groovy-test-3.0.8-sources-
jar/groovy/mock/interceptor/LooseExpectation.groovy
* /opt/cola/permits/1218299125_1634929835.17/0/groovy-test-3.0.8-sources-jar/groovy/util/GroovyTestCase.java
* /opt/cola/permits/1218299125_1634929835.17/0/groovy-test-3.0.8-sources-
jar/groovy/mock/interceptor/StubFor.groovy
* /opt/cola/permits/1218299125_1634929835.17/0/groovy-test-3.0.8-sources-jar/groovy/test/GroovyTestSuite.java
* /opt/cola/permits/1218299125_1634929835.17/0/groovy-test-3.0.8-sources-
1.3193 pkg-dir 2.0.0

1.3193.1 Available under license:

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3194 ajv 6.12.6 6.12.6

1.3194.1 Available under license:

The MIT License (MIT)

Copyright (c) 2015-2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3195 joi 6.10.1 6.10.1

1.3195.1 Available under license:
Copyright (c) 2012-2014, Walmart and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* The names of any contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/joi/graphs/contributors

1.3196 domhandler 2.4.2 2.4.2

1.3196.1 Available under license:
Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3197 bufferutil 1.2.1 1.2.1

1.3197.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
1.3198 merge-descriptors 1.0.1
1.3198.1 Available under license:
(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3199 yauzl 2.10.0
1.3199.1 Available under license:
The MIT License (MIT)

Copyright (c) 2014 Josh Wolfe

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3200 stack-trace 0.0.10
1.3200.1 Available under license:

Copyright (c) 2011 Felix Geisendrfer (felix@debuggable.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3201 js-cookie 2.1.1 2.1.1
1.3201.1 Available under license:

Copyright 2014 Klaus Hartl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3202 jetty-servlet 9.3.27.v20190418

1.3203 jws 3.2.2

1.3203.1 Available under license:

Copyright (c) 2013 Brian J. Brennan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3204 http-errors 1.7.2

1.3204.1 Available under license:

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com
Copyright (c) 2016 Douglas Christopher Wilson doug@somethingdoug.com

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3205 memory-pager 1.5.0 1.5.0

1.3205.1 Available under license:
The MIT License (MIT)

Copyright (c) 2017 Mathias Buus

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.3206 babel-register 7.13.14
1.3206.1 Available under license:

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3207 lodash.assign 4.2.0

1.3207.1 Available under license:

Copyright jQuery Foundation and other contributors <https://jquery.org/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

=====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.3208 babel-runtime 6.26.0

1.3208.1 Available under license :

Copyright (C) 2012-2014 by various contributors (see AUTHORS)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) jQuery Foundation, Inc. and Contributors, All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2015 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3209 @babel/plugin-proposal-object-rest-spread 7.13.8

1.3209.1 Available under license :
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, 

Open Source Used In DNAC Platform Antelope 14516
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3210 infinity-agent 1.0.2 1.0.2
1.3210.1 Available under license :
The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3211 content-security-policy-builder 1.0.0
1.0.0
1.3211.1 Available under license :
The MIT License (MIT)

Copyright (c) 2015 Evan Hahn
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3212 bl 1.2.3

1.3212.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013-2016 bl contributors

*bl contributors listed at <https://github.com/rvagg/bl#contributors>*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
1.3213 widest-line 3.1.0

1.3213.1 Available under license:

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3214 type 1.2.0

1.3214.1 Available under license:

No license file was found, but licenses were detected in source scan.

/!
* mocha
* Copyright(c) 2011 TJ Holowaychuk <tj@vision-media.ca>
* MIT Licensed
*/

Found in path(s):
* /opt/cola/permits/1158773028_1619835415.79/0/component-type-v1-2-0-0-g168462c-1-tar-gz/component-type-cdacb78/test/mocha.js

1.3215 require_optional 1.0.1 1.0.1

1.3215.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, public display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement You may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.3216 mongodb-migrations 1.0.1 1.0.1

1.3216.1 Available under license :
MIT License

Copyright (c) 2018 Droplr, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3217 slf4j-log4j 2.8.2

1.3217.1 Available under license :
Apache Log4j SLF4J Binding
Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3218 regex-not 1.0.2

1.3218.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016, 2018, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3219 pako 0.2.9

1.3219.1 Available under license:
(The MIT License)

Copyright (C) 2014-2016 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3220 path-is-inside 1.0.2

1.3220.1 Available under license:

Dual licensed under WTFPL and MIT:

---

Copyright 20132016 Domenic Denicola <d@domenic.me>

This work is free. You can redistribute it and/or modify it under the terms of the Do What The Fuck You Want To Public License, Version 2, as published by Sam Hocevar. See below for more details.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. You just DO WHAT THE FUCK YOU WANT TO.

---

The MIT License (MIT)

Copyright 20132016 Domenic Denicola <d@domenic.me>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

Open Source Used In DNAC Platform Antelope 14530
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.3221 domutils 1.5.1

1.3221.1 Available under license:
Copyright (c) Felix Bhm
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

THIS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 ARISING IN ANY WAY OUT OF THE USE OF THIS,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3222 dom 1.0

1.3222.1 Available under license:
No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2000 World Wide Web Consortium,
 * (Massachusetts Institute of Technology, Institut National de
* Recherche en Informatique et en Automatique, Keio University). All
* Rights Reserved. This program is distributed under the W3C's Software
* Intellectual Property License. This program is distributed in the
* hope that it will be useful, but WITHOUT ANY WARRANTY; without even
* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
* PURPOSE. See W3C License http://www.w3.org/Consortium/Legal/ for more
* details.
*/

No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
* used to check the 3 constraints against each complex type
* (should be each model group):
* Unique Particle Attribution, Particle Derivation (Restriction),
* Element Declarations Consistent.
*/

No license file was found, but licenses were detected in source scan.

# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,
Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/msg/XMLSerializerMessages.properties
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xpath/regex/message JA.properties
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xpath/regex/message FR.properties
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/msg/XPointerMessages.properties
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/msg/AXPValidationMessages.properties
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/msg/XPointerMessages.properties
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/msg/XMLSchemaMessages.properties
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/msg/DSDFACM.java

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

// Unique Particle Attribution

Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/models/DSDFACM.java
No license file was found, but licenses were detected in source scan.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/datatypes/package.html

No license file was found, but licenses were detected in source scan.

/
* Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/XSModelGroupImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/traversers/XSDHandler.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xx/DateDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xx/XMLSchemaValidator.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/XSSimpleTypeDecl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/DateTimeDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/traversers/XSDUniqueOrKeyTraverser.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/YearMonthDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/MonthDayDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/DayDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/YearDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/TimeDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/DurationDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/MonthDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/AbstractDateTimeDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/xx/ASAttributeChecker.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xx/ASNotationDeclaration.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xx/ASElementDeclaration.java

No license file was found, but licenses were detected in source scan.

/*
 * Copyright (c) 2001 World Wide Web Consortium,
 * (Massachusetts Institute of Technology, Institut National de
 * Recherche en Informatique et en Automatique, Keio University). All
 * Rights Reserved. This program is distributed under the W3C's Software
 * Intellectual Property License. This program is distributed in the
 * hope that it will be useful, but WITHOUT ANY WARRANTY; without even
 * the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See W3C License http://www.w3.org/Consortium/Legal/ for more details.
 */

Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASNotationDeclaration.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASModel.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom3/as/ASElementDeclaration.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/DocumentEditAS.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/CharacterDataEditAS.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/ASNamedObjectMap.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/ASObjectList.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/DOMASWriter.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/ASDataType.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/ASObject.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/DocumentAS.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/ASContentModel.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/DOMImplementationAS.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/ASEntityDeclaration.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/NodeEditAS.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/ElementEditAS.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/DOMASException.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/DOMASBuilder.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom3/as/ASEntityDeclaration.java

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
* limitations under the License.
 */

Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DOMImplementationListImpl.java

No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/DeferredDocumentTypeImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/XSException.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/wml/dom/WMLInputElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/xs/models/XSCMLeaf.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/XMLNamespaceBinder.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/impl/dv/xs/FullDVFactory.java
Open Source Used In DNAC Platform Antelope 14541

* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/wml/dom/WMLTimerElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/jaxp/validation/DOMValidatorHelper.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/util/XMLChar.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xercesimpl/xx/XSModelImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xml/serialize/Serializer.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xercesimpl/dtd/XMLDTDValidator.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/wml/dom/WMLOptgroupElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xercesimpl/dv/xs/AnySimpleDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/util/ErrorHandlerProxy.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/wml/dom/WMLTableElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xercesimpl/xsi/parser/XMLPullParserConfiguration.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/xni/grammars/Grammar.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/html/dom/HTMLHeadingElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xercesimpl/xsi/grammars/Grammar.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xercesimpl/validation/ValidationManager.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/html/dom/HTMLListElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xercesimpl/validation/ValidationManager.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/html/dom/HTMLULListElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/wml/dom/WMLBElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/dom/AttrNSImpl.java
Open Source Used In DNAC Platform Antelope 14542

* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/dv/xs/PrecisionDecimalDV.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/dv/dtd/XML11IDREFDatatypeValidator.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/dv/ValidationContext.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/xs/XXSimpleTypeDefinition.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/dv/util/ByteListImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/xpath/regex/RangeToken.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/XMLSchemaValidator.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/io/UTF8Reader.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/xs/XSSimpleTypeDelegate.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/impl/xs/XSNamedMap.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/wml/WMLTableElement.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/xni/XMLDTDContentModelHandler.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/xni/XSNamespaceList.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/xni/XSNamedMap.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources.jar/org/apache/xerces/xni/NotationDeclarationImpl.java
Open Source Used In DNAC Platform Antelope 14549
Open Source Used In DNAC Platform Antelope 14552
jar/org/apache/xerces/impl/xx/PSVIErrorList.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/dtd/XML11IDDatatypeValidator.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/validation/DOMResultAugmentor.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/parser/XMLParserConfiguration.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xs/XSNotationDeclaration.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLElement.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/jaxp/TeeXMLDocumentFilterImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dv/dtd/ENTITYDatatypeValidator.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/WMLPrevElement.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLBrElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLCardElement.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DeferredProcessingInstructionImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xx/XSGrammarBucket.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/util/SynchronizedSymbolTable.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xml/XMLEntityDescription.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLBodyElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/dtd/XML11NSDTDValidator.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/xni/parser/XMLDTSDataSource.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/wml/dom/WMLSelectElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/html/dom/HTMLButtonElementImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/EntityReferenceImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xx/traversers/XSDGroupTraverser.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/impl/xx/XSAttributeDecl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
jar/org/apache/xerces/dom/DeferredEntityReferenceImpl.java
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-
Open Source Used In DNAC Platform Antelope 14558
Open Source Used In DNAC Platform Antelope 14560
Open Source Used In DNAC Platform Antelope 14561
Open Source Used In DNAC Platform Antelope 14564
Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/datatypes/ByteList.java

No license file was found, but licenses were detected in source scan.

/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/* Checks if this content model has had its min/maxOccurs values reduced for
 * purposes of speeding up UPA. If so, this content model should not be used
 * for any purpose other than checking unique particle attribution
 * @return a boolean that says whether this content has been compacted for UPA */

Found in path(s):
* /opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/xs/models/XSCMValidator.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

/**
* DOM Level 3 LS CR - Experimental.
* Create a new <code>LSParser</code>. The newly constructed parser may
* then be configured by means of its <code>DOMConfiguration</code>
* object, and used to parse documents by means of its <code>parse</code>
* method.
* @param mode  The <code>mode</code> argument is either
*   <code>MODE_SYNCHRONOUS</code> or <code>MODE_ASYNCHRONOUS</code>, if
*   <code>mode</code> is <code>MODE_SYNCHRONOUS</code> then the
*   <code>LSParser</code> that is created will operate in synchronous
*   mode, if it's <code>MODE_ASYNCHRONOUS</code> then the
*   <code>LSParser</code> that is created will operate in asynchronous
*   mode.
* @param schemaType  An absolute URI representing the type of the schema
*   language used during the load of a <code>Document</code> using the
*   newly created <code>LSParser</code>. Note that no lexical checking
*   is done on the absolute URI. In order to create a
*   <code>LSParser</code> for any kind of schema types (i.e. the
*   <code>LSParser</code> will be free to use any schema found), use the value
*   <code>null</code>.
* @return  The newly created <code>LSParser</code> object. This
*/

* <p>Note:</p> For W3C XML Schema [<a href='http://www.w3.org/TR/2001/REC-xmlschema-1-20010502'>XML Schema Part 1</a>] applications must use the value
* <code>"http://www.w3.org/2001/XMLSchema"</code>. For XML DTD [<a href='http://www.w3.org/TR/2000/REC-xml-20001006'>XML 1.0</a>],
* applications must use the value
* <code>"http://www.w3.org/TR/REC-xml"</code>. Other Schema languages
* are outside the scope of the W3C and therefore should recommend an
* absolute URI in order to use this method.
* @return  The newly created <code>LSParser</code> object. This
* `<code>LSParser</code>` is either synchronous or asynchronous depending on the value of the `<code>mode</code>` argument.
* `<p>` Note: By default, the newly created `<code>LSParser</code>` does not contain a `<code>DOMErrorHandler</code>`, i.e. the value of the "a`<a href='http://www.w3.org/TR/2003/WD-DOM-Level-3-Core-20030609/core.html#parameter-error-handler'>error-handler</a>`" configuration parameter is `<code>null</code>`. However, implementations may provide a default error handler at creation time. In that case, the initial value of the `<code>error-handler</code>` configuration parameter on the new created `<code>LSParser</code>` contains a reference to the default error handler.
* `@exception DOMException` NOT_SUPPORTED_ERR: Raised if the requested mode or schema type is not supported.*

found in path(s):
* `/opt/cola/permits/1217727294_1635408911.57/0/xercesimpl-2-12-0-sources-jar/org/apache/xerces/dom/CoreDOMImplementationImpl.java`

1.3223 commons-digester 1.8.1
1.3223.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Digester
Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.3224 grpc-context 1.27.1
1.3224.1 Available under license:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided You use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "["] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
1.3225 better-assert 1.0.2

1.3225.1 Available under license:
No license file was found, but licenses were detected in source scan.

# better-assert

Better c-style assertions using [callsite](https://github.com/visionmedia/callsite) for self-documenting failure messages.

## Installation

$ npm install better-assert

## Example

By default assertions are enabled, however the __NO_ASSERT__ environment variable will deactivate them when truthy.

```js
var assert = require('better-assert');

test();

function test() {
  var user = { name: 'tobi' };
  assert('tobi' == user.name);
  assert('number' == typeof user.age);
}

AssertionError: 'number' == typeof user.age
  at test (/Users/tj/projects/better-assert/example.js:9:3)
  at Object.<anonymous> (/Users/tj/projects/better-assert/example.js:4:1)
  at Module._compile (module.js:449:26)
```
## License

(The MIT License)

Copyright (c) 2012 TJ Holowaychuk &lt;tj@vision-media.ca&gt;

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Found in path(s):
* /opt/cola/permits/1110635455_1607050460.22/0/better-assert-1-0-2-4-tgz/package/Readme.md

### 1.3226 jackson-annotations 2.4.0

#### 1.3226.1 Available under license:

This copy of Jackson JSON processor annotations is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
1.3227 bytes 3.0.0 3.0.0

1.3227.1 Available under license:
(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Jed Watson <jed.watson@me.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3228 junit-jupiter-engine 5.7.0

1.3228.1 Available under license:
Eclipse Public License - v 2.0
--------------------

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC
LICENSE (AGREEMENT). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

### 1. Definitions

Contribution means:
* **a)** in the case of the initial Contributor, the initial content Distributed under this Agreement, and
* **b)** in the case of each subsequent Contributor:
* **i)** changes to the Program, and
* **ii)** additions to the Program;
where such changes and/or additions to the Program originate from and are Distributed by that particular
Contributor. A Contribution originates from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

Contributor means any person or entity that Distributes the Program.

Licensed Patents mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

Program means the Contributions Distributed in accordance with this Agreement.

Recipient means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

Derivative Works shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

Modified Works shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

Distribute means the acts of **a)** distributing or **b)** making available in any manner that enables the transfer of a copy.

Source Code means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

Secondary License means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

### 2. Grant of Rights

**a)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

**b)** Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

**c)** Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual
property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

**(d)** Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

**(e)** Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

### 3. Requirements

**3.1** If a Contributor Distributes the Program in any form, then:

* **a)** the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

* **b)** the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
  * **i)** effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  * **ii)** effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
  * **iii)** does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
  * **iv)** requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

**3.2** When the Program is Distributed as Source Code:

* **a)** it must be made available under this Agreement, or if the Program **(i)** is combined with other material in a separate file or files made available under a Secondary License, and **(ii)** the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and
* **b)** a copy of this Agreement must be included with each copy of the Program.

**3.3** Contributors may not remove or alter any copyright, trademark, attribution notices, disclaimers of warranty, or limitations of liability (notices) contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

### 4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners
and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor (Commercial Contributor) hereby agrees to defend and indemnify every other Contributor (Indemnified Contributor) against any losses, damages and costs (collectively Losses) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: **a)** promptly notify the Commercial Contributor in writing of such claim, and **b)** allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

### 5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN AS IS BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

### 6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient’s rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

#### Exhibit A - Form of Secondary Licenses Notice

> This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}.

Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Open Source Licenses

This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE-<subcomponent>.md files.
1.3229 decamelize 2.0.0

1.3229.1 Available under license:
MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3230 compressible 2.0.18

1.3230.1 Available under license:
(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Jeremiah Senkpiel <fishrock123@rocketmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3231 socket.io-adapter 0.5.0 0.5.0
1.3231.1 Available under license :
(The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3232 isobject 3.0.1 3.0.1
1.3232.1 Available under license :
The MIT License (MIT)

Copyright (c) 2014-2017, Jon Schlinkert.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software,
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3233 parse-passwd 1.0.0
1.3233.1 Available under license:
The MIT License (MIT)

Copyright (c) 2016 Brian Woodward

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3234 socket.io-parser 3.3.1
1.3234.1 Available under license:
The MIT License)

Copyright (c) 2014 Guillermo Rauch <guillermo@learnboost.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3235 gson 2.2.4

1.3236 minimist 1.2.5 1.2.5
1.3236.1 Available under license:

This software is released under the MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1.3237 guice 4.0

1.3237.1 Available under license:

Google Guice - Core Library
Copyright 2006-2015 Google, Inc.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3238 drill-java-exec 1.19.0

1.3238.1 Available under license :

Drill : Exec : Java Execution Engine
Copyright 2021 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTicular PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License);
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3239 pcre 2.48.2-0ubuntu4.4
1.3239.1 Available under license:

PCRE LICENCE

Please see the file LICENCE in the PCRE distribution for licensing details.

End
PCRE LICENCE

-------------

PCRE is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language.

Release 8 of PCRE is distributed under the terms of the "BSD" licence, as specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself. The data in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also included in the distribution is a set of C++ wrapper functions, and a just-in-time compiler that can be used to optimize pattern matching. These are both optional features that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS

-----------------------------

Written by: Philip Hazel
Email local part: ph10
Email domain: cam.ac.uk

University of Cambridge Computing Service,

Copyright (c) 1997-2020 University of Cambridge
All rights reserved.
PCRE JUST-IN-TIME COMPILATION SUPPORT
-------------------------------------

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

Copyright(c) 2010-2020 Zoltan Herczeg
All rights reserved.

STACK-LESS JUST-IN-TIME COMPILER
---------------------------------

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

Copyright(c) 2009-2020 Zoltan Herczeg
All rights reserved.

THE C++ WRAPPER FUNCTIONS
----------------------------

Contributed by: Google Inc.

Copyright (c) 2007-2012, Google Inc.
All rights reserved.

THE "BSD" LICENCE
------------------

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

End

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3240 @babel/helper-replace-supers 7.13.12
1.3240.1 Available under license:
MIT License

Copyright (c) 2014-present Sebastian McKenzie and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2021 Cisco Systems, Inc. All rights reserved.